

Proceedings of

ICCC 2016

Paris | 27 June - 1 July

**7th International Conference
on Computational Creativity**



9 782746 691551

François Pachet, Amilcar Cardoso, Vincent Corruble, Fiammetta Ghedini (Editors)

Sony CSL
Paris, France

<http://computationalcreativity.net/iccc2016/>

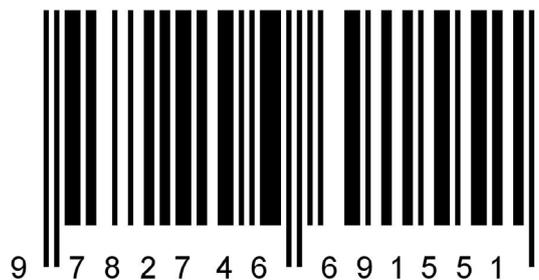
First published 2016

TITLE: PROCEEDINGS OF THE SEVENTH INTERNATIONAL CONFERENCE ON COMPUTATIONAL CREATIVITY

EDITORS: François Pachet, Amilcar Cardoso, Vincent Corruble, Fiammetta Ghedini

ISBN: 9782746691551

Technical editor: Fiammetta Ghedini



Preface

This volume contains the papers presented at ICCC 2016, the 7th International Conference on Computational Creativity held in Paris from June 26th to July 1st, 2016. The conference was hosted at Université Pierre & Marie Curie, in Paris.

Computational creativity is the art, science, philosophy and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative. As a field of research, this area is thriving, with progress in formalising what it means for software to be creative, along with many exciting and valuable applications of creative software in the sciences, the arts, literature, gaming and elsewhere. The ICCC conference series, organized by the Association for Computational Creativity since 2010, is the only scientific conference that focuses on computational creativity alone and also covers all its aspects.

We received 82 paper submissions, a record number for ICCC, which confirms the growing interest for this field. Papers were submitted in five categories: 1) technical papers advancing the state of art in research, 2) system and resource description papers, 3) study papers presenting enlightening novel perspectives, 4) cultural application papers presenting the usage of creative software, and 5) position papers arguing for an opinion. Each submission was reviewed by 4 program committee members and then discussed among the reviewers, if needed, to resolve controversial and borderline cases. Senior Program Committee Members led discussions and also prepared recommendations based on the reviews and discussions. In total, over 400 reviews and meta-reviews were carried out in the process.

The committee accepted 51 full papers. Papers were presented either as oral presentations, posters or demos, depending on the nature of the contribution.

The three-and-a-half days of the ICCC 2016 scientific program consisted in a series of exciting sessions for oral presentations of papers and a special session for posters and demos.

The program included an invited talk by Todd Lubart, Professor of Psychology, entitled “Homo Creativus: A psychological perspective”.

This conference included many events related to creativity and computers, all held on the Jussieu campus.

Two workshops were held: the 4th International Workshop on Musical Meta Creation (MUME 2016), and the 4th Computational Creativity & Games workshop (CCGW16).

Two tutorials were also organised, one on the Engagement-Reflection Model and another on Computational Creativity, organised by the PROSECCO project.

A series of talks from ERC funded projects were given, with artists drawing in real-time during the talk, to experiment with novel ways of disseminating such projects.

A concert of a Baroque comic opera “Casparo”, which tells the story of a humanoid robot, composed by Luc Steels (libretto by Oscar Villaroya) was performed. A special Flow-Machines session highlighting the main results of the project was held as well as short music performances with the interactive systems developed in this project.

This year we inaugurated a video competition (11 submissions, 4 were retained, and 2 were given a prize offered by Sony CSL). The winner of the video competition was Alida Horsley for the video *Hidden Pasts, Digital Futures*. The best paper award has been awarded to Maximos Kaliakatsos-Papakostas, Roberto Confalonieri, Joseph Corneli, Asterios Zacharakis and Emilios Cambouropoulos for the paper *An Argument-based Creative Assistant for Harmonic Blending*. We thank our sponsors, from which we received very useful support: Lip6, UPMC, Sony CSL, The Journal of Artificial Intelligence Research, the PROSECCO network, AAI. Special thanks to the ERC-funded Flow-Machines and ERCComics projects.

We thank the program committee, the senior program committee and other reviewers for their hard work in reviewing papers and the EasyChair platform that made our work easier.

François Pachet
Program chair

Amilcar Cardoso
General chair

Vincent Corruble
Local chair

Fiammetta Ghedini
Publicity chair

July 2016

Conference Chairs

General Chair: F. Amílcar Cardoso, University of Coimbra
Program Chair: François Pachet, SONY CSL Paris
Publicity Chair: Fiammetta Ghedini, SONY CSL Paris
Local Chair: Vincent Corruble, LIP6, UPMC (Paris 6)

Senior Program Committee

Oliver Bown; Design Lab, University of Sydney
Simon Colton, Goldsmiths College, University of London
Pablo Gervás, Universidad Complutense de Madrid
Nada Lavrač, Jozef Stefan Institute
Mary Lou Maher, University of North Carolina – Charlotte
Nick Montfort, Massachusetts Institute of Technology
Alison Pease, Imperial College London
Rafael Perez Y Perez, Universidad Autónoma Metropolitana at Cuajimalpa
Graeme Ritchie, University of Aberdeen
Rob Saunders, University of Sydney
Hannu Toivonen, University of Helsinki
Tony Veale, University College Dublin
Dan Ventura, Brigham Young University
Geraint Wiggins, Queen Mary, University of London

Program Committee

Kat Agres, Queen Mary, University of London
Wendy Aguilar, IIMAS - UNAM
Josep Blat, Universitat Pompeu Fabra
Giordano Cabral, UFRPE, Recife, Brazil
Michael Cook, Goldsmiths College, University of London
Joseph Corneli, Goldsmiths, University of London
Vincent Corruble, LIP6, Université Pierre et Marie Curie (Paris 6)
Alberto Diaz, Universidad Complutense de Madrid
Mark d'Inverno, Goldsmiths, University of London
Arne Eigenfeldt, Simon Fraser University
Liane Gabora, University of British Columbia
Ashok Goel, Georgia Institute of Technology
Andrés Gómez de Silva Garza, Instituto Tecnológico Autónomo de México
Hugo Gonçalves Oliveira, CISUC, University of Coimbra
Jeremy Gow, Goldsmiths, University of London
Kazjon Grace, University of North Carolina at Charlotte
Raquel Hervás, Universidad Complutense de Madrid
Amy K. Hoover, University of Central Florida
Bipin Indurkha, AGH University of Science and Technology
Anna Jordanous, University of Kent
Robert Keller, Harvey Mudd College
Carlos León, Universidad Complutense de Madrid
Antonios Liapis, University of Malta
Maria Teresa Llano Rodriguez, Goldsmiths, University of London
Ramon Lopez De Mantaras, IIIA – CSIC

Penousal Machado, CISUC, Department of Informatics Engineering, University of Coimbra
Pedro Martins, University of Coimbra
Brian Magerko, Georgia Institute of Technology
Ruli Manurung, Faculty of Computer Science, Universitas Indonesia
Jon McCormack, Monash University
David Meredith, Aalborg University
Diarmuid O'Donoghue, National University of Ireland, Maynooth
Alexandre Miguel Pinto, University Of Coimbra
Enric Plaza, IIIA-CSICS
Senja Pollak, Jozef Stefan Institute and University of Ljubljana
Matthew Purver, Queen Mary University of London
Mark Riedl, Georgia Institute of Technology
Pierre Roy, Sony CSL
Marco Schorlemmer, Artificial Intelligence Research Institute, IIIA, CSIC
Emily Short
Adam M. Smith, University of California Santa Cruz
Ricardo Sosa, SUTD, Singapore
Oliviero Stock, FBK-irst
Julian Togelius, New York University
Tatsuo Unemi, Soka University
Frank van der Velde, University of Twente
Lav Varshney, University of Illinois at Urbana-Champaign
Dekai Wu, HKUST
Ping Xiao, University of Helsinki
Georgios N. Yannakakis, Institute of Digital Games, University of Malta
Martin Znidarsic, Jožef Stefan Institute

Table of Contents

<i>Preface</i>	5
Keynote Talk	
<i>Homo Creativus: A psychological perspective</i>	viii
Todd Lubart	
Search	
<i>Novelty-Seeking Multi-Agent System</i>	1
Simo Linkola, Tapio Takala and Hannu Toivonen	
<i>Supportive and Antagonistic Behaviour in Distributed Computational Creativity via Coupled Empowerment Maximisation</i>	9
Christian Guckelsberger, Christoph Salge, Rob Saunders and Simon Colton	
<i>Mere Generation: Essential Barometer or Dated Concept?</i>	17
Dan Ventura	
<i>Searching for Surprise</i>	25
Georgios N. Yannakakis and Antonios Liapis	
<i>Role of Simplicity in Creative Behaviour: The Case of the Poietic Generator</i>	33
Antoine Saillenfest, Jean-Louis Dessalles and Olivier Auber	
Evaluation	
<i>Investigating Listener Bias Against Musical Metacreativity</i>	42
Philippe Pasquier, Adam Burnett and James Maxwell	
<i>Preference Models for Creative Artifacts and Systems</i>	52
Debarun Bhattacharjya	
<i>Evaluating digital poetry: Insights from the CAT</i>	60
Carolyn Lamb, Daniel Brown and Charles Clarke	
<i>Dependent Creativity: A Domain Independent Metric for the Assessment of Creative Artifacts</i>	68
Celso França, Luis Fabricio Wanderley Goes, Alvaro Amorim, Rodrigo Rocha and Alysson Ribeiro Da Silva - Regent	
Interaction	
<i>Modes for Creative Human-Computer Collaboration: Alternating and Task-Divided Co-Creativity</i>	77
Anna Kantosalo and Hannu Toivonen	
<i>Experience Driven Design of Creative Systems</i>	85
Matthew Yee-King and Mark d’Inverno	
<i>Applying Core Interaction Design Principles to Computational Creativity</i>	93
Oliver Bown and Liam Bray	
<i>Designing Improvisational Interfaces</i>	98
Jon McCormack and Mark d’Inverno	
Models of Creativity	
<i>Visual Hallucination For Computational Creation</i>	107
Leonid Berov and Kai-Uwe Kuhnberger	
<i>Crossing the horizon: exploring the adjacent possible in a cultural system</i>	115
Pietro Gravino, Bernardo Monechi, Vito D. P. Servedio, Francesca Tria and Vittorio Loreto	
<i>Computational Creativity Conceptualisation Grounded on ICCP Papers</i>	123
Senja Pollak, Biljana Mileva Boshkoska, Dragana Miljkovic, Geraint Wiggins and Nada Lavrac	
<i>Proceedings of the Seventh International Conference on Computational Creativity, June 2016</i>	v

<i>An institutional approach to computational social creativity</i>	131
Joseph Corneli	
<i>Understanding Musical Practices as Agency Networks</i>	139
Andrew R. Brown	
<i>A History of Creativity for Future AI Research</i>	147
Mark d’Inverno and Arthur Still	

Visual Arts

<i>Deep Convolutional Networks as Models of Generalization and Blending within Visual Creativity</i>	156
Graeme McCaig, Steve Dipaola and Liane Gabora	
<i>X-Faces: The eXploit Is Out There</i>	164
Joao Correia, Tiago Martins, Pedro Martins and Penousal Machado	
<i>Before A Computer Can Draw, It Must First Learn To See</i>	172
Derrall Heath and Dan Ventura	
<i>Creative Generation of 3D Objects with Deep Learning and Innovation Engines</i>	180
Joel Lehman, Sebastian Risi and Jeff Clune	
<i>Digits that are not: Generating new types through deep neural nets</i>	188
Kazakci, Mehdi Cherti and Balazs Kegl	

Narratives

<i>Murder Mystery Generation from Open Data</i>	197
Gabriella Barros, Antonios Liapis and Julian Togelius	
<i>Framing Tension for Game Generation</i>	205
Phil Lopes, Antonios Liapis and Georgios N. Yannakakis	
<i>What If A Fish Got Drunk? Exploring the Plausibility of Machine-Generated Fictions</i>	213
Maria Teresa Llano Rodriguez, Christian Guckelsberger, Rose Hepworth, Jeremy Gow, Joseph Corneli and Simon Colton	

Language and Text

<i>Exploring the Role of Word Associations in the Construction of Rhetorical Figures</i>	222
Paloma Galvan, Virginia Francisco, Raquel Hervas, Gonzalo Mandez and Pablo Gervas	
<i>Meta4meaning: Automatic Metaphor Interpretation Using Corpus-Derived Word Associations</i>	230
Ping Xiao, Khalid Alnajjar, Mark Granroth-Wilding, Kathleen Agres and Hannu Toivonen	
<i>One does not simply produce funny memes! - Explorations on the Automatic Generation of Internet humor</i>	238
Hugo Gonalo Oliveira, Diogo Costa and Alexandre Miguel Pinto	
<i>Poetry from Conceptual Maps - Yet Another Adaptation of PoeTryMe’s Flexible Architecture</i>	246
Hugo Gonalo Oliveira and Ana Oliveira Alves	
<i>Analysis of the correlations between the knowledge structures of an automatic storyteller and its literary production</i>	254
Ivan Guerrero Roman and Rafael Perez Y Perez	

Structure

<i>Flexible Generation of Musical Form: Beyond Mere Generation</i>	264
Arne Eigenfeldt, Oliver Bown, Andrew Brown and Toby Gifford	
<i>Generative Choreography using Deep Learning</i>	272
Luka Crnkovic-Friis and Louise Crnkovic-Friis	
<i>Investigating the Musical Affordances of Continuous Time Recurrent Neural Networks</i>	278
Steffan Ianigro and Oliver Bown	
<i>How Blue Can You Get? Learning Structural Relationships for Microtones via Continuous Stochastic Transduction Grammars</i>	286
Dekai Wu	

<i>A Music-generating System Based on Network Theory</i>	294
Shawn Bell, Liane Gabora	

Beyond the Fence

<i>Has computational creativity successfully made it «Beyond the Fence» in musical theatre?</i>	303
Anna Jordanous	
<i>The «Beyond the Fence» Musical and «Computer Says Show» Documentary</i>	311
Simon Colton, Maria Teresa Llano, Rose Hepworth, John Charnley, Catherine V. Gale, Archie Baron, François Pachet, Pierre Roy, Pablo Gervas, Nick Collins, Bob Sturm, Tillman Weyde, Daniel Wolff and James Robert Lloyd	

Blending

<i>Free Jazz in the Land of Algebraic Improvisation</i>	322
Claudia Elena Chirita and José Luiz Fiadeiro	
<i>An Argument-based Creative Assistant for Harmonic Blending</i>	330
Maximos Kaliakatsos-Papakostas, Roberto Confalonieri, Joseph Corneli, Asterios Zacharakis and Emiliós Cambouropoulos	
<i>A Process Model for Concept Invention</i>	338
Roberto Confalonieri, Enric Plaza and Marco Schorlemmer	
<i>Optimality Principles in Computational Approaches to Conceptual Blending: Do We Need Them (at) All?</i> ...346	
Pedro Martins, Senja Pollak, Tanja Urbancic and Amilcar Cardoso	
<i>Learning to Blend Computer Game Levels</i>	354
Matthew Guzdial and Mark Riedl	

Software Platforms

<i>The FloWr Online Plat-form: Automated Programming and Computational Creativity as a Service</i>	363
John Charnley, Simon Colton, Maria Teresa Llano Rodriguez and Joseph Corneli	
<i>Computational Creativity Infrastructure for Online Software Composition: A Conceptual Blending Use Case</i>	371
Martin Znidarsic, Amilcar Cardoso, Pablo Gervas, Pedro Martins, Raquel Hervas, Ana Alves, Hugo Oliveira, Ping Xiao, Simo Linkola, Hannu Toivonen, Janez Kranjc and Nada Lavrac	

Dance

<i>CoChoreo: A Generative Feature in iDanceForms for Creating Novel Keyframe Animation for Choreography</i> ..	380
Kristin Carlson, Philippe Pasquier, Herbert H. Tsang, Jordon Phillips, Thecla Schiphorst and Tom Calvert	
<i>ROBODANZA: Live Performances of a Creative Dancing Humanoid</i>	388
Ignazio Infantino, Agnese Augello, Adriano Manfré, Giovanni Pilato and Filippo Vella	
<i>Interactive Augmented Reality for Dance</i>;	396
Taylor Brockhoeft, Jennifer Petuch, James Bach, Emil Djerekarov, Margareta Ackerman and Gary Tyson	

Keynote Talk 2016

Homo Creativus: A psychological perspective

Todd Lubart

Biography

Todd Lubart is Professor of Psychology at the Université Paris Descartes, and Member of the Institut Universitaire de France. He received his Ph.D. from Yale University and was an invited professor at the Paris School of Management (ESCP). His research focuses on creativity, its identification and development in children and adults, the role of emotions, the creative process and intercultural issues. Todd Lubart is author or co-author of numerous books, research papers, and scientific reports on creativity, including the books *Defying the crowd: Cultivating creativity in a culture of conformity* (NY: Free Press, 1995), *Psychologie de la créativité* (The psychology of creativity, Paris: Colin, 2003), and *Enfants Exceptionnel* (Exceptional Children, Bréal, 2006). He is the co-founder of the International Centre for Innovation in Education (ICIE), and the associate editor of *Gifted and Talented International*.



Abstract

What is creativity and what are its psychological underpinnings? More than a century of research in psychology provides an initial understanding of the definition of creativity, sources of individual differences, and ways to measure them. The “ingredients” of creativity including both cognitive and personality facts will be highlighted. Then the way these ingredients come into play during the creative process of producing ideas will be explored based on work in diverse domains, such as the fine arts, literary composition, design and engineering. The role of a favorable environment, including social and technological facets will be discussed. Finally, work on the appreciation and uptake of creative productions in the field will be presented.

SEARCH

Search 

Novelty-Seeking Multi-Agent Systems

Simo Linkola

Department of Computer Science and HIIT
University of Helsinki
slinkola@cs.helsinki.fi

Tapio Takala

Department of Computer Science
Aalto University School of Science
tta@cs.hut.fi

Hannu Toivonen

Department of Computer Science and HIIT
University of Helsinki
hannu.toivonen@cs.helsinki.fi

Abstract

This paper considers novelty-seeking multi-agent systems as a step towards more efficient generation of creative artifacts. We describe a simple multi-agent architecture where agents have limited resources and exercise self-criticism, *veto* power and voting to collectively regulate which artifacts are selected to the domain i.e., the cultural storage of the system. To overcome their individual resource limitations, agents have a limited access to the artifacts already in the domain which they can use to guide their search for novel artifacts.

Creating geometric images called spirographs as a case study, we show that novelty-seeking multi-agent systems can be more productive in generating novel artifacts than a single-agent or monolithic system. In particular, *veto* power is in our case an effective collaborative decision-making strategy for enhancing novelty of domain artifacts, and self-criticism of agents can significantly reduce the collaborative effort in decision making.

Introduction

Novelty is often considered a central component of creativity (e.g. Boden (1992)). Obviously, an artifact that is not novel can hardly be considered creative. This paper studies the capability of cooperative multi-agent systems to seek and produce *novel* artifacts, and the effects of social decision-making strategies on this capability. Our focus is on seeking novelty; other aspects of creativity, such as surprise and value, are left for future work.

According to the systems view of Csikszentmihalyi (1988), creative systems consist of three intertwined parts: individual agents, society and domain. A set of interacting agents forms a society. The domain is a cultural component constructed by the society by selecting artifacts worth preserving. Each part in the system is in constant interaction with other parts, e.g. individuals try to learn from the domain and bring about transformations, while it is the society that collectively decides which transformations are valued and stored in the domain.

In this work, we view the agent society as a whole, and consider the artifacts introduced to the domain as the end result of the agent population's cultural knowledge of the artifact type. From this point of view, it is important that

the agent society is capable of distributed self-regulation in controlling which artifacts are accepted to the domain.

We examine how the number of agents, the amount of their collective resources and their access to the domain amalgamate with decision-making strategies of the society. Specifically, we are interested in how self-criticism, voting and *veto* power (the ability of individual agents to reject artifacts) enhance the overall novelty of artifacts accepted to the domain. Further on, we study how much work the system has to do to produce a certain amount of domain artifacts. In our case study, we use simple agents that create spirographs.

Our main contribution is the study of overall novelty of domain artifacts produced using different social decision-making strategies, especially self-regulation and *veto* power.

This paper is structured as follows. After reviewing related work in the next section, we describe the novelty-seeking agent architecture. We then illustrate and evaluate the architecture using spirographs as the artifacts.

Related Work

Multi-agent systems are a large research area (for an overview, see, e.g., Shoham and Leyton-Brown (2009)). Within the field, our work can be characterized as a system with multiple autonomous agents, where the agents diverge in information they possess (they each have a location and some memory) but not in their interests (they all aim to generate novel artifacts). Further on, the agents are cooperative rather than competitive. The focus of this work is on creativity of agent systems and more specifically on novelty-seeking agents. Next, we briefly review related work on creative agents; a more comprehensive overview can be obtained from the review of computational social creativity by Saunders and Bown (2015).

We build our research upon existing work on creative and curious agents, especially work done by Saunders and Gero.

Saunders and Gero (2001a) present a curious agent searching for novelty in the space of geometric images produced by a spirograph. The agent learns a categorization of the produced images by showing them as input to a self-organized map, or SOM (Kohonen 1995). The novelty of a new image is computed as the pixel-wise deviation from the best matching cell's image in the SOM. The agent's curiosity is modeled as a tendency to make smaller mutations in the generating parameters when more novelty is found. This

helped the agent to concentrate on areas in the parameter space where more variability was found.

In another experiment they let a society of agents seek novelty in images produced by genetic programming (Saunders and Gero 2001b). The agents have variable degrees of curiosity, modeled as a hedonic function that gets its maximum at a certain level of novelty. The agents communicate through their creations, giving positive feedback to those artifacts that match their hedonic function. Societal formations, such as cliques, were found to emerge.

We have adopted a similar approach, simulating a society of communicating agents that try to produce novel spirographs. However, we do not utilize the hedonic function but seek only to maximize novelty. Moreover, the agents in our experiments do not learn a model, such as a SOM, of previously seen artifacts. Instead, they memorize a limited number of the encountered artifacts as they are. This is a simpler solution and also less sensitive to parameters of the model (e.g. those of SOM).

Sosa and Gero (2005) have studied design as a social phenomenon with change agents (designers) and adopter agents (consumers). They conclude that emergent social phenomena — such as gatekeepers and opinion leaders — can stem from simple social mechanisms, and that the effect of an individual on a society depends both on the individual attributes and on the social structures.

Gabora and Tseng (2014) have studied a society of agents capable of inventing and imitating ideas, and of realizing the ideas as actions. In their work, each agent has a set of limbs and the agents make actions by moving the limbs. Gabora and Tseng (2014) observe that societies where agents can chain simple actions to more complex ones obtain higher average fitness and that self-regulation increases the mean diversity of the actions.

Finally, Lehman and Stanley (2008) introduce a novelty search where the main interest is not, *per se*, in satisfying certain objective goal. Instead, the aim is to find a diverse set of behaviors, i.e. behaviors that are novel enough with respect to other behaviors in the set. The search for an expanding set of novel behaviors often leads to a point where a fixed objective goal is also satisfied. Our work has a similar interest, a set of novel behaviors or artifacts, but we consider multi-agent systems without central control.

Agent Architecture

We now describe our architecture of a novelty-seeking agent system. The designs of individual agents and the society of agents have been kept as simple as possible. We make no claims of the novelty of the architecture; rather, our contribution is in the aim to maximize the diversity of artifacts created and the experimental results concerning factors behind the resulting diversity. We outline the big picture of the architecture first and then give the details.

We have a society (population) S of homogeneous agents. Each agent $S_i \in S$ has a fixed amount of resources at its disposal, in particular a constant amount of individual memory; in other respects, the agents are identical.

We model the behavior of the population via iterations: at each iteration, each agent creates a candidate artifact based

on its current position and memory. Agents then proceed to collectively decide which of the candidate artifacts to add to the domain.

In our model, the agents can be self-critical and choose not to present their own artifact as a potential candidate. They can also exercise *veto* power to reject other agents' candidates. The agents are cooperative so self-criticism and especially the *veto* power are intended to be used for the benefit of the society, not of any individual agent.

We will next more closely explain how individual agents function, and then how the multi-agent system operates as a whole.

Individual Agents

We consider agents that have a generative function producing artifacts from one or more parameters. In our model (following Saunders and Gero (2001a)), the agents live in the generative function's parameter space and can only explore different artifacts by moving in the parameter space.

Agents appreciate artifacts based on their novelty: the more novel the artifact is to the agent, the more it is appreciated. To this end, each agent has a limited memory of artifacts, and a function which can measure a distance between any two artifacts. An agent can memorize artifacts it sees during the process to its memory. If the memory is full, memorizing a new artifact will erase the oldest one.

An agent calculates the novelty of a new artifact as the minimum distance between the new artifact and any artifact currently in the agent's memory. More precisely, an agent S_i with artifact memory M_i of size m , $M_i = (A_1, A_2, \dots, A_m)$, calculates the novelty $N_i(A)$ of artifact A to be

$$N_i(A) = \min_{A' \in M_i} d(A, A'), \quad (1)$$

where $d(\cdot)$ is the distance function.

Pseudocode for the behavior of a single agent is given in Algorithm 1; details are given in the text below.

Algorithm 1 Agent behavior during a single iteration

- 1: invent a new artifact close to the agent's current location and move to the new location
 - 2: **if** the new artifact passes self-criticism **then**
 - 3: memorize the new artifact
 - 4: publish the new artifact as a candidate for the domain
 - 5: **end if**
 - 6: participate in social decision making to select which artifact, among candidates published by all agents, is added to the domain
 - 7: select and memorize artifacts from domain
-

To invent a new artifact and to move to a new location (line 1), the agent considers a fixed number of possible new locations using random walk in the parameter space (called a search beam). For each possible location, it then considers the artifact produced by the respective parameter values and chooses the one with maximum novelty with respect to the agent's own memory. It then moves to the corresponding position in the parameter space.

In order to model self-criticism, agent S_i has a novelty threshold s_i which it uses to determine if the created artifact is novel enough for its liking (line 2). If the created artifact passes the threshold, i.e. if $N_i(A) \geq s_i$, the agent memorizes the artifact and also publishes it as a potential domain artifact candidate (lines 3–4). In a single agent setting, these published artifacts will create the domain on their own.

Multi-Agent Architecture

To keep our model simple, the multi-agent system runs with minimal agent-to-agent interaction. The interactions are done solely via generated artifacts and are twofold: (1) agents use collective decision making to select artifacts to the domain D , and (2) agents can examine and memorize current domain artifacts in D to guide their own search.

In each iteration, domain artifact candidates are published by individual agents. The selection to the domain takes place in two phases (line 6).

First, agents exercise *veto* power: any agent S_i rejects any other agent's artifact A whose calculated novelty is below a threshold v_i , in a manner similar to self-criticism. Formally, given a set C of candidate artifacts, the set

$$C^* = \{A \in C \mid \forall S_i : N_i(A) \geq v_i\} \quad (2)$$

of candidates survives to the next step.

Second, agents vote on which remaining artifact in C^* to add to the domain. (If C^* is empty, none is added.) The voting procedure considers the calculated novelties of artifacts in C^* , and the winner is the artifact A^* which is considered on average most novel:

$$A^* = \arg \max_{A \in C^*} \left(\frac{1}{|S|} \sum_{S_i \in S} N_i(A) \right). \quad (3)$$

The artifact A^* is then added to the domain D .

Agents have access to the domain artifacts which they can examine and memorize (line 7). Memorizing an artifact will add it to the agent's memory (and erase the oldest artifact from the memory if its full). In our model, agents have two means to explore domain artifacts: draw k artifacts at random or select the closest k artifacts in the parameter space. We will denote these domain artifact memorizing strategies as random_k and closest_k . In both strategies the agent memorizes the artifacts blindly in the sense that a single artifact can appear multiple times in the agent's memory.

The domain is a set of artifacts, but for notational purposes we consider it as a temporally ordered sequence of artifacts $D = (A_1, A_2, \dots, A^*)$. This allows us later to denote all the artifacts in the domain up to the j th artifact by $D^j = (A_1, A_2, \dots, A_j)$.

Case study: Spirographs

We illustrate the novelty-seeking agent architecture by generating spirographs, a type of geometric images, like Saunders and Gero (2001a) did. While generation of a spirograph is a mechanistic process given the necessary parameters, finding parameter values that produce creative spirographs — in our case more specifically novel ones — is a non-trivial problem.

Spirograph

Spirograph is a toy used to draw epicyclic curved patterns with two interlocking gears of different sizes. A rotating gear (g) of radius r is positioned next to a fixed gear (G) of radius R such that the gear's teeth interlock. A pen fixed to some point in g at distance ρ from the center draws a pattern when the gear is rotated. Points on the curve are given by equations

$$x = (R \pm r) \cos(\theta) + \rho \cos(\theta + t) \quad (4)$$

$$y = (R \pm r) \sin(\theta) + \rho \sin(\theta + t) \quad (5)$$

where the sign of r determines whether g is exterior or interior to G . θ is the rotation of g 's center around G , and t is the rotation of g self, given by

$$t = \theta(R - r)/r. \quad (6)$$

The pen's movement is cyclic, returning to the starting point when both gears have made an integer number of rotations, i.e. when $\theta = 2\pi N/R$, where N is the least common multiple of r and R . Small N gives distinguishable calligraphic patterns, whereas shaded circular bands result when r/R tends towards irrational ($N \rightarrow \infty$).

A real physical spirograph is constrained by $R > 0$ and $\rho < r$, and $r < R$ if g is inside G . In our experiment, we use an abstract computational toy, allowing any (real) values in the formula. Without loss of generality, R can be fixed and r, ρ defined relative to that. Values of $\rho > r$ (meaning that the pen is outside of g) and $\rho < 0$ are also possible, though the latter only produces mirrored equivalents of positive values (the pen is in a reversed position w.r.t. g 's center).

Compared to Saunders and Gero (2001a) the main difference is that we also let the pen radius ρ vary, giving us two parameters to mutate while traversing the search space.

A Spirograph-Generating Agent

We will now describe in detail how a spirograph-generating agent in our experiments behaves. As described above, we run our agents in a simulation where each agent is triggered to act on every iteration. Agents follow the procedure illustrated in Algorithm 1 every time they act.

Agents live in the 2-dimensional parameter space of spirographs, where the location of an agent is determined by its values for r and ρ . Each point (r, ρ) in the parameter space corresponds to a single spirograph defined by r, ρ , and $R = 200$. Agents are initialized to start at random locations in the continuous parameter space by drawing the initial location (r, ρ) from the uniform distribution $r, \rho \sim \mathcal{U}(-199, 199)$.

Spirographs are first drawn as 500×500 greyscale images where gear G is located in the center. Because r can be negative (gear g is exterior to G), some areas of the parameter space actually produce plain white images as the whole spirograph is drawn outside the image.

To reduce the spirograph generation time, each spirograph is drawn with only 20 full rotations of gear g around gear G 's center. This has the effect that some spirographs are only drawn partially, but as neither the completeness of the spirographs nor the generating function is in the focus here,

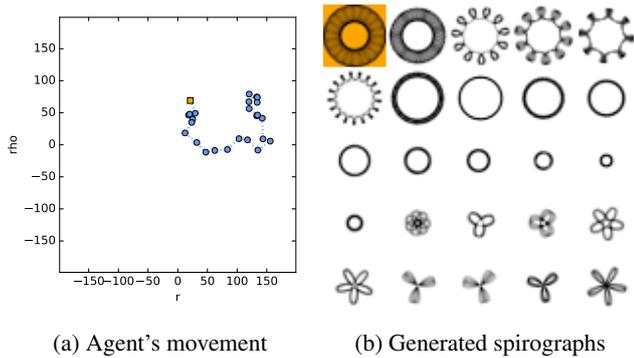


Figure 1: A single agent’s behavior, its movement in the 2-dimensional parameter space (1a) and generated spirographs (ordered left-to-right, top-to-bottom) (1b).

it does not affect the experiments. Finally, to reduce evaluation time, spirographs are rescaled to 32×32 greyscale images.

For inventing a new spirograph, an agent located in a point (r, ρ) in the parameter space considers a fixed amount of new points around it. Each new point (r', ρ') is sampled from a two-dimensional normal distribution with $r' \sim \mathcal{N}(r, 8)$ and $\rho' \sim \mathcal{N}(\rho, 8)$, then both r and ρ are clamped to $-199 \leq r, \rho \leq 199$, and a spirograph corresponding to the point is created as described above.

For each new spirograph, its novelty is calculated as in Equation 1, and the spirograph considered the most novel is selected. The difference $d(\cdot)$ between two images, used in the equation, is defined as the Euclidian distance between the 1024–element vectors formed from grey-scale values of each 32×32 image’s pixels. Although this does not fully correspond to perceptual distance between images, it technically serves our purpose.

Figure 1 illustrates a sample of 25 iterations of a single agent’s behavior, its movement in the parameter space and the spirographs it has created.

Evaluation

We next report on empirical evaluation of the proposed agent architecture using spirographs as the creative artifacts.

The questions we aim to answer empirically are the following. (1) How does the number of agents affect the novelty of artifacts produced to the domain? (2) What is the effect of the beam size on the performance? (3) How does self-criticism of agents affect the novelty, and what is the effect of the *veto* power? (4) How does agents’ access to the domain affect novelty? We also study how these factors affect the rate at which artifacts are introduced to the domain.

Experimental Setup Novelty can be difficult to define in many domains, and it obviously depends a lot on the background. In the experiments of this paper, the novelty of each artifact added to the domain is measured in relation to the artifacts that the agent society has already added to the domain. Such a measure allows comparison across different

Simulation parameter	Default value
Target domain size, $ D $	200
Number of agents, $ S $	16
Self-criticism threshold, s_i	3.2
<i>Veto</i> power threshold, v_i	3.2
Total search beam width	256
Total agent memory	512
Memorization strategy	closest ₃

Table 1: Default parameter values for the experiments.

systems that aim to produce novel artifacts of the same type, whether they are single-agent or multi-agent systems.

Let A_j denote the artifact added to the domain D as its j th artifact. The novelty of A_j is measured as its distance to the nearest artifact already in the domain:

$$N^j(A_j) = \min_{A' \in D^{j-1}} d(A_j, A'), \quad (7)$$

where D^{j-1} is the set of artifacts in the domain before A_j is added to it. Further on, we define $N^1(A_1) = 0$.

Based on the novelty of individual artifacts in the domain, we define an aggregate measure as the average over all artifacts’ novelties:

$$N^*(D) = \frac{1}{|D| - 1} \sum_{2 \leq j \leq |D|} N^j(A_j), \quad (8)$$

and use $N^*(D)$ to compare performance of different system configurations.

In the experiments, we simulate the agent system until a fixed number (200) of artifacts has been accepted to the domain and compute their mean novelty N^* as the measure how novel the artifacts in the domain are on average.

The effort needed to produce a given number of artifacts varies across different settings since the exercise of self-criticism and *veto* power can result in iterations with no candidate artifacts at all. We therefore also study the number of iterations of the agent system needed to produce the artifacts.

Each agent has some resources, in particular a fixed amount of memory and a search beam (the number of locations it considers per iteration). To make comparisons fair across different numbers of agents, the total amount of these resources in the society are kept constant when the number of agents varies.

(There are other aspects that affect the computational complexity but they are ignored here. For instance, with the above division of a constant amount of memory across agents, a society consisting of a smaller number of agents makes a larger total number of comparisons between artifacts in the search beams and the memory. On the other hand, a larger society spends more efforts on mutual evaluation, vetoing, and voting on candidate artifacts produced by the society.)

The default parameter values of our experiments are listed in Table 1. The total search beam width and agent memory are divided equally to agents.

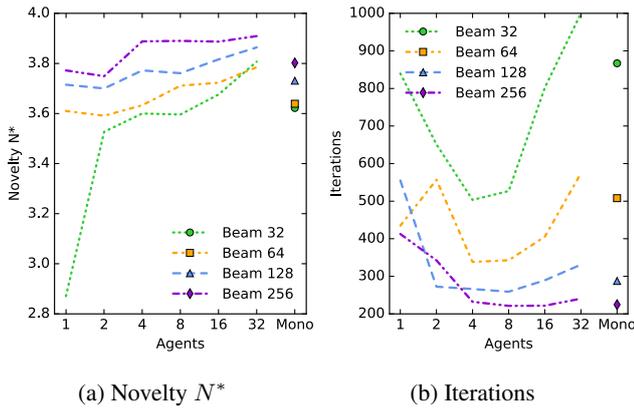


Figure 2: Effect of the number of agents on the novelty N^* (2a) and on the effort required to produce 200 novel artifacts to the domain (2b). Points at the right ends of the panels are for the baseline method Mono.

Results

We now report our experimental results with the above-described architecture of novelty-seeking agents.

Population size The effect of population size on the overall behavior of an agent system is of key interest. Ideally, a multi-agent system should have emergent properties that a single-agent system does not have while not introducing excessive overhead due to agent communication and coordination.

Figure 2 shows how the behavior of our multi-agent system is affected by the number of agents in the society. Different lines show different search beam widths; for now, consider the shapes of the curves, we will return to a comparison between them below.

Panel 2a shows that the overall novelty N^* of artifacts added to the domain increases with the number of agents. This is a desired effect for an agent architecture and indicates that agent collaboration, in particular the selection of artifacts to the domain works effectively. The effect is clearer with smaller beam widths (lower lines in the figure).

Panel 2b complements the picture by showing the corresponding effort, expressed in terms of the number of iterations required to produce 200 novel artifacts to the domain. Here, we observe a less trivial behavior when the number of agents increases. First, the required effort drops until about 4 agents. This is explained by the fact that a larger number of agents can search a more diverse set of options. The required effort starts to increase, however, when the number of agents grows further. When the number of agents grows, the society also becomes collectively more critical about the novelty of candidate artifacts. In our case, some 16–32 agents seem to be the critical amount, but the exact amount is of course dependent on the application.

The two panels of Figure 2 illustrate an inherent trade-off in systems like this: the more critical the society, the higher the novelty of its output is but smaller in size. Based on the

figure, in our setting some 4–16 agents seem to give a good compromise between quality and efficiency.

We next briefly compare the results of the multi-agent system to three different simple alternatives.

First, a comparison to a single-agent system with otherwise similar functionality and identical resources (Figure 2, leftmost points of the lines) shows that as a rule, a multi-agent system produces more novelty and often in less time than a single agent.

Second, an efficient and simple method to obtain 200 spirographs is to sample 200 random points uniformly from the parameter space. Artefacts produced this way have an average novelty of $N^* = 1.14$, markedly lower than the novelties obtained by agent systems with at least two participants (3.5–3.9).

Third, consider a monolithic hybrid between the two baselines above called “Mono”. Mono has no location in the parameter space and so it does not use random walk. It instead samples points uniformly from the parameter space at each iteration and, like our agents, chooses the best of them at each iteration. The Mono system also exercises self-criticism/veto with the same threshold as the agents. In contrast to our agents, Mono has a complete memory of the domain artifacts and is maximally informed in that sense.

A comparison to the novelty obtained by the Mono baseline (panel 2a, separate points at the right end of the panel) shows that from approximately four agents up, agent societies are competitive with and even outperform the monolithic system with complete memory. At the same time, the agent system is more effective in producing the 200 artifacts, up to some 16 agents (panel 2b).

Search beam width Let us now consider the different search beam widths in Figure 2.

First, a comparison of the relative performances of different search beam widths gives the expected results: a wider search finds more novel results (2a) and does it more effectively (2b). Among the different beam widths, the narrower ones tend to be more interesting because a common assumption in multi-agent systems is that the agents are relatively simple and operate under severe resource constraints. In contrast, when the beam width grows without limit, agents start to have complete information about the search space.

As already suggested above, different search beam widths behave differently when the number of agents is changed. As a rule, the number of agents has a larger effect when the search beam is narrow. This is natural, since with narrow beams the individual agents are more constrained. A larger number of agents helps overcome the limitation and find more novel results (2a). On the other hand, when the number of agents becomes large, self-criticism and especially the *veto* power hit the constrained agents harder and they need a longer time to find novel results (2b).

Selection of candidates to the domain We now move on to consider how different methods to select candidates to the domain affect the behavior of the society. This is the central social aspect of our model: we model social interaction by

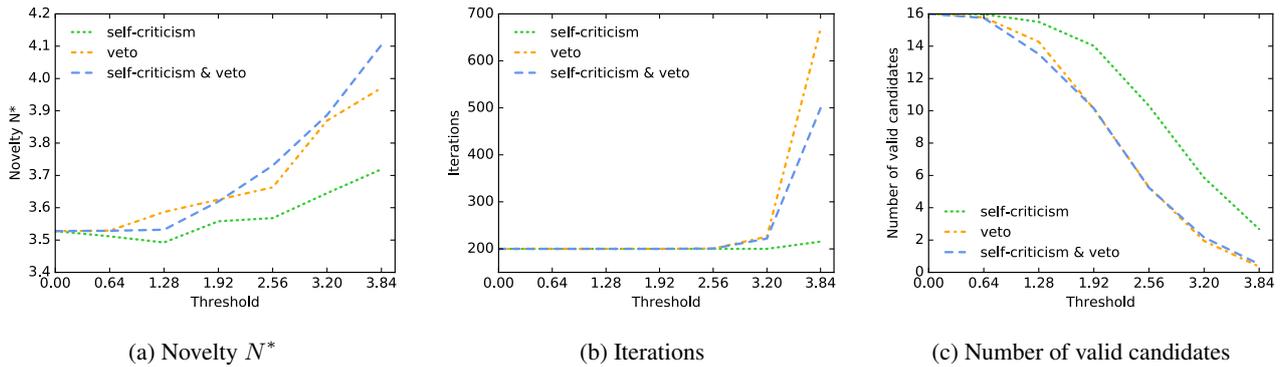


Figure 3: Effect of self-criticism and *veto* thresholds on the novelty N^* (3a), on the effort required to produce 200 novel artifacts to the domain (3b) and on the number of artifacts passing the thresholds (3c).

submission and evaluation of candidate artifacts and collaborative selection of which of them to add to the domain.

Self-criticism and veto power. Recall that the selection of candidates to the domain is controlled by two thresholds, the self-criticism threshold s_i and the *veto* threshold v_i , and an artifact is acceptable if its novelty is not lower than the respective threshold. For simplicity, in our experiments the thresholds are not agent-specific but rather constant across all agents.

Figure 3 illustrates the effects of self criticism and *veto* power using three curves in each panel: one where the threshold s_i for self-criticism varies over the experiments and the *veto* threshold is zero, one where the *veto* threshold v_i is varied and the self-criticism threshold is zero, and one where both are varied in synch ($s_i = v_i$).

Figure 3a shows how the novelty of artifacts selected to the domain varies as a function of the threshold. The immediate and expected observation is that a higher threshold increases the novelty of artifacts.

It is more interesting to compare the three curves. Among them, using a threshold for self-criticism has the smallest effect, while using a *veto* threshold has a much more pronounced effect. In the case of *veto* power, the effect of the threshold is multiplied when it is applied by multiple agents, even if they are on average less informed of the kind of artifacts produced by an agent than the agent itself. The result speaks for the “wisdom of the crowd”. The effect of using both thresholds is practically equal to just using *veto* with the same threshold.

Figure 3b shows the corresponding amounts of efforts required to produce 200 novel artifacts to the domain. The results are very sensitive to the *veto* threshold: the required effort grows suddenly at a certain point while the self-criticism threshold has at the same point almost no effect.

The conclusions from panels 3a and 3b are two-fold. First, the use of *veto* power and self criticism can improve the novelty of results significantly without increasing the effort needed. Second, however, an excessive *veto* threshold can have a sudden negative effect on the efforts. This is at least partially due to our application, spirographs, and how the generating function can only generate certain types of

images causing the distance between any two images to cap at ~ 4.5 .

Figure 3c provides further insight into the use of resources when the thresholds change, by showing how many artifacts on average pass the threshold(s) per iteration. Obviously, higher thresholds reduce the amount of valid candidates. In our setting, at a *veto* threshold of 3.84 the number of valid candidates drops approximately to 0.5 artifacts per iteration, causing a deep increase in the number of iterations needed to produce the required number of artifacts to the domain (panel 3b).

The most interesting result here is the effect of self-criticism: it controls the number of candidate artifacts submitted, reducing the efforts invested by the society to evaluating and selecting candidates to the domain. It turns out that self-criticism behaves nicely: its use improves novelty (3a) without increasing the number of iterations much (3b), but most importantly it can effectively reduce the collective evaluation effort of the agent society (3c).

Voting method. In addition to the ‘best mean’ voting method to choose one of the candidates from C^* to add to the domain, we also experimented with several other voting methods, namely ‘best singular’, ‘least worst’ and ‘instant run-off voting’ (IRV). In ‘best singular’ voting, an artifact with the highest single agent’s novelty calculation is chosen. ‘Least worst’ can be seen conceptually as a variant of the *veto* mechanism: it chooses an artifact which has least worst single novelty calculation. In ‘IRV’, agents first rank all candidates to a preference order, and then proceed to recursively prune candidates from the rankings based on which are not in the first place in any of the already pruned ranking lists.

Our empirical results with these alternative voting methods (not shown) indicate that ‘best mean’ clearly outperforms ‘best singular’ and ‘least worst’ methods and is on par with ‘IRV’. We use ‘best mean’ because of its simplicity.

Domain memorization In our model, agents have a limited memory of both their own experience and of artifacts in the domain. In each iteration, an agent accesses k artifacts in the domain and uses them to replace the oldest artifacts in the agent’s memory. We experiment with mem-

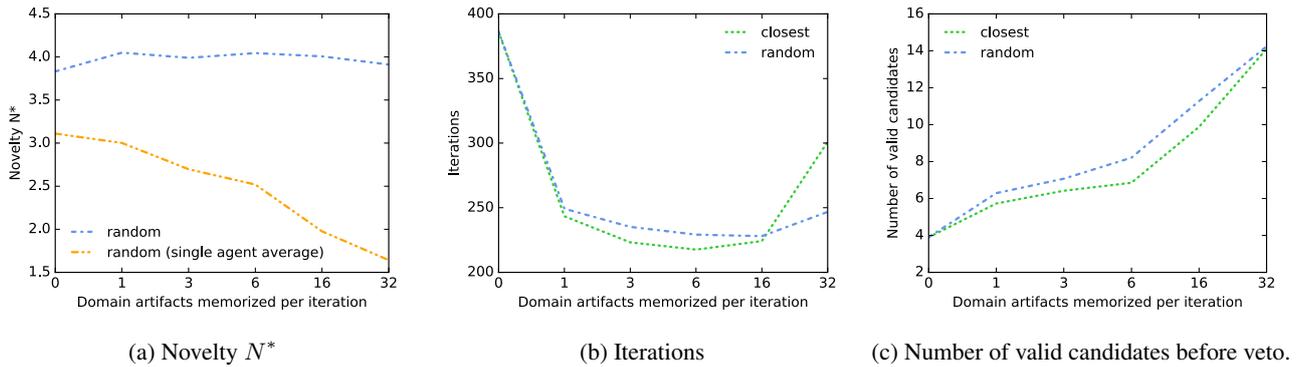


Figure 4: Effect of domain artifact memorization on novelty N^* of the domain compared to that of a single agent (4a), on the effort needed to produce 200 novel artifacts to the domain (4b) and on the number of artifacts passing the self-criticism threshold (4c).

orization techniques closest_k and random_k (as explained in section Multi-Agent Architecture) in a setting of 16 agents, each with 32 slots of memory, and the number of memorized items varying as $k \in \{0, 1, 3, 6, 16, 32\}$. Obviously, with $k = 0$ there is no memorization from the domain and the agents generate artifacts independently. The results are shown in Figure 4.

The upper line in Figure 4a shows that k has practically no effect on N^* (for clarity, we show random_k only, as the behavior of closest_k turned out to be practically identical; see Discussion). The lower line shows for different memorization settings the average novelty of a single agent, i.e. N^* computed from the candidate artifacts an agent has produced itself. In contrast to the overall novelty (the upper line), a larger value of k has a negative effect on the average performance of a single agent, which plunges to about $1/2$ when $k = 32$. This is expected: as k grows, an agent has less memory about its own products (at most one own artifact per k artifacts from the domain) and therefore is more prone to produce similar artifacts again.

Figure 4b shows the efforts needed to produce 200 domain artifacts. We observe that any amount of memorization produces the artifacts in about $2/3$ of the iterations compared to what $k = 0$ needs, but the memorization strategy does not seem to have much impact. The effort needed is at its lowest when $k \in \{3, 6\}$, and rises somewhat at $k = 32$ when the agent’s whole memory is repopulated at each iteration.

Figure 4c shows the average number of candidates that passed an agents’ self-criticism on each iteration. The curves are strictly increasing with k , suggesting that memorization of domain artifacts has a positive effect on guiding a single agent’s search.

Overall, the memorization with a conservative k (in our case $k \in \{3, 6\}$) has a positive effect on the society when comparing to $k = 0$ as the multi-agent system performs more efficiently as a whole (4b). The optimum appears to be a compromise: with very low k the society takes more time to produce the domains artifacts (4b) while high k overrides the self-criticism (4c) as the agents do not remember their own artifacts, lowering their own individual novelty (4a).

Discussion

We discuss selected technical aspects, reliability of the results we obtained, and paths towards creative multi-agent systems.

Population size With random initialization, smaller populations are clearly more prone to system-wide aberration (higher iteration counts) as all agents might be initialized into unproductive areas of the parameter space. Increasing the number of agents improves the average effectiveness of our multi-agent systems as at least some agents are more likely to be instantiated in (or at least near to) the productive areas.

Selection of candidates to the domain At a first sight, self-criticism and *veto* power seem to be surprisingly effective: self-criticism lowers the amount of collective effort needed to choose domain artifacts, and *veto* increases their novelty. However, in our setting each candidate artifact still needs to be evaluated by all agents. As a future work, it would be useful to revise the domain selection procedure to be more local in order to acquire better scalability.

The effects of the population size and social decision-making methods in our experiments are similar to what Sosa and Gero (2005) report. In small populations the effect of interaction between individuals is limited because of the low number of agents, and larger populations take more time to form a consensus. In our experiments this is reflected in how smaller populations do not reach as high overall novelty for the artifacts (Fig. 2a), and the time to reach a certain number of artifacts grows in larger populations as more agents exhibit their right to use *veto* power (Fig. 2b).

Memorization The two memorization strategies introduced, random_k and closest_k , behave nearly identically in the experiments, although one could think that the more informed closest_k would guide the agent’s search more effectively. Our initial examination suggests that the identical behavior might be influenced by two different reasons. First, the topology of the parameter space in our experiments is complex: a small change in the parameters can cause a rapid change in the artifacts. This fluctuation might inhibit closest_k from guiding the search effectively. Second, the

number of memory slots that the society collectively has is quite large compared to the amount of domain artifacts generated. This might imply that there is enough memory for random_k to continuously sample a representative set of the domain items into the society's collective memory.

Reliability of the results Our results have been obtained through simulations that involve randomness. While randomness certainly has a high role in the suggested system, the behavior between different runs with same system settings is stable enough to make conclusions from the results.

A more important issue is how specific the results are to spirographs. Spirographs are a good test case in their complexity: sometimes even small movements in the parameter space can cause big changes in the resulting spirographs, while there also are large areas producing essentially the same result.

To test if our results hold in other domains, we experimented with agents that searched for different colors in an image, and found qualitatively similar results. In particular, the dependency of novelty and iterations on the threshold for criticism had a similar form as in Figures 3a and 3b. There appeared to be a turning point in the threshold, above which novelty is higher and the number of iterations turns into steep increase. The reason for this effect may be that the domain becomes 'saturated' in the sense that the probability of finding novel enough artifacts rapidly decreases.

Creativity vs. novelty Saunders and Gero (2001a) propose agents that have a bell-shaped hedonistic curve as a function of novelty. Such a curve can be motivated by the value related to novelty (very familiar artifacts are of no new value) and of utility of that novelty (very strange artifacts cannot be utilized). Our novelty-seeking agents just look at one side of this, since our goal has been specifically to create novel artifacts. Adding aspects of value will change the model, possibly resulting in something similar to the hedonistic curve.

The ultimate goal is to develop creative agent systems. While we have only been dealing with novelty here. Formally, a minimal addition to the current system to make the agents more creative is that each agent also has function $E(A)$ which calculates the value or aesthetics of the artifact. We could then use both the novelty and aesthetics in the voting process. They both might have their own thresholds, but aesthetics probably should not be so heavily vetoed as aesthetics is much more subjective than novelty.

Conclusions

Novelty is a key criterion for creativity (Boden 1992). We have described and evaluated a novelty-seeking multi-agent architecture as a step towards creative multi-agent systems.

Our evaluation shows that a society of novelty-seeking agents can be more productive in generating novel artifacts than a single-agent or monolithic system. Obviously, a larger number of agents can be more effective in exploring the search space.

We found out that self-criticism and *veto* power can be powerful features in novelty-seeking agent systems. Self-criticism of agents can reduce the collaborative effort in evaluating candidate artifacts, while *veto* is an effective way to collaboratively reject candidates that are not novel.

Future work for developing the novelty-seeking agent architecture has numerous possible directions. First, agents could interact in numerous ways, in particular exchanging coordinates, artifacts and their evaluations. Second, agents could be adaptive to their own experience as well as to the society, e.g. by adjusting their random walk step size, self-criticism, and use of *veto* power. Third, emergence of social phenomena like community structure would be interesting to study, and also to apply in making candidate artifact selection more local and thereby more scalable. Fourth, experiments in more domains are needed.

In our efforts to study and understand creative agent systems, the next big question will be to consider seeking both *novel* and *valuable* artifacts.

Acknowledgments

This work has been supported by the European Commission under the FET grant 611733 (ConCreTe) and by the Academy of Finland under grant 276897 (CLiC).

References

- Boden, M. 1992. *The Creative Mind*. London: Abacus.
- Csikszentmihalyi, M. 1988. Society, culture, and person: A systems view of creativity. In Sternberg, R. J., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge University Press. 325–339.
- Gabora, L., and Tseng, S. 2014. The social impact of self-regulated creativity on the evolution of simple versus complex creative ideas. In *Proceedings of the Fifth International Conference on Computational Creativity*, 8–15. Ljubljana, Slovenia: Josef Stefan Institute, Ljubljana, Slovenia.
- Kohonen, T. 1995. *Self-Organizing Map*. Berlin: Springer.
- Lehman, J., and Stanley, K. O. 2008. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (ALIFE XI)*, 329–336. Cambridge, MA: MIT Press.
- Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial Life* 21(3):366–378.
- Saunders, R., and Gero, J. S. 2001a. A curious design agent: A computational model of novelty-seeking behaviour in design. In *Proceedings of the Sixth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2001)*, volume 1, 345–350. Sydney, Australia: CAADRIA.
- Saunders, R., and Gero, J. S. 2001b. The digital clockwork muse: A computational model of aesthetic evolution. In *The AISB'01 Symposium on AI and Creativity in Arts and Science, SSAISB*, 12–21. York, UK: AISB Press.
- Shoham, Y., and Leyton-Brown, K. 2009. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. New York, NY, USA: Cambridge University Press.
- Sosa, R., and Gero, J. S. 2005. Social models of creativity. In *Proceedings of the International Conference of Computational and Cognitive Models of Creative Design VI*, 19–44. Heron Island, Australia: Key Centre of Design Computing and Cognition, University of Sydney, Australia.

Supportive and Antagonistic Behaviour in Distributed Computational Creativity via Coupled Empowerment Maximisation

Christian Guckelsberger¹, Christoph Salge², Rob Saunders^{3,4} and Simon Colton^{1,4}

¹Computational Creativity Group, Goldsmiths, University of London, UK

²Adaptive Systems Research Group, University of Hertfordshire, UK

³University of Sydney, Australia; ⁴The Metamakers Institute, Falmouth University, UK
c.guckelsberger@gold.ac.uk

Abstract

There has been a strong tendency in distributed computational creativity systems to embrace embodied and situated agents for their flexible and adaptive behaviour. Intrinsically motivated agents are particularly successful in this respect, because they do not rely on externally specified goals, and can thus react flexibly to changes in open-ended environments. While supportive and antagonistic behaviour is omnipresent when people interact in creative tasks, existing implementations cannot establish such behaviour without constraining their agents' flexibility by means of explicitly specified interaction rules. More open approaches in contrast cannot guarantee that support or antagonistic behaviour ever comes about. We define the information-theoretic principle of coupled empowerment maximisation as an intrinsically motivated frame for supportive and antagonistic behaviour within which agents can interact with maximum flexibility. We provide an intuition and a formalisation for an arbitrary number of agents. We then draw on several case-studies of co-creative and social creativity systems to make detailed predictions of the potential effect the underlying empowerment maximisation principle might have on the behaviour of creative agents.

Introduction

If we peek into a painting class, we might observe teachers prescribing certain techniques to tackle a task, and students suggesting each other different brushes or materials to achieve their goals. While the teachers' behaviour can be considered as positive but antagonistic, we understand the students as supportive. Both forms of behaviour are closely related to constraints. Constraints limit the space of possible creative trajectories, and thereby make the exploration of an initially vast set of creative possibilities feasible. This focus also allows an agent to achieve mastery of some techniques before approaching others (Stokes, 2005). Constraints also present the challenge to achieve similar results by different means. Overcoming constraints is the key to transforming a space of possibilities, one crucial aspect of creativity (Boden, 1995). We consider *positive, but antagonistic behaviour* as imposing constraints on another agent temporarily, so they can learn to master them. When maintaining constraints permanently, this can turn into harm, and lead to the sabotage of other's creative endeavours. *Support* in contrast is present if an agent actively helps another in e.g. learning a particular technique to overcome constraints.

We suggest that embracing such supportive and both positive and negative antagonistic behaviour could advance research in distributed computational creativity (CC). Reproducing such behaviour could not only improve our understanding of human creativity, but it could also prove to be essential in the construction of genuinely autonomous creative systems. If we want artificial agents to be taken seriously as partners in creative activities, we require them to challenge us. In other words, we want them to constrain the actions we can undertake, so we can practice the mastery of the remaining and discover alternative routes. Likewise, we want such systems to help us escape from situations where we are very limited in our potential interactions with a certain medium. Negative antagonistic behaviour might allow for the emergence of cliques and diverging creative paths.

The challenge of realising such behaviour in a distributed CC system lies in finding a way to formalise and operationalise support and antagonism in an interactive and dynamic context, and preferably allow for the flexible and seamless transition between these two modes. We need to define these behaviours in a generic way, that allows agents to act in open-ended environments without clear goals, which are commonplace in CC. Such a situation cannot be mastered with predetermined behaviour, and we will address this issue by means of intrinsically motivated agents.

In this paper, we analyse existing co-creativity and social creativity systems as representatives of distributed CC, and conclude that there is no means yet to foster supportive and antagonistic behaviour in such agents without prescribing specific interactions, and thereby limiting the agents' flexibility. We suggest to use the information-theoretic, intrinsic motivation of *empowerment* (cf. Salge, Glackin, and Polani, 2014) to formalise the degree to which an agent is constrained in a creative activity. As our main contribution, we define the principle of coupled empowerment maximisation (CEM) as a generic mechanism to enable the emergence of supportive and antagonistic behaviour in distributed CC systems, without putting explicit constraints on the types of interactions. Empowerment corresponds to an agent's potential influence on the environment at a certain time. Coupled empowerment maximisation consequently motivates an agent to act in a way which maximises or minimises this capacity for other agents. Importantly, it allows to seamlessly shift between supportive and antagonistic behaviour.

Background

Co-creative and social creativity systems are only meaningful if each agent has a different perspective on a shared world, allowing them to complement each other, and for creativity to emerge from their interaction. Only embodied, situated and intrinsically motivated agents afford such a genuinely personal perspective (Guckelsberger and Salge, 2016). We will briefly describe and motivate these notions, and relate to existing projects in the field.

Embodied and Situated Agents

There is a common notion that creativity does not occur in a vacuum (cf. Jordanous (2015)). It is a *situated* activity, in that it relates to a cultural, social and personal context. Moreover, and in line with Saunders et al. (2010), we suggest that a large portion of creative behaviour, just like other processes constituting intelligence (cf. Rosch, Thompson, and Varela, 1992), is conditioned on an agent's *embodiment*. Put differently, we suggest that creativity is structured by how an agent's morphology, sensors and actuators enable its interaction with the world.

Robots are becoming increasingly popular in CC research (cf. Saunders et al., 2010; Saunders and Gemeinboeck, 2014; Brodbeck, Hauser, and Iida, 2015). Nevertheless, being embedded in the physical environment is neither sufficient nor necessary for an agent to be deemed embodied and situated. It is not sufficient because a robot could be governed by a central controller alone, following a classic computationalist approach. In contrast, embodied and situated agents must implement a tight interplay between physical and information-theoretic aspects of the agent, i.e. between the sensors, actuators, limbs and the controller. Pfeifer, Iida, and Bongard (2005) note that embodiment is only given if changes to one component can affect every other; moving from a greyscale to a color camera sensor might allow an agent to differentiate the consequences of its actions further, potentially leading to more diverse behaviour.

We take the stance that embodiment does not require a physical environment, so long as a virtual environment gives rise to the same effects. Nevertheless, many studies employ robots, because their situatedness makes the simulation of a rich environment obsolete. Furthermore, a physical environment affords a more natural interaction between humans and artificial agents. It also allows for morphological computation, where part of an agent's computational burden is taken over by its morphology, e.g. by constraining its joints. Saunders et al. (2010) argue that taking advantage of the physical world can expand an agent's behavioural range.

Being embodied and situated comes with a restricted access to the world, i.e. an agent can only perceive and affect parts of it. This leads to the emergence of an *Umwelt* (Von Uexküll, 1982), i.e. an agent's world of significance, which shapes its intrinsically motivated goals or the way that extrinsically motivated goals are perceived. Changing an agent's embodiment can change its *Umwelt*, and therefore also the way it interacts with the world and other agents. Pickering (2005) argues that this embodied and situated perspective leads to creativity when exploiting opportunities,

and overcoming embodiment-relative constraints in an environment. We believe that this systemic view represents the main motivation for distributed CC, over any mere engineering concerns. Here, creativity emerges from the interaction of multiple agents, both human and artificial, with different perspectives on the world, and on potentially shared tasks.

Embodied and situated agents also challenge the mini-me problem in CC, i.e. the problem that creativity is often attributed to the designer instead of the artificial agent. The behavioural complexity of embodied and situated agents is to a large extent determined by their interaction with the environment. Instead of explicitly programming, we have to engineer for emergence, leading to more robust behaviour which might be novel and surprising even for the designer.

Intrinsic Motivation

Pickering (2005) argues that human creativity cannot be properly understood, or modelled, without an account of how it emerges from the encounter between the world and intrinsically active, exploratory and productively playful agents. *Intrinsic motivation* was first named by White (1959) while observing animals engaging in such behaviours in the absence of an obvious reinforcement or reward. Ryan and Deci define the term from a psychological point of view as "Performing an activity for its inherent satisfactions rather than for some separable consequence" (Ryan and Deci, 2000). Being *extrinsically motivated* in contrast means to perform an activity for an externally prompted, instrumental value. Oudeyer and Kaplan (2008) complement this view with a definition informed by robotics and AI. The converging point is the reliance on the sensorimotor flow and agent-internal experience alone, independent of the involved channels' semantics. Intrinsically motivated agents are not dependent on externally defined goals, but can still form goals intrinsically. This allows for higher flexibility and adaptivity especially in open-ended environments which are commonplace in creative activities. Intrinsic motivation was identified in philosophy (Kieran, 2014) and in psychological experiments as an important factor in producing more creative artefacts (Amabile, 1985), by driving the exploration of creative options.

Related work in CC focusses mainly on the notions of curiosity and novelty, but also on surprise (Maher, Brady, and Fisher, 2013) and expectation (Grace and Maher, 2014, 2015). In co-creativity and social creativity, models of curiosity is particularly popular: Saunders (2007) developed a system of curious design agents which evolve abstract art. In *Curious Whispers*, intrinsically motivated robots generate and play music to each other (Saunders et al., 2010). Merrick and Maher (2009) employ curiosity to support the learning of tasks in adaptive characters in multiuser games. In *Accomplice*, Saunders and Gemeinboeck (2014) establish a playful interaction of curious robots with a human audience.

Co-Creativity and Social Creativity

In creativity studies, co-creativity refers to several people contributing to the creative process in a blended manner (Candy and Edmonds, 2002). In this paper, we will use the term for the more specific *human-computer co-creativity*

(Davis, 2013), describing the interaction of one person or multiple people with one or more artificial agents to generate a creative product. There are many subcategories such as mixed initiative systems, live algorithms and collaborative AI for artistic tasks, each stressing different aspects such as the order of interaction, time constraints, or the task concerned. Much research has been done on robotic live music improvisation, e.g. *Ja'maa*, a modification of the percussion robot *Haile* (Weinberg, Driscoll, and Thatcher, 2006), and the interactive Marimba player *Shimon* (Hoffman and Weinberg, 2010). Other researchers look at co-creativity in sketching: In the *Drawing Apprentice* system, a person and a software agent take turns to add to a virtual canvas (Davis et al., 2014). Jacob and Magerko (2015) investigate human-computer co-creativity in dance and interactive art by means of the *Viewpoint AI* system. Here, a human performer and virtual agent collaborate to improvise movements in real-time. A co-creative system which is less about artistic tasks is the ongoing *Computational Play Project*, where robots will eventually engage with children and toys in pretend play, i.e. the “subsequent enactment of a narrative experience using physical objects” (Magerko et al., 2014).

Within CC, the notion of *social creativity*, comprising creative cultures, creative societies, and computer social creativity, refers to computer-computer interaction, i.e. groups of artificial agents which produce and share artefacts. Co-creativity thus represents the fundamental mechanism in social creativity systems, if understood as creative interactions between purely artificial agents. There are two overlapping perspectives on the use of social creativity systems: One is inspired from research in artificial life and sociology, and employs systems as testbeds for investigating social factors in human creativity. Here, the produced artefacts are of minor interest (cf. Saunders and Bown, 2015). For instance, Steels (1995) as well as Saunders and Grace (2008) study the emergence of shared vocabularies and the formation of agent cliques engaging in “language games”.

The second perspective is directed towards the development of autonomous creative systems, and considers the mechanisms inherent in social creativity, e.g. dialogue, reflections and multiple perspectives, as means to achieve this goal (cf. Corneli et al., 2015). Such systems could produce valuable artefacts, but their value and novelty might be intrinsic to the system, i.e. only meaningful to the artificial agents themselves. They often draw on concepts from cognitive science such as the *Blackboard* architecture which is inspired by the *Global Workspace* model (Baars, 2005). The latter expresses the idea that distributed sources of knowledge or different roles, represented by competing mental processes, can be leveraged to cooperatively solve problems that no single constituent could solve alone.

The Blackboard architecture is particularly popular in poetry and narrative generation, but these systems struggle to incorporate competition and cooperation in a flexible and adaptive way. Only few implement a strong coupling between an agent’s body and its environment: The story generator by Laclaustra et al. (2014) creates stories by recording the interaction of multiple agents with different roles. Eigenfeldt investigated music in social creativity, both by means

of the software agent ensembles *Drum Circle* (Eigenfeldt, 2007) and *Musebots* (Eigenfeldt, Bown, and Carey, 2015), which other researchers can modify to produce a collective composition. A physical realisation of an ensemble is given by the 12 arm drum robot *MahaDeviBot*, where each limb is controlled by one agent (Eigenfeldt and Kapur, 2008).

Social creativity systems usually employ at least one agent to direct the flow of actions. For instance, Laclaustra et al. (2014) define the role of a “director” in their story generation system, which sets new goals for the acting agents. In the *Virtual Storyteller*, Theune et al. (2003) use a director to introduce new characters and objects, give characters specific goals, and deny them to perform certain actions. Similarly, Eigenfeldt’s ensembles employ a conductor agent to control the composition (Eigenfeldt, Bown, and Carey, 2015). In co-creativity, the human is usually, but not always in control and introduces goals into the system: *Curious Whispers* encourages people to interact with the robots via a synthesiser. While some take a traditional “master” role in trying to teach the robots tunes, others act more passively and try to learn from- and copy the robots. Our formalism is designed to foster sensible agent behaviour in both cases.

Case-Studies

We conducted three case-studies to analyse if and how present co-creativity and social creativity systems realise supportive and antagonistic behaviour. We evaluated systems which situate intrinsically or extrinsically motivated agents in a physical or virtual environment.

Curious Whispers Developed by Saunders et al. (2010), *Curious Whispers* represents a society of intrinsically motivated robots which generate and listen to tunes. The main goal of this social creativity system is to investigate the effects of embodiment on creativity in a physical environment. Each robot is equipped with a pair of microphones, touch sensors to avoid collisions, a speaker and four wheels. The robots are driven by an intrinsic measure of interest-ness, which is quantified by mapping the novelty of the current sensor input on a Wundt curve. Novelty is quantified by comparing how new percepts are encoded in a self-organising map serving as the robot’s long-term memory. The robots can listen to two sources of sound at a time, and move closer to the one which is considered more interesting.

Saunders et al. (2010) suggest that engaging in social relations represents one crucial means of how embodiment can foster creativity. In their system, such relations remain shallow: robots play their tunes when getting bored of listening to others, and might consequently be engaged by other robots which find their tune interesting. The programmed behaviours of the individual agents in *Curious Whispers* are deliberately minimal, so any supportive and antagonistic behaviour would be an emergent property of the system. Exposure to tunes biases the robots in the generation of new instances. They thus appear to engage in mutual support to explore the space of potential tunes as they are passed between and modified by each other. Nevertheless, there is no apparent antagonistic behaviour, and at no point do the agents act to directly influence the performance of others.

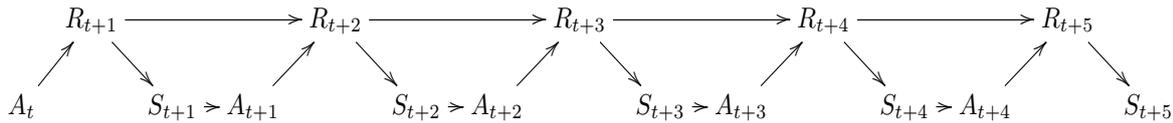


Figure 1: Causal Bayesian network representing the interaction of a memoryless agent with the world as a perception-action loop unrolled in time. Arrows denote causal dependencies between its sensors S , actuators A and the rest of the world R .

Shimon *Shimon* (Hoffman and Weinberg, 2010) is an extrinsically motivated, robot marimba player which improvises in real-time to a human pianist’s performance. It consists of four arms which can be moved separately on a shared rail in parallel to the marimba’s front side. Each arm has one mallet for the bottom-, and one for the top-row keys. A MIDI-listener attached to the electric piano and an adjustable metronome are used as sensors. The project implements embodied cognition by understanding music not as a sequence of notes, but as a choreography of movements constrained by the robots morphology. The performance is based on different interaction modules which analyse and react to the sensory input. Each responds to a different challenge, e.g. to react in time with the right tempo, or to play beat-matched, synchronised and chord-adaptive patterns.

The description of a live performance sheds light on the nature of interaction. Each interaction module matches a phase in performance, and is addressed independently through the human player who provides the notes, beat and tempo. It lasts until a certain condition, e.g. a limit of played bars, is met. This rigid setup, together with a preset of rhythms and a pre-programmed crescendo finale, shows that provided a time and stimulus, the robot performs a prescribed set of hard-coded interactions to establish support. At no time does it challenge the player.

Drawing Apprentice Davis et al. (2015) introduce a co-creative system in which an extrinsically motivated software agent and a person take turns to improvise line drawings on a virtual canvas. The *Drawing Apprentice* is based on a cognitive architecture inspired by Enactivism, which considers creativity as emerging from an improvised interaction with the environment and other agents. It differentiates three types of awareness, which are associated with different layers of perceptual logic. The system receives a line input from the user, analyses and adopts the perceptual layer the user is currently in, and generates an improvised response. Each layer focusses on a different scale of the drawing, determines how and over which timespan the system will analyse the user’s input, and puts constraints on the possible responses. For instance, the local logic only takes the user’s last input into account, and complements it by mirroring, scaling or translation. The regional logic in contrast analyses a series of past strokes and employs gestalt principles to group them, while the global logic analyses the whole composition.

The *Drawing Apprentice* reflects *Shimon*’s system architecture to some extent, as it constrains potential responses by means of dedicated modules. Nevertheless, it autonomously selects which module to perform. The system arguably supports and inspires the user in their activity, by complementing their drawing in interesting ways. Nevertheless, the sys-

tem is explicitly grounded in such supportive behaviour by design. Some responses might feel like a constraint to the user, but the design does not seem to embrace such antagonistic behaviour explicitly. Being extrinsically motivated, the system can only react in previously anticipated ways.

Summary The case-studies show us that present systems with intrinsically motivated agents exhibit emergent and thus highly flexible and adaptive behaviour, but do not have a means to establish a truly supportive or antagonistic mode of interaction. Systems with extrinsically motivated agents prescribe such interactions rigidly, but are limited to what the system designer anticipates as supportive in a certain situation beforehand, which is particularly difficult in a physical environment without a clearly defined interaction interface or fixed goals. Importantly, no project realises antagonistic behaviour. Next, we will introduce the CEM principle to overcome this situation. We later recall the case-studies and show how the principle could apply.

Formal Model

We propose the CEM principle as a candidate mechanism to enable the emergence of supportive and antagonistic behaviour in co-creative and social creativity systems, without putting explicit constraints on the interactions. We first provide an intuition and a formal definition of empowerment and the empowerment maximisation (EM) principle, followed by a formalisation of CEM and an algorithmic description for a scenario with two agents.

Empowerment and Empowerment Maximisation

Empowerment, the quantity underlying the CEM principle, is defined over the relationship between an agent’s actuators and sensors, and as such is sensitive to the agent’s embodiment and *Umwelt*. In a deterministic environment, empowerment quantifies an agent’s options in terms of availability and visibility. In a stochastic setting, this generalises to the potential influence of an agent’s actions on its environment, and to the extent to which it can perceive this influence afterwards. Empowerment is measured in bits of information (Shannon, 1948). It is zero when the agent has no control over its sensors, i.e. when all actions lead to the same perception, and it grows when different actions lead to different perceivable outcomes. For simplicity, the interaction presented here is discrete in time and space. Nevertheless, continuous implementations exist and were evaluated both in virtual environments and in robotics. An introduction to empowerment with a survey of motivations, intuitions and past research can be found in (Salge, Glackin, and Polani, 2014).

At the centre of the empowerment definition is the interpretation of an agent’s embodiment as an information-

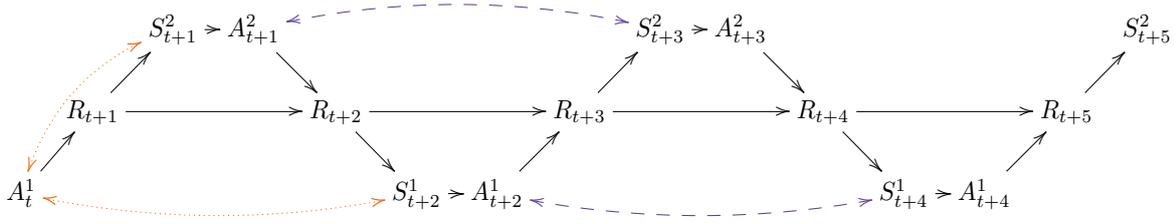


Figure 2: Perception-action loop for two agents (S^1, A^1) , (S^2, A^2) interacting in turnwise order. The first agent (S^1, A^1) is coupled to the second (S^2, A^2) . Dotted (orange) lines indicate the estimation of future sensor states and dashed (purple) lines represent the calculation of empowerment, which comprises further estimation steps.

theoretic communication channel. For any arbitrary separation between an agent and a world we can define sensor variables S and actuator variables A , as those states that allow for the in- and outflow of information to the agent, respectively. This interaction with the world is usually described as a perception-action loop (Touchette and Lloyd, 2004), which can be analysed by means of a causal Bayesian network as in Fig. 1 and Pearl’s interventional calculus (Pearl, 2000). In the figure, arrows imply causation between random variables: the agent’s actions A only depend on its sensor input S , which in turn is determined by the rest of the system R . The latter is affected both by the preceding system state and the agent’s actions. The interventional causal probability distribution $p(S_{t+1}|S_t, A_t)$ therefore represents the (potentially noisy) communication channel between actions and future sensor states.

Empowerment is then defined as the maximum potential information flow (Ay and Polani, 2008) that could possibly be induced by a suitable choice of actions, in a particular state s_t . This can be formalised as the channel capacity:

$$\begin{aligned} \mathfrak{E}_{s_t} &= \max_{p(a_t)} I(S_{t+1}; A_t) \\ &= \max_{p(a_t)} H(S_{t+1}) - H(S_{t+1}|A_t) \\ &= \max_{p(a_t)} \sum_{\mathcal{A}, S} p(s_{t+1}|s_t, a_t) p(a_t) \log \frac{p(s_{t+1}|s_t, a_t)}{\sum_{\mathcal{A}} p(s_{t+1}|s_t, \hat{a}_t) p(\hat{a}_t)} \end{aligned}$$

Here, $I(S_{t+1}; A_t)$ represents the mutual information between sensors and actuators, which is based on the difference of regular $H(S_{t+1})$ and conditional entropy $H(S_{t+1}|A_t)$. For details on these information-theoretic notions, see (Cover and Thomas, 2006).

Empowerment is *local*, i.e. the agent’s knowledge of the local dynamics $p(S_{t+1}|S_t, A_t)$ are sufficient to calculate the quantity. The information-theoretic grounding makes it domain-independent and universal, i.e. it can be applied to every possible agent-world interaction, as long as this interaction can be modelled as a probabilistic perception-action loop. This implies that it can be computed on arbitrary agent morphologies, and can cope with changes being made to it. Finally, empowerment as presented here is task-independent, i.e. it is not evaluated relative to a specific goal.

Crucially, empowerment does not measure an agent’s actual, but their potential influence on the environment. The EM principle suggests that an agent should, in absence of any explicit goals, choose actions which are likely to lead to

states with higher influence on the environment, i.e. more options. A greedy agent would thus choose the action with the highest expected empowerment, i.e. which most likely leads to future sensor states with maximum empowerment. Based on the properties outlined in the previous paragraph, EM satisfies the criteria for intrinsic motivation by Oudeyer and Kaplan (2008), which were outlined earlier.

Coupled Empowerment Maximisation

Coupled Empowerment Maximisation is defined on the embodiment of multiple agents, and represents an extension of the general maximisation principle. It is based on the observation that an agent’s actions might not only affect its own, but also the empowerment of other agents. CEM represents an action policy which explicitly considers this relationship. While we assume that an agent would always maximise its own empowerment over the long term, we suggest to look at how both maximising and minimising other agents’ empowerment shapes the behaviour of individual agents and groups. Given the general intuition of empowerment from the previous section, we hypothesise that minimising or maximising another agent’s empowerment establishes a general frame for supportive and both positive and negative antagonistic behaviour, respectively. A positive, antagonistic agent constrains others temporarily in order to benefit them over the long term. A negative, antagonistic agent in contrast maintains these constraints permanently.

We will define the coupled empowerment for an arbitrary number of agents, but limit our examples to two. For simplicity, we will also assume agents to interact in a turn-wise order. Fig. 2 shows the extended perception-action loop for two interacting agents (S^1, A^1) and (S^2, A^2) . Here, agents are considered as distinct from the rest of the world R . Due to the turnwise interaction, they do not have to account for their sensor states at intermediate stages where they are not permitted to act, e.g. the second agent at $t + 2$, and we consequently omitted these variables. The diagram shows that by performing a certain action a_t at time t , the first agent potentially affects both the second agent’s sensor state at $t + 1$, and its own sensor at $t + 2$, which in turn also depends on the second agent’s action choice at $t + 1$.

CEM suggests that the active agent chooses its actions in order to both maximise its own expected empowerment and to maximise or minimise the empowerment of the coupled agents. This is formalised by Eq. 1, the general action selection policy. Here, parameters α_i determine the influ-

$$\pi(s_t) = \arg \max_{a_t} [\alpha_n \cdot \pm E[\mathfrak{E}^n]_{a_t} + \alpha_{n-1} \cdot \pm E[\mathfrak{E}^{n-1}]_{a_t} + \dots + \alpha_1 \cdot E[\mathfrak{E}^1]_{a_t}] \quad (1)$$

ence of individual couplings. We use the notion of *expected* empowerment here, because the active agent cannot be sure about how the other agents might behave. The calculation of coupled empowerment therefore involves several estimation steps, which are illustrated by means of Alg. 1 and Fig. 2. For the supportive case and two agents, the active, first agent has to calculate the expected coupled empowerment of each of its actions a_t . As a first step, the agent has to estimate which potential follow-up sensor states of the second agent S_{t+1}^2 can be reached via a_t . The agent then has to take into account how the second agent could potentially act, in order to estimate its own future sensor state S_{t+2}^1 . This is indicated by the dotted (orange) lines in Fig. 2. From there, the agent has to perform another round of estimations in order to infer the local dynamics $P(S_{t+3}^2|a_{t+2}, s_{t+2})$ and $P(S_{t+4}^1|a_{t+3}, s_{t+3}^1)$, which eventually enable the calculation of both agents' empowerment (dashed, purple lines in Fig. 2). Finally, the agent has to calculate the expected coupled empowerment $E[\mathfrak{E}^C]_{a_t}$, by combining its own $E[\mathfrak{E}^1]_{a_t}$ and the second agent's expected empowerment $E[\mathfrak{E}^2]_{a_t}$, given the current action a_t .

CEM is not constrained to a particular number of agents; nevertheless, the computational complexity grows exponentially the more agents are involved. Note that this is not problematic if we employ several empowerment maximising agents e.g. in a social creativity system, as long as each agent is only coupled to a small number of others. Different means of optimisation exist, e.g. based on monte-carlo techniques (Salge, Glackin, and Polani, 2014), the information-bottleneck method (Anthony, Polani, and Nehaniv, 2014) and deep neural networks (Mohamed and Rezende, 2015).

Algorithm 1 Calculating the action policy of the first agent in a two-agent scenario, based on supportive CEM.

```

function  $\pi(s_t, \alpha)$ 
  for all  $a_t \in A_t^1$  do
    Estimate  $P(S_{t+1}^2|a_t, s_t)$ 
    for all  $s_{t+1} \in S_{t+1}^2$  do
       $\mathfrak{E}_{s_{t+1}}^2 \leftarrow \text{CALCEMPowerment}(s_{t+1})$ 
      for all  $a_{t+1} \in A_{t+1}^2$  do
        Estimate  $P(S_{t+2}^1|a_{t+1}, s_{t+1})$ 
        for all  $s_{t+2} \in S_{t+2}^1$  do
           $\mathfrak{E}_{s_{t+2}}^1 \leftarrow \text{CALCEMPowerment}(s_{t+2})$ 
        end for
      end for
    end for
     $E[\mathfrak{E}^2]_{a_t} \leftarrow \sum_{s_{t+1}} P(s_{t+1}|a_t, s_t) \times \mathfrak{E}_{s_{t+1}}^2$ 
     $E[\mathfrak{E}^1]_{a_t} \leftarrow \sum_{s_{t+1}} P(s_{t+1}|a_t, s_t) \times \sum_{a_{t+1}} \sum_{s_{t+2}} P(s_{t+2}|a_{t+1}, s_{t+1}) \times \mathfrak{E}_{s_{t+2}}^1$ 
     $E[\mathfrak{E}^C]_{a_t} \leftarrow \alpha \times E[\mathfrak{E}^2]_{a_t} + (1 - \alpha) \times E[\mathfrak{E}^1]_{a_t}$ 
  end for
  Perform  $a_t : a_t = \arg \max_{A_t} E[\mathfrak{E}^C]_{a_t}$ 
end function

```

Coupled Empowerment Maximisation in Computational Creativity

Saunders and Gemeinboeck stress that intrinsic motivation is at the core of the creative process, when agents engage in “a reflective exploration of possibilities” (Saunders and Gemeinboeck, 2014). Empowerment quantifies the possibilities available to an agent in a certain situation in a very generic way. Klyubin, Polani, and Nehaniv (2008) argue that empowerment maximisation could be realised by, or even help constituting specialised motivations, such as curiosity and novelty which CC research focused on in the past. The goal of this section is to provide an intuition of CEM and motivate its potential in embodied and situated distributed CC by recalling the previous examples and case-studies.

Supportive and Antagonistic Behaviour Revisited

We suggest that CEM establishes a generic frame for supportive and antagonistic behaviour to emerge in distributed CC. We already motivated the potential benefits of negative antagonistic behaviour in a social creativity scenario, but not for human-computer co-creativity: Since a positive antagonistic agent might struggle with determining the time-frame for imposing constraints, e.g. in the presence of uncertainty, we suggest to use empowerment minimisation in co-creativity as a shortcut for positive antagonism.

Davis et al. (2015) note that from an enactivist perspective, expertise in a field is not only about knowledge, but to a large part about the mastery of an agent's sensorimotor contingencies (O'Regan and Noe, 2001). Maximising empowerment, either in respect to the own or another agent's embodiment, translates to developing more nuanced action-percept couplings, and can thus be interpreted as maximising an agent's sensorimotor expertise.

Empowerment maximisation is not goal-directed, and Klyubin, Polani, and Nehaniv (2008) hypothesise it to be a good policy in the absence of any explicit goals. Nevertheless, by coupling the maximising agent to other, goal-driven agents such as the human collaborator in a co-creativity system, or to “director” and “conductor” agents in social creativity, we can induce the goals of the coupled agent into the active agent's behaviour. We would consequently expect a CEM-driven agent to either support or sabotage the current goal of the agent it is currently coupled to. At the same time, the maximising agent takes on some control, by influencing the empowerment of the other.

Recalling the Case-Studies

This section illustrates how CEM can foster supportive and antagonistic behaviour in the earlier case studies. One way to affect empowerment is by constraining or widening an agent's options directly. An agent's empowerment is maximum, if all potential actions are available and lead to clearly separable outcomes. A positive but antagonistic *Drawing Apprentice* could challenge the human co-creator by limiting its toolbox to thick brushes only, or by restricting the

colour palette to cold tones. *Shimon* could maximise its own empowerment by moving its mallets into a position which allow it to react most flexibly to the pianist in the next time step. It could support the pianist vice versa by playing a tune which allows for many potential responses.

Empowerment can also be affected by limiting the aspects which an agent's sensor can differentiate in the environment: The positive, but antagonistic *Drawing Apprentice* might switch the output of the virtual canvas to greyscale, while maintaining the internal colour scheme. The human partner would consequently perceive many colours alike. The software agent would challenge them to practice *Grisaille*, a technique of painting exclusively in shades of grey, which was particularly popular among the old masters.

Klyubin, Polani, and Nehaniv (2008) demonstrate that empowerment can serve as an immediate guide for sensor and actuator evolution during an agent's lifetime. This requires the modifying agent to have access to the actuators and sensors. In the *Drawing Apprentice*, the virtual canvas serves as proxy to the human's perceptions and actions; *Shimon* in contrast cannot directly access the perceptual apparatus of its human collaborator. It could increase its own empowerment by evolving its actuator to apply more force to its mallets, allowing for a wider range of distinct sounds.

In *Curious Whispers*, the sensors and actuators of other robots are also inaccessible. This scenario illustrates how empowerment can be affected by modifying an agent's environment. Here, it becomes most obvious how an embodied agent's behaviour is connected to its morphology and the external environment, and how important different roles and abilities are to distributed CC. In *Curious Whispers*, a negative antagonistic agent could disturb other agents listening to a performance, by playing a noise which makes it impossible to differentiate between the different sounds that were originally played. Consequently, the listening agents will not be able to pick up the exact tune for their own performances. If the other agents were able to express a different spectrum of tones, they would maximise their empowerment by switching to the part which the antagonistic agent could not disturb, eventually leading to the differentiation of artefacts. We can think of even more complex interventions: If there were movable parts in the environment, a supportive agent might improve another agent's rehearsal by moving these parts into a position where they block noise out.

Conclusion and Future Work

We suggest to understand antagonistic and supportive behaviour in distributed CC as imposing constraints on other agents, and helping them to overcome them. We translated the information-theoretic notion of empowerment to the creative domain to formalise the degree to which embodied, situated and intrinsically motivated agents are constrained in their creative activity. We then defined the principle of coupled empowerment maximisation as a means to enable support and antagonistic behaviour in distributed CC systems. We used CEM as an *intuition pump* to demonstrate which behaviours CEM-driven agents might exhibit in three existing co-creativity and social creativity systems. Although this is one possible application, the strength of CEM lies in

its capacity to allow for the implementation and subsequent emergence of supportive or antagonistic behaviour *online*.

The emergent behaviour of embodied, situated and intrinsically motivated agents is often surprising, and hard to predict. The most important next step is therefore to evaluate the formalism in an actual co-creativity or social creativity system, and to investigate the effects under experimental conditions. The examples in this paper focus on agents in a common sense. Nevertheless, we are also interested in applying CEM to scenarios where agency is attributed to objects. This might allow us to evolve artefacts to compete with others, or to maximise synergistic effects. Our research yielded a strong correspondence between the environment that an agent can act upon, and the notion of conceptual spaces. As part of future work, we want to investigate more thoroughly how (coupled) empowerment maximisation relates to the exploration and transformation of conceptual spaces. We only considered "constraints" in terms of which actions are *possible* for an agent in a certain situation. Nevertheless, in many creative processes, options can also be limited by what is *desireable*, e.g. from an aesthetics point of view. Integrating such "soft constraints" into the formalism represents another, promising avenue of research.

Acknowledgments

CG is funded by EPSRC grant EP/L015846/1 (IGGI). CS is funded by the H2020-641321 socSMCs FET Proactive project. SC's work was supported by EPSRC grant EP/J004049 (Computational Creativity Theory).

References

- Amabile, T. M. 1985. Motivation and Creativity: Effects of Motivational Orientation on Creative Writers. *Personality and Social Psychology* 48(2):393–399.
- Anthony, T.; Polani, D.; and Nehaniv, C. L. 2014. General Self-Motivation and Strategy Identification: Case Studies Based on Sokoban and Pac-Man. *TCAIG* 6(1):1–17.
- Ay, N., and Polani, D. 2008. Information Flows in Causal Networks. *Advances in Complex Systems* 11(1):17–41.
- Baars, B. J. 2005. Global Workspace Theory of Consciousness: Toward a Cognitive Neuroscience of Human Experience. *Progress in Brain Research* 150:45–53.
- Boden, M. 1995. Creativity and Unpredictability. *Stanford Humanities Review* 4(2):1–18.
- Brodbeck, L.; Hauser, S.; and Iida, F. 2015. Morphological Evolution of Physical Robots through Model-Free Phenotype Development. *PLoS one* 10(6).
- Candy, L., and Edmonds, E. A. 2002. Modeling Co-Creativity in Art and Technology. In *Proc. 4th Conf. Creativity and Cognition*, 134–141.
- Corneli, J.; Jordanous, A.; Shepperd, R.; Llano, M. T.; Misztal, J.; Colton, S.; and Guckelsberger, C. 2015. Computational Poetry Workshop: Making Sense of Work in Progress. In *Proc. 6th Int. Conf. Computational Creativity*, 268–275.
- Cover, T. M., and Thomas, J. A. 2006. *Elements of Information Theory*. Wiley-Interscience, 2nd edition.

- Davis, N.; Popova, Y.; Sysoev, I.; Hsiao, C.-P.; Zhang, D.; and Magerko, B. 2014. Building Artistic Computer Colleagues with an Enactive Model of Creativity. In *Proc. 5th Int. Conf. Computational Creativity*, 38–45.
- Davis, N.; Hsiao, C.-P.; Popova, Y.; and Magerko, B. 2015. An Enactive Model of Creativity for Computational Collaboration and Co-creation. In Zagalo, N., ed., *Creativity in the Digital Age*. Springer. 223–243.
- Davis, N. 2013. Human-Computer Co-Creativity: Blending Human and Computational Creativity. In *Proc. 9th Conf. AIIDE*, volume 2013, 9–12.
- Eigenfeldt, A., and Kapur, A. 2008. An Agent-Based System for Robotic Musical Performance. In *Proc. Int. Conf. New Interfaces for Musical Expression*, 144–149.
- Eigenfeldt, A.; Bown, O.; and Carey, B. 2015. Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble. In *Proc. 6th Int. Conf. Computational Creativity*, 134–141.
- Eigenfeldt, A. 2007. Drum Circle: Intelligent Agents in Max/MSP. In *Proc. Int. Conf. Computer Music*, 2–5.
- Grace, K., and Maher, M. L. 2014. What to Expect When You're Expecting: The Role of Unexpectedness in Computationally Evaluating Creativity. In *Proc. 5th Int. Conf. Computational Creativity*.
- Grace, K., and Maher, M. L. 2015. Specific Curiosity as a Cause and Consequence of Transformational Creativity. In *Proc. 6th Int. Conf. Computational Creativity*, 260–267.
- Guckelsberger, C., and Salge, C. 2016. Does Empowerment Maximisation Allow for Enactive Artificial Agents? In *Proc. 15th Conf. ALIFE (to appear)*.
- Hoffman, G., and Weinberg, G. 2010. Gesture-based human-robot Jazz improvisation. In *2010 IEEE Int. Conf. Robotics and Automation*, 582–587.
- Jacob, M., and Magerko, B. 2015. Interaction-based Authoring for Scalable Co-creative Agents. In *Proc. 6th Int. Conf. Computational Creativity*, 236–243.
- Jordanous, A. 2015. Four PPPPerspectives on Computational Creativity. In *AISB 2015 Symposium on Computational Creativity*, 16–22.
- Kieran, M. 2014. Creativity as a Virtue of Character. In Paul, E. S., and Kaufmann, S. B., eds., *The Philosophy of Creativity: New Essays*. Oxford Scholarship. 125–144.
- Klyubin, A. S.; Polani, D.; and Nehaniv, C. L. 2008. Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems. *PLoS one* 3(12):1–14.
- Laclaustra, I. M.; Ledesma, J. L.; Méndez, G.; and Gervás, P. 2014. Kill the Dragon and Rescue the Princess: Designing a Plan-based Multi-agent Story Generator. In *Proc. 5th Int. Conf. Computational Creativity*.
- Magerko, B.; Permar, J.; Jacob, M.; Comerford, M.; and Smith, J. 2014. An Overview of Computational Co-creative Pretend Play with a Human. In *Proc. 1st Workshop Playful Virtual Characters, 14th Conf. Intelligent Virtual Agents*.
- Maher, M. L.; Brady, K.; and Fisher, D. H. 2013. Computational Models of Surprise in Evaluating Creative Design. In *Proc. 4th Int. Conf. Computational Creativity*, 147–151.
- Merrick, K. E., and Maher, M. L. 2009. *Motivated Reinforcement Learning. Curious Characters for Multiuser Games*. Springer.
- Mohamed, S., and Rezende, D. 2015. Stochastic Variational Information Maximisation. In *Proc. 29th Conf. NIPS*, 1–9.
- O'Regan, K., and Noe, A. 2001. A Sensorimotor Account of Vision and Visual Consciousness. *Behavioral and Brain Sciences* 24:939–1031.
- Oudeyer, P.-Y., and Kaplan, F. 2008. How Can We Define Intrinsic Motivation? In *Proc. 8th Int. Conf. Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, 93–101.
- Pearl, J. 2000. *Causality: Models, Reasoning and Inference*. Cambridge University Press.
- Pfeifer, R.; Iida, F.; and Bongard, J. 2005. New Robotics: Design Principles for Intelligent Systems. *Artificial Life* 11(1-2):99–120.
- Pickering, J. 2005. Embodiment, Constraint and the Creative Use of Technology. In *Proc. Freedom and Constraint in the Creative Process in Digital Fine Art*.
- Rosch, E.; Thompson, E.; and Varela, F. J. 1992. *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press.
- Ryan, R., and Deci, E. 2000. Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology* 25(1):54–67.
- Salge, C.; Glackin, C.; and Polani, D. 2014. Empowerment – an Introduction. *Guided Self-Organization: Inception* 67–114.
- Saunders, R., and Bown, O. 2015. Computational Social Creativity. *Artificial Life* 21(3):366–378.
- Saunders, R., and Gemeinboeck, P. 2014. Accomplice: Creative Robotics and Embodied Computational Creativity. In *Proc. 50th Anniversary Convention of the AISB Symposium*.
- Saunders, R., and Grace, K. 2008. Towards a Computational Model of Creative Cultures. In *AAAI Spring Symposium: Creative Intelligent Systems*.
- Saunders, R.; Gemeinboeck, P.; Lombard, A.; Bourke, D.; and Kocaballi, B. 2010. Curious Whispers: An Embodied Artificial Creative System. In *Proc. 1st Int. Conf. Computational Creativity*, 100–109.
- Saunders, R. 2007. Towards a Computational Model of Creative Societies Using Curious Design Agents. *Engineering Societies in the Agents World VII*:340–353.
- Shannon, C. E. 1948. A Mathematical Theory of Communication. *Bell Systems Technical Journal* 27:379–423.
- Steels, L. 1995. A Self-Organizing Spatial Vocabulary. *Artificial Life* 332(1995):319–332.
- Stokes, P. D. 2005. *Creativity From Constraints: The Psychology of Breakthrough*. Springer.
- Theune, M.; Faas, S.; Faas, E.; Nijholt, A.; and Heylen, D. 2003. The Virtual Storyteller: Story Creation by Intelligent Agents. In *Proc. Conf. Techn. for Interactive Digital Storytelling and Entertainment*, 204–215.
- Touchette, H., and Lloyd, S. 2004. Information-Theoretic Approach to the Study of Control Systems. *Physica A: Statistical Mechanics and its Applications* 331:140–172.
- Von Uexküll, J. 1982. The Theory of Meaning. *Semiotica* 42(1):25–78.
- Weinberg, G.; Driscoll, S.; and Thatcher, T. 2006. Jamaa A Middle Eastern Percussion Ensemble for Human and Robotic Players. In *Proc. Int. Conf. Computer Music*, 464–467.
- White, R. W. 1959. Motivation Reconsidered. *Psychological review* 66(5):297–333.

Mere Generation: Essential Barometer or Dated Concept?

Dan Ventura

Computer Science Department
Brigham Young University
Provo, UT 84602 USA
ventura@cs.byu.edu

Abstract

The computational creativity community (rightfully) takes a dim view of supposedly creative systems that operate by *mere generation*. However, what exactly this means has never been adequately defined, and therefore the idea of requiring systems to exceed this standard is problematic. Here, we revisit the question of mere generation and attempt to qualitatively identify what constitutes exceeding this threshold. This exercise leads to the conclusion that the question is likely no longer relevant for the field and that a failure to recognize this is likely detrimental to its future health.

Introduction

For many of us in the computational creativity community the idea of artifact generation, in and of itself, has come to be looked on as something less than an accomplishment, even though in many of the domains in which we operate, the generation of even just reasonable artifacts is still well beyond the capabilities of any current system. Indeed, the expression *mere generation* has become something of a favored pejorative whose history reaches back at least to the meeting of the *Third International Conference on Computational Creativity* held in Dublin in 2012, during which the tagline, “Scoffing at Mere Generation for more than a Decade”¹, became a conference theme. This theme was explicitly revisited during the 2015 meeting of the conference in Park City, at which small buttons showing the expression “mere generation” struck through were included in the registration packets (see Fig. 1). Many of the conference attendees delightedly wore the buttons, but others at the event, especially those that may not have attended earlier conferences, were less enamored with or bemused by them, and may possibly have found them offensive.

It became clear that though at least some of us have been endorsing this dogma of disdain for many years, it is perhaps not as self-evident as we might think that it is. The purpose of this paper is to suggest that the idea of “mere generation” needs to be revisited by the community, that at the very least, we should clarify what is meant by the expression, and that, in fact, its use with respect to modern

¹Coined by the host of that conference, the inimitable Tony Veale

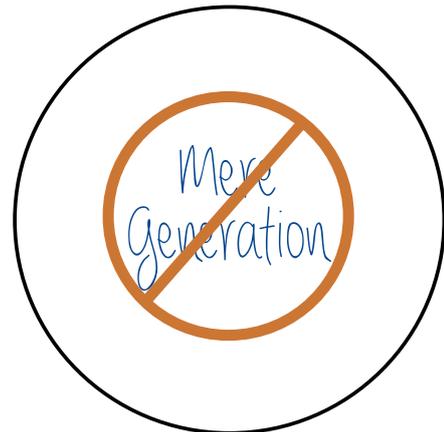


Figure 1: The button handed out in Park City at the *Sixth International Conference on Computational Creativity*, 2015. But, what does this mean, exactly?

systems claiming correspondence with the field of computational creativity should probably be deprecated.

This question of when a system has crossed the line from mere generation to something more is related to the question of system evaluation, which has begun to be addressed by work such as Ritchie’s metrics (Ritchie 2007), Colton et al.’s FACE/IDEA framework (Colton, Charnley, and Pease 2011) and Jordanous’ SPECS methodology (Jordanous 2012)². The goal of these types of approaches—and it is a critical one—is to suggest viable ways of measuring, either absolutely or relatively the “creativity level” of (some aspect of) a system. In contrast, the purpose of this paper is to argue that, by any reasonable measure, we, as a field, have at some point crossed an important threshold on our quest for computationally creative systems.

The approach we will follow here is reminiscent of a *gedanken* experiment inspired by Ritchie’s metrics (Ventura 2008)—we are going to examine a spectrum of candidate (computationally) creative processes, from *definitely-mere-generation* to *definitely-not-mere-generation*, characterized

²And even a recent blogpost at <http://www.bestofbotworlds.com/node/22>

by several prototypical algorithms that fall at different points along that spectrum. As we traverse the spectrum, we will at each point consider the existence (or lack thereof) of three characteristics: *novelty*, *value* and *intentionality*, as surrogate indicators for the existence of (some form of) creativity. Note that the first two are most commonly addressed with respect to product³, while the third deals with process. For our purposes, these characteristics will be defined as follows⁴:

novelty: the quality of being new, original or unusual; this is relative to the population of artifacts in the domain in question and can apply in the personal or historical sense.

value: the importance, worth or usefulness of something; this would typically be ascribed by practitioners of the domain in question.

intentionality: the fact of being deliberative or purposive; that is, the output of the system is the result of the system having a goal or objective—the system’s product is correlated with its process.

The prototype algorithms used to populate our spectrum are meant to be representative of the breadth of approaches under consideration by the field rather than definitive or exhaustive and are abstract enough that it is likely that the majority of historical and extant CC algorithms can be, to a fairly accurate approximation, typified by one of them or by some “convex combination” of a couple of them. However, we will not attempt to support that hypothesis here.

Instead, we will argue that the field of computational creativity as a whole has moved well past a critical threshold and that the derisory merely generative system of yore is nearly impossible to find amongst the systems that we see today, certainly those built by people conversant in the field and, in many instances, by those who are not (yet). So, we can still enjoy our scoffing, albeit without any real targets at which to direct our ridicule. This is not to suggest that our quest for computational creativity is complete—far from it—but we can say with conviction that we have thrown off our moorings and left port.

A Generation Odyssey

As a running rhetorical example, we will address the problem of generating artifacts from the Japanese poetic form *haiku*, perhaps the most famous example of which is the following by Bashō⁵

*Furu ike ya
kawazu tobikomu
mizu no oto*

³But also note that our use of “product” here is abstract, and that, in particular, the artifact produced might itself be a process.

⁴These definitions can be formalized, but for this discussion that will not be useful.

⁵A well-known translation, due to Donald Keene, that both faithfully captures the literal meaning and yet (necessarily) loses a great deal of the impact goes as follows:

*The ancient pond
A frog leaps in
The sound of water*

Algorithm 1 Generation using a stochastic process.

```
Create()
  a = {}
  while not done do
    a = a + random_atom()
  return a
```

Traditional haiku is a simple, elegant poetic form used for juxtaposing two ideas or images. It is characterized by a single-stanza structure with 17 total syllables, divided into three phrases with syllable counts 5-7-5. Themes that speak to the Japanese reverence for nature dominate the form; syntax is somewhat loose, deferring to structure; and implied semantics are favored over the explicit. We have chosen haiku because its realizations are small enough to allow analysis and demonstration of multiple examples while being complex enough to admit treatment of a range of important issues. From here on, when an example is useful, for understandability, we will sacrifice the purity of the original Japanese form and use English.

Randomization

It is difficult to conceive of a simpler form of generation than that of a stochastic process, so that is where we will begin. The first level of generation, then, consists only of producing a set of atomic elements, as shown in Algorithm 1. For the generation of *haiku*, this means simply generating some number of words and stopping, without regard for syllable count, line count or syntax (which is often not a huge concern in haiku anyway), let alone semantic cohesion or theme. An actual example, which was generated using a simple web app⁶, is shown below⁷.

*sadistic ideal adopter
devil seducer diametric
accursed blabbermouth*

Similarly meaningless output in other domains is easily conceived: a collage of random shapes (or even an image composed of random pixels), a musical composition of random notes (pitches and durations), a recipe composed of a random list of ingredients (and amounts), a neologism as a random sequence of letters, etc. There should be little question whether this system is merely generative; indeed, some may argue it doesn’t even rise to that aspiration.

The output of such a stochastic system will almost certainly be novel, in the sense that as the size of the artifact increases, the likelihood that it has been generated before by any system (computational or otherwise) becomes vanishingly small. However, this novelty is not intentional—the system is randomly choosing an artifact without any notion of novelty. At the same time, system output will almost

⁶<http://www.textfixer.com/tools/random-words.php>

⁷Produced by randomly selecting a (random) number of words. Line breaks were chosen arbitrarily for formatting purposes and are not part of the artifact.

Algorithm 2 *Generation by plagiarizing an inspiring set I.*

```
Create(I)
  a = random_select(I)
  return a
```

certainly not be valuable, by the same kind of probabilistic argument—as the size of the artifact increases, the likelihood that it has any meaning/utility/aesthetic quality becomes equally small. Another way to say the same thing: as a function of artifact size, the set of all possible candidate artifacts grows very quickly and, in particular, it grows much more quickly than does the set of valuable candidate artifacts.

Plagiarization

A significant improvement in artifact quality can be achieved by making use of an inspiring set that contains examples of quality artifacts, and the simplest use that can be made of such a set is blatant plagiarism. An abstract plagiarizing system is shown in Algorithm 2, and the output of this kind of system is significantly improved over that of Algorithm 1 in the sense that it will be real haiku, *per se*. It also has another advantage over Algorithm 1 in that it has acquired some knowledge of what haiku is. This is knowledge only in the most rudimentary sense: functionally, the system “knows” only that haiku is anything in its inspiring set and anything not in its inspiring set is not haiku. Of course, in a third sense, this system represents a step backwards from that of Algorithm 1 because it has no autonomy (the one thing that Algorithm 1 has going for it)—it cannot generate anything novel. Assuming that haiku on the web are all examples of quality artifacts (a patently ridiculous assumption that does not affect our current argument), such a plagiarizing system can be constructed by employing a simple Google search for haiku and choosing randomly from amongst the returned results. One such search, for example, turns up the following⁸:

*A cricket disturbed
the sleeping child; on the porch
a man smoked and smiled*

Similarly simple systems can easily be built for images, music, recipes, etc.

The output of a plagiarizing system reveals a character complementary to that of the stochastic: it will by definition not be novel, and, again by definition it will be valuable. Like the novelty of the stochastic system, though, this value lacks intention—the system is regurgitating an artifact without any notion of value (beyond that of implicitly ascribing value to the inspiring set). This time, the set of candidate artifacts is fixed and is a strict subset of the set of all valuable artifacts.

⁸Found at <http://examples.yourdictionary.com/examples-of-haiku-poems.html>

Algorithm 3 *Generation by memorizing an inspiring set I.*

```
Create(I)
  model = memorize(I)
  a = random_from_memory(model)
  return a
```

Memorization

A slightly more sophisticated version of Algorithm 2 builds a model of the inspiring set by memorization (see Algorithm 3). While the lookup-table approach of Algorithm 2 could be considered a form of memorization, what is meant here is that the inspiring set is re-represented in some way by the system, ideally without loss of fidelity. This is the first level at which building the system is not trivial (given commonly available resources)—a model that memorizes typically does so by overfitting data (that is, the parameters of the model are under-constrained by the data), so an inspiring set of interesting size will require a very powerful model for its memorization. As a result of this memorization, the system has “internalized” the inspiring set in a nontrivial way, though without learning any generalizing principles. If the memorization is perfect, the result is likely indistinguishable from regurgitation—any generated artifact will be a faithful copy of a member of the inspiring set, even though that set has been re-represented. However, this memorization process at least admits the possibility of some level of variance from the inspiring set, due to faulty memory, model capacity or fidelity issues, etc. For example, if the memorization process involves some form of compression, it is possible that the compression will not be lossless, resulting in errors during reproduction⁹:

*A cricket disturbed
the creeping mold; on the porch
a man choked and died*

These errors may, in fact, be thought of as features rather than bugs, and even packaged as a very simple form of creativity¹⁰; however, the system realistically only has the same level of “knowledge” of haiku as that of Algorithm 2. Indeed, because the system’s goal is memorization, any errors in reproduction that it makes would likely go undetected (by the system itself) and it has no mechanism for evaluating the quality of a perturbation—the detrimental norm will not be distinguishable from the serendipitous exception (and in this sense, we have returned to the lack of knowledge exhibited by systems like Algorithm 1).

Because a memorization system is attempting to mimic the output of a plagiarizing system, the artifacts it produces will essentially be characterized the same way: value with lack of novelty, again without intention. When errors are introduced, the two characteristics are inversely affected: novelty likely increases while value likely decreases, for the

⁹A serendipitous, if morbid, perturbation of the cricket haiku done by the author

¹⁰Though doing so would likely be construed as hucksterism by our community

Algorithm 4 *Generation by generalizing an inspiring set I.*

```
Create(I)
  model = build_model(I)
  a = generalize_from_model(model)
  return a
```

same reasons given in discussing stochastic systems. Intention is unaffected by error as the system has no mechanism for evaluation. This time, the set of candidate artifacts intersects the set of all valuable artifacts, with the size of the intersection dependent on the fidelity of the memorization (assumed to be high).

Generalization

Another step along the spectrum regains some of the autonomy that was lost with the introduction of an inspiring set. Algorithm 4 shows a system that goes beyond memorization by modeling the inspiring set in such a way that generalization is possible. This is typically accomplished by some form of regularization of the model combined with a bias (either implicit or explicit). The resulting model output can demonstrate significant variance from the inspiring set, and the trick for producing reasonable output is to discover the right amount of regularization and the right bias, both non-trivial propositions. In the case of haiku, regularization might force the model to represent words as abstract entities, such as parts of speech, and a bias might favor poems with a syllable count of (or near) 17 and/or words related to nature. As an example, such a simple generalizing model could, given an inspiring set similar to the cricket haiku above, produce something like¹¹

*The snowflake reveals
a quiet rock near a tree
a fish blows or falls*

Note that the model itself and either or both the regularization and the bias may be learned or explicitly designed, and any or all of these may be interpretable or they may not. In any case, a generalizing system must be acknowledged to have a significantly deeper knowledge about haiku than any of its three predecessors, even if that knowledge is still naïve or even somewhat incorrect. At this level we may begin to see the natural introduction of pastiche, if the model is particularly good.

Here for the first time, we begin to see artifacts that may non-vacuously exhibit both novelty and value. Novelty will be limited to the degree that the model makes explicit use of constructs found in the inspiring set. Value will be limited because any valid generalization of the inspiring set may be output. The set of candidate artifacts has increased significantly over that available to the plagiarizing and memorizing models, but is much smaller than that available to the stochastic. For the first time, the system can be said to have

¹¹Created by the author using a part-of-speech generalization of the cricket haiku, and strong bias for 17 syllables and selection of nature-related words of the requisite parts of speech

Algorithm 5 *Generation by modeling an inspiring set I and filtering candidate artifacts via a fitness function fitness().*

```
Create(I, fitness())
  model = build_model(I)
  while score <  $\theta$  do
    a = generate(model)
    score = fitness(a)
  return a
```

at least some limited (implicit) intentionality in both the novelty and value it produces: the model regularization enforces some level of generalization (and thus novelty) from the inspiring set by disallowing too much complexity; and the bias (can) enforce some notions of value.

Filtration

Moving farther along the spectrum, we see the first evidence of self-evaluation, in the form of an objective or fitness function. Algorithm 5 extends Algorithm 4 by filtering its generative results, using some notion of fitness. The model now may be designed for some other purpose than (just) generalization; the modeling step can now afford to “take more risk”, because the generated artifacts are vetted after the generative step. In order to be useful, the fitness function should evaluate aspects of the model not already implicitly managed by the generalizing model. For example, if the model includes a bias for 17-syllable stanzas, it is likely redundant for the the fitness function to compute a score for syllable count. Instead, the fitness function will be most useful for measuring holistic characteristics of the artifact. In the case of haiku, these might include notions such as overall valence or affect of the stanza, semantic relationships among constituent words, novelty, etc., and several of them could be composed in some way to compute the fitness score. The use of such a filter would likely preclude the creation of the snowflake haiku (which was generalized from the cricket haiku) because it would score poorly for semantic cohesion, and as a result probably low in (at least) affect as well. However, another haiku generalized by the same model could score significantly better, passing the fitness threshold and therefore being output as a viable artifact¹²:

*The sunlight reveals
a quiet path near a brook
a tree drinks or sleeps*

Use of a filtering function consequent to a generative step can be thought of (somewhat simplistically) as analogous to a musician listening to her composition after writing it or a chef tasting a dish after he conceived the recipe for it (we will see a better approximation to this in systems further along the spectrum).

Just as with the generalizing model, the filtering model can produce both novelty and value, and for the same rea-

¹²Also created by the author using the same generalization model as the previous section, with serendipitous word choices that increase the semantic cohesion and affect

Algorithm 6 Knowledge-based generation by modeling an inspiring set I , employing a fitness function `fitness()` and leveraging a knowledge base K .

```
Create( $I$ , fitness(),  $K$ )
  model = build_model( $I$ )
  while score <  $\theta$  do
    a = generate(model,  $K$ )
    score = fitness(a)
  return a
```

sons. However, both the value and the novelty are likely to be increased because for the first time we see explicit intention in the form of the fitness function. Further, since the fitness function can, at least notionally, address both value (by filtering for semantics, affect, etc.) and novelty (by filtering using some form of distance from inspiring set), both characteristics can be said for the first time to be intentional. This is a significant milestone.

Inception

Yet another level of generation is attained with the addition of a knowledge-base, which is used to affect/augment the model, consequently injecting additional depth and nuance into the generalization process, thereby allowing the fitness threshold to be increased, and leading to better artifacts. In Algorithm 6, the knowledge-base is incorporated solely into the generative step, but variations can include it in the modeling step and/or the fitness evaluation as well. It can be very general or domain specific. In the case of haiku, useful domain knowledge would include such things as semantic relationships amongst words, alternative grammatical constructs and exceptions, common facts, metaphor, etc.

Such a system might produce a variation on the tree haiku like the following¹³:

*In golden torpor
while insects hum over a rill
an old oak dozes*

In the excellent movie *Inception*, Leonardo DiCaprio and his team are given the task of infiltrating a man's mind, while he is in an induced dream-state, in order to implant an idea. The tricky part is that for the idea to germinate, the man must not realize that it has been implanted but must instead believe that it originated with himself. To avoid detection, the team induces a dream-within-a-dream-within-a-dream scenario, obfuscating their presence by constructing multiple levels of indirection. At least as an end game, the CC community faces a similar challenge—how to inject knowledge into a computationally creative system without leaving the injector's fingerprints all over the resulting artifacts. We leave this as a challenge for the future.

Since the knowledge-based model builds on the filtering model, both intentional novelty and intentional value can be

¹³Again created by the author by making use of synonymy and other relational semantics, metaphor, and grammatical variation to modify the tree haiku

Algorithm 7 Creative generation by modeling an inspiring set I and leveraging a knowledge base K followed by evaluating the perception of the generated artifact.

```
Create( $I$ ,  $K$ )
  model = build_model( $I$ )
  while score <  $\theta$  do
    a = generate(model,  $K$ )
    score = evaluate(perceive(a))
  return a
```

found here as well, and, making use of the additional knowledge now available, that intention can be more nuanced, resulting in a concomitant increase in value (and possibly in novelty as well).

Creation

The final stop on our journey abstracts the system's evaluation mechanism and introduces perceptual ability, as shown in Algorithm 7. This new ability means the system can ground concepts perceptually, giving it at least a rudimentary ability to understand the world. Leveraging this understanding leads to additional improvement in results. The most obvious perceptual abilities that might be incorporated into such a system include vision, audition, chemical analysis (smell/taste) and touch. With these, a system can look at the haiku as well as listen to it being read aloud, allowing the evaluation of factors such as visual appearance, prosody, (both visual and aural) flow, etc.

Just these basic perceptual abilities have the potential to significantly improve results, but there is no reason that computational systems need be limited to just these. Additional derived and invented perceptual capabilities can be conceived, including other types of signal processing (radiation, atmospheric pressure, network flow), and abstract percepts such as the detection of affective and social cues, etc.

Here is a nice English haiku¹⁴ that cleverly plays on Bashō's famous poem and that could notionally be created by such a system:

*By an ancient pond
a bullfrog sits on a rock
waiting for Bashō*

Intentional novelty and value are featured here as well, but with the advantage that the intentionality is now perceptually grounded. The benefit of this should be evident: grounding allows the possibility of natural cross-domain creativity (write a haiku that describes what silence looks like), and it improves the possibility of mutual comprehension (assuming shared percepts).

An intentional detour

Before ending our expedition and considering what we may have learned, we must first discuss a somewhat orthogonal but important concern about how intention might be

¹⁴Written by Scott Alexander

Algorithm 8 *Creating haiku through random generation filtered by a fitness function, $\text{Fitness}()$, which returns a score that is computed as a convex combination of feature values that measure the goodness of the artifact along the characteristic dimensions of syllable count, line count, theme and semantics.*

```

Haiku()
  while score <  $\theta$  do
    a = generate()
    score = Fitness(a)
  return a
Fitness(a)
  y = syllable_count(a)
  l = line_count(a)
  t = theme(a)
  s = semantics(a)
  return  $\alpha_y y + \alpha_l l + \alpha_t t + \alpha_s s$ 

```

“located” in a CC system. Consider the approaches of Algorithms 8 and 9 for creating haiku. The first is a pure generate and test procedure, albeit with a (postulated) sophisticated test mechanism. The second is an iterative, controlled generative procedure¹⁵. In what ways do they differ? One difference might be temporal, as Algorithm 8 may take significant time to produce an artifact whose fitness is above threshold. On the other hand, this approach may be capable of generating haiku that Algorithm 9 can not, because its generation process is not limited in any way. However, given enough time for Algorithm 8 and enough breadth of theme and vocabulary for Algorithm 9, one might argue that they are equivalent in their potential observable behavior (that is, in the set of artifacts that they can generate). Further, both approaches employ the same domain knowledge about haiku (structure, theme, semantics, etc.).

The real difference between the two is in *where* that knowledge is leveraged. In Algorithm 8, the knowledge is used as a *post hoc* filtering mechanism. In Algorithm 9, the knowledge is used to restrict the generation process *in situ*. The question is, *are these approaches fundamentally different in their creative ability?* Also, note that the question is no closer to resolution if one considers the meta-creative case in which the system may change its domain knowledge/summative criteria through learning, interaction, environmental effects, etc.—such changes could be effected in either the fitness function or in the generative process¹⁶.

Another, related, difference between the two is in what

¹⁵Note that in both cases, the summative characteristics/domain knowledge shown (structure, thematic range, semantics) is meant to be representative only; the idea is that any and all such knowledge would be incorporated into both algorithms, either as a part of the fitness function or as a part of the generative process, respectively.

¹⁶How these changes might be effected is another question; at first blush, it seems that perhaps self-modification of the fitness function could be significantly easier than self-modification of the generative process, but that might be a consequence only of the way those two constructs have been rendered here.

Algorithm 9 *Creating haiku through an iterative process of first choosing a theme, then choosing theme-appropriate candidate words, then selecting some subset of those words (along with helper words) that contains 17 syllables and can be broken into three lines, then ordering the words to convey an acceptable level of semantics.*

```

Haiku()
  while not done do
    theme = choose_theme()
    wordset = find_words(theme)
    while not 17 syllables in three lines do
      words = select_words(wordset)
      while unacceptable semantics do
        a = reorder(words)
  return a

```

the system can explain: the filtering system of Algorithm 8 can explain *why* the artifact is novel and has value, but it can not give a satisfactory account of how the artifact was produced; the generative system of Algorithm 9 can to some approximation explain not only the novelty and value of its output but also the reason it was generated. Interestingly, there exist human creators of both ilks as well: those that are method-conscience and those that are not.

Where in the World are We?

The journey of generation that we have just taken is illustrated in Figure 2, with the stochastic system defining one extreme of the spectrum, while the other is left undefined. The ordering of the various approaches and the relative spac-



Figure 2: A spectrum of generative systems. The threshold beyond which our systems are no longer merely generative lies somewhere, but it is not clear where. Even more concerning, as our systems become more sophisticated and we progress farther afield, we as a community may continue to insist that the threshold is just beyond our currently charted territory. (Original artwork courtesy of Krey Ventura.)

ing between them may be debatable, and the exact placement of a particular system on the spectrum may be unclear, but the general picture is accurate to some approximation sufficient for the current discussion. Given this, we can now address two critical questions:

1. *Where is the threshold*—how far along the spectrum must one go in order to be safe from the label mere generation?
2. *Where are we*—where on the spectrum is a typical modern computationally creative system?

The edge of the world

What exactly must a system do (or what characteristics must it possess) to avoid being damned as mere generation? One might argue that the question doesn't really matter, because the expression is just meant as a catchy maxim that articulates a philosophy we as a field profess; however, the question actually matters a great deal, because we are often guilty of employing this philosophical tenet as a concrete measuring stick, with the common result that a system fails to measure up. This is problematic because we can't then concretely say why it fails to meet the standard—it fails simply because it is merely generative.

As a way to instigate a discussion on the matter, we offer several “lines in the sand” which, if achieved, could be considered sufficient to show that a system can no longer be considered merely generative:

1. it can be demonstrated to possess any knowledge whatsoever
2. it can be demonstrated to possess knowledge that it has had some hand in structuring/acquiring
3. it can be demonstrated that it has some reasonable chance of producing both novelty and value
4. it can be demonstrated that it has some reasonable chance of producing both novelty and value and at least one of these is intentional
5. it can be demonstrated that it has some reasonable chance of producing both novelty and value and both of these are intentional

These candidate thresholds are ordered by increasing level of demand, and they correspond roughly to demarcating mere generation as solely randomization up to and including generalization. A reasonable argument can likely be made for any of these, and we argue that anything more demanding will exclude real computational creativity. We further argue that the line should be drawn to be as inclusive as possible, while respecting the spirit of rejecting mere generation.

A related concern is the potential for an insidious shifting of this threshold over time—as increasingly sophisticated and accomplished systems are developed, the threshold is continuously shifted beyond their reach—not by the lay person, but by the community itself. This can be argued to be a natural consequence of and even stimulus for progress; however, it is at least as likely that the effect is instead a depression of growth—without some magical talisman unattain-

able by mere mortals, no matter how far afield we sail, we will never see the shores of Valinor¹⁷.

In the past, when the bounds of the world were not yet understood, it was not uncommon for seafarers to fear traveling into the unknown. While we as a community do not fear going where no one has gone before, it is possible that we are overly tentative about admitting that perhaps we already have. And, in fact, we have, not in the exceptional case at this point, but in the common one. Certainly, we aren't where we want to be yet, but we need to own the fact that we, at least, are not in port any longer. We have sailed beyond the edge of the map.

Triangulating our position

In other words, by any reasonable measure, we as a community (taking that term to include many researchers and systems that have not yet participated in a titular CC event and may not even be aware that the field exists) have moved *en masse* beyond the threshold of mere generation.

As an example, consider the soon-to-be-released *No Man's Sky*¹⁸ being developed by Hello Games. It is being touted as a science fiction exploration game set in a vast, open universe created entirely by procedural generation (see Figure 3). Early press has seemed positive and previews look pretty spectacular. If it is, in fact, as large as it claims to be¹⁹ and is purely procedurally generated as claimed (and how could it be anything else at that scale), and we are tempted to dismiss it as mere generation, we likely miss out on something pretty extraordinary, miss out on potentially growing our field and, what's worse, potentially risk losing our credibility.

Taking the most conservative threshold mentioned above (intentional novelty and intentional value) and being conservative in our analysis of the output potential of the various types of systems, we would require a system to have, at the least, the ability to filter artifacts. If we are more liberal in our analysis of the prototype models, or our placement of threshold, it is even easier to argue that we have made the leap, and that scoffing at mere generation has made the transition from inviolate charge to historical amusement.

The problems of the day are more complicated, as they should be, including questions like:

- how can we build computationally creative systems that are more autonomous (i.e. that have fewer of the designer's fingerprints all over them)?²⁰

¹⁷If you haven't read *The Lord of the Rings* trilogy and *The Silmarillion*, you should

¹⁸Release dates: June 21 in North America, June 22nd in Europe, and June 24th in the UK, <http://www.no-mans-sky.com/>

¹⁹ 1.8×10^{19} worlds is the latest estimate, according to <http://www.gamespot.com/articles/how-to-play-no-mans-sky-a-detailed-breakdown/1100-6435316/>

²⁰Note that the position taken here suggests that imbuing a system with greater autonomy might have the unfortunate effect of pushing it back onto the wrong side of the mere generation threshold due to a (hopefully temporary) precipitous drop in output qual-

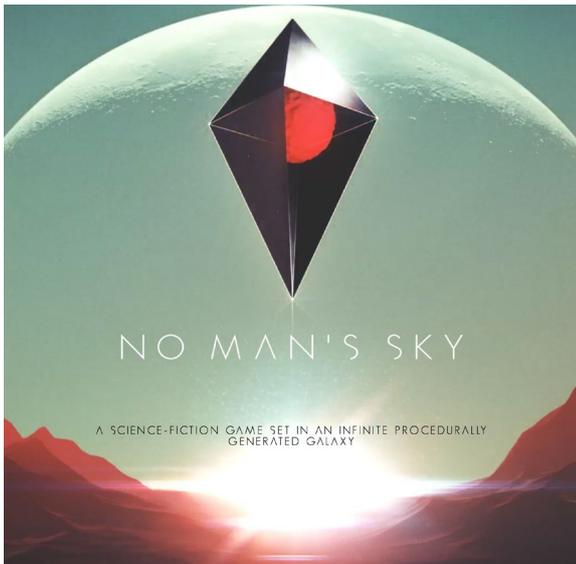


Figure 3: Can an “infinite procedurally generated galaxy” be produced by mere generation? (Image from www.no-mans-sky.com)

- how can we scale our systems to the real world (i.e. produce real artifacts of value and novelty)?
- how can we apply CC ideas to new domains (i.e. those perhaps less obviously amenable to computational approaches)?
- how can we do cross-domain (computational) creativity?

As an example apropos the last question, consider the problem of crossing humor with haiku. One way to do it, of course, is simply to write haiku with humorous themes²¹, but a more sophisticated approach might consider parodying the form itself, as demonstrated by this cleverly horrible haiku-like poem²²:

*Anything can be
a haiku if you try hard
eno - u - u - gh*

Last Words

We have argued for a need to revisit the idea of mere generation, for a need to better define what it means, that its use as a measuring stick for modern systems is outdated, and that continuing its use will be detrimental to our field.

ity. This seems consistent with a community view that esteems (intentional) novelty and value in its systems—if the change was a real advance, system behavior will eventually improve such that it surpasses the less-autonomous, and, in doing so, will easily find itself again on the right side of the mere generation threshold.

²¹Actually this has been done for centuries, at least to some extent, in the *senryū* poetic form

²²This haiku was discovered by accident at <http://shirt.woot.com/offers/haiku-3>. If there is a better attribution, it is not known.

We have considered a spectrum of generation, populated with prototypical computational creativity algorithms and have argued that these are both abstract enough and varied enough in complexity to adequately represent the breadth of approaches in our field. Using this spectrum, we’ve argued that, as a field, we have surpassed the mere generation threshold.

Yet, our field seems to be growing very slowly, for all its appeal. In particular, our flagship conference seems to be characterized by a high rate of churn, the participants a combination of a small core of regular contributors and a larger contingent of hopeful initiates that fail to persist. Some churn is to be expected, and is likely even healthy, but too high a rate is detrimental, and it is very possible that such a high rate is correlated with our continued scoffing at mere generation.

We have a very long ways to go before we find our first computational Da Vinci, O’Keeffe, Einstein, Freud, Mozart, Turing or Dickens. But, we have come a bit farther as a field than we give ourselves credit for, and in particular, we are well past the doldrums of mere generation and exploring the uncharted territory beyond. We should take care not to overstate our achievements, but at the same time, we should take equal care not to understate them either. Further, there are many systems and researchers that have, even without the benefit of our collective wisdom/disdain, managed to navigate quite a ways into the wilds themselves, and it would be judicial of us to acknowledge this and make connections with them, expanding our understanding and our field.

Acknowledgements

A tip of the hat to the anonymous reviewer who suggested the obvious-in-hindsight yet brilliant idea of making the title a haiku.

References

- Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.
- Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.
- Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.
- Ventura, D. 2008. A reductio ad absurdum experiment in sufficiency for evaluating (computational) creative systems. In *Proceedings of the 5th International Joint Workshop on Computational Creativity*, 11–19.

Searching for Surprise

Georgios N. Yannakakis and Antonios Liapis

Institute of Digital Games, University of Malta, Msida, Malta

{georgios.yannakakis; antonios.liapis}@um.edu.mt

Abstract

Inspired by the notion of *surprise* for unconventional discovery in computational creativity, we introduce a general search algorithm we name *surprise search*. Surprise search is grounded in the divergent search paradigm and is fabricated within the principles of metaheuristic (evolutionary) search. The algorithm mimics the self-surprise cognitive process of creativity and equips computational creators with the ability to search for outcomes that deviate from the algorithm's expected behavior. The predictive model of expected outcomes is based on historical trails of where the search has been and some local information about the search space. We showcase the basic steps of the algorithm via a problem solving (maze navigation) and a generative art task. What distinguishes surprise search from other forms of divergent search, such as the search for novelty, is its ability to diverge not from earlier and seen outcomes but rather from predicted and unseen points in the creative domain considered.

Introduction

The search for unconventional computational outcomes has traditionally been the core challenge of computational creativity (Boden 2004). As a response to this challenge, several notions or dimensions of creativity have been investigated, either as search heuristics or as criteria for the assessment of the creative process and its outcomes. *Value* and *novelty* have arguably been the most popular of those notions (Boden 1995; Ritchie 2007; Wiggins 2006). According to Ritchie (2007), value is the degree to which an outcome is of high quality whereas novelty is the degree to which an outcome is dissimilar to existing examples within a domain. Boden (2004) and Ritchie (2007) claim that novelty and value are the essential criteria for assessing creativity and Wiggins (2006) provides definitions for novelty and value as different features that are relevant to creativity.

According to alternative views within computational creativity, however, novelty and value are not sufficient for the discovery of unconventional outcomes (Grace et al. 2014; Kulkarni and Simon 1988). Boden (1995) argued for the distinction between novelty, unexpectedness and value. This distinction is derived from the observation that creative outputs of high value are not merely novel but also unexpected. Both outcomes from creative artwork and outcomes from creative problem solving are often attributed creativity due

to the unexpectedness they elicit to an audience of evaluators. The notion of surprise appears to be an underlying aspect of the creative process which eventually is manifested on the final creative outcome (Macedo et al. 2009; Macedo and Cardoso 2002). As novelty does not cater for the temporal aspects of discovery, it is suggested that *surprise* is included as a core assessment dimension of a generated outcome (Grace et al. 2014; Maher 2010). Further, computational processes that realize *transformational creativity* (Boden 2004) in which the creator breaks the domain's rules and leads to unconventional problem solving and highly novel yet important artifact creation are far from being achieved. Recent views on aspects of surprise, however, have been proposed as potentiators of transformational creativity (Grace and Maher 2015).

Following the perspective of a large volume of work within computational creativity (Grace and Maher 2015; Grace et al. 2014; Maher, Brady, and Fisher 2013; Macedo and Cardoso 2002; Macedo et al. 2009; Macedo and Cardoso 2001) we argue that surprise is distinct from novelty and value: an outcome can be both novel and valuable, but not necessarily surprising. While surprise is naturally geared and driven by novelty, it stems from a violation of an expectation (Maher, Brady, and Fisher 2013) rather than from a new unseen outcome. Expectation does not necessarily imply novelty; consequently, surprise can be seen as novelty in a temporal space of unseen or expected outcomes (temporal novelty), rather than in a space of existing and already seen outcomes. Studies in cognitive science suggest that humans are not only capable of self-surprise but, most importantly, that surprise is a core internal driver of creativity (Grace and Maher 2015) and a crucial component of general intelligence (Ortony and Partridge 1987). Thus, in our view, surprise constitutes a powerful drive for computational discovery as it incorporates predictions of an expected outcome that it attempts to deviate from. These predictions may be based on relationships in the solution space as well as historical trends derived from the algorithm's sampling of the domain. By modeling surprise, not only do we attempt to advance our knowledge in understanding the phenomenon but — most importantly for the purpose of this paper — we equip artificial creators with capacities to search for surprising outcomes (Macedo et al. 2009).

When it comes to computational search the dominant ap-

proach towards obtaining outcomes of high value is to ad-hoc design a function that will reward outcomes with respect to a particular *objective*. An objective function characterizes the value (or quality) of the outcome, and is used in the majority of evolutionary computation studies. However, *divergent* search beyond objectives, such as *novelty*, has proven far more efficient in a number of tasks such as robot navigation (Lehman and Stanley 2011a) and locomotion (Lehman and Stanley 2011b). Similarly, in open-ended evolution studies within artificial life (Channon 2001) it is typical to consider open-ended search for e.g. survival (Yaeger 1994; Adami, Ofria, and Collier 2000) instead of particular objectives. Given the subjective and human-centric nature of creativity, studies within computational creativity and generative systems (Boden 2004; Ritchie 2007; Wiggins 2006) have also focused on the creative capacity of search rather than accomplishing specific objectives.

In this paper we draw inspirations from the above perspectives in computational creativity, divergent search and open ended evolution and we propose the use of *surprise* as a new form of evolutionary divergent search for computational creativity. Our hypothesis is that the search for surprise (i.e. *surprise search*) is beneficial to computational creativity as it complements our search capacities with highly efficient and robust algorithms beyond the search for objectives or mere novelty. As a first step towards testing our hypothesis, we herein introduce the idea of surprise search and propose a general evolutionary algorithm that realizes it. We also provide examples of the surprise search algorithm within the domains of problem solving (for maze navigation) and generative art.

Novelty of this Paper

It is important to note that several studies have already used the notion of surprise for computational modeling (Grace et al. 2014; Grace and Maher 2015; Maher, Brady, and Fisher 2013; Macedo and Cardoso 2002; Macedo et al. 2009; Macedo and Cardoso 2001; Saunders and Gero 2004). These formulations of surprise are similar to ours as they measure a degree of deviation from expected outcomes which are predicted by a model. Macedo and Cardoso (2002; 2009; 2001) performed extensive experiments to test whether the surprise scores derived from their model of surprise match the ones rated by humans under similar circumstances. In other relevant studies, a surprise score has been used to assess the creative capacity of design outcomes (Grace et al. 2014; Maher, Brady, and Fisher 2013). To the best of our knowledge, however, no study utilizes surprise directly as a heuristic within the generative or the creative search process. The model of surprise in our proposed algorithm both drives the computational search for unexpected outcomes and can also be used to evaluate the degree of unexpectedness of an obtained solution, artwork or computational product.

Other aspects of unexpectedness such as intrinsic motivation (Oudeyer, Kaplan, and Hafner 2007; Merrick and Maher 2009) and artificial curiosity (Schmidhuber 2010; Saunders and Gero 2004) have also been modeled in the literature. The concepts of novelty within reinforcement learn-

ing research are also interlinked to the idea of surprise search (Kaplan and Hafner 2006; Oudeyer, Kaplan, and Hafner 2007). As a high-level concept, surprise (as described in this paper) matches the notion of Schmidhuber (2010) which rewards new patterns of a growing world model that a curious agent attempts to learn. As an algorithm, however, the search for surprise does not resemble artificial curiosity and intrinsic motivation as it builds upon evolutionary divergent search and is motivated by open-ended evolution rather than improving a world model.

Surprise, Novelty and Value

In this section we discuss the notion of surprise as a potential form of divergent search for computational creativity which is manifested as both unconventional problem solving and computational art. For that purpose we first draw inspiration from the literature and attempt to define surprise; we then compare it against the notions of novelty and value (Ritchie 2007) which arguably define the most popular criteria of creativity assessment for computational outcomes.

Surprise

The study of surprise has been central in neuroscience (Donchin 1981), psychology (Ekman 1992), and cognitive science (Ortony and Partridge 1987; Kulkarni and Simon 1988). In neurophysiology there has been evidence for the existence of particular event-related brain potentials that can be attributed to unexpected events and, thus, used as predictors of unexpectedness and event memorability (Donchin 1981). In psychology and emotive modeling studies, surprise defines one of Ekman's six basic emotions (Ekman 1992) that can be derived from generic and culture-independent facial expressions. While the facial expression of surprise is generic and manifested similarly across people, surprise is characterized by startle physiological responses. As a result, the classification of surprise as an emotive state has been questioned by several cognitive science studies and instead defined as a temporal-based cognitive process of the unexpected (Meyer, Reisenzein, and Schützwohl 1997; Lorini and Castelfranchi 2007), a violation of a belief (Ortony and Partridge 1987; Kulkarni and Simon 1988), or a reaction to a mismatch (Lorini and Castelfranchi 2007).

Beyond value and novelty, surprise has defined a core criterion for assessing both the creative outcomes and the creative process of a computational creator. Within studies in computational creativity, surprise (along with novelty and value) has been attributed to the creative output of a computational process in creative design and beyond (Grace et al. 2014; Maher 2010; Maher, Fisher, and others 2012; Maher, Brady, and Fisher 2013), defined as a response to novelty (Wiggins 2006) or used to model agent behavior (Macedo and Cardoso 2002; Macedo et al. 2009; Macedo and Cardoso 2001). Computational models of surprise have also been used for traffic control (Horvitz et al. 2005), for detecting surprising features in images (Itti and Baldi 2005), and for detecting interesting experiments for computational scientific discovery (Kulkarni and Simon 1988).

Variants types and taxonomies of surprise have been suggested in the literature. An important distinction is between *active* versus *passive* surprise (Ortony and Partridge 1987; Grace et al. 2014): the first being the explicit expectation that was formed actively prior to a stimulus, the latter being a mere assumption arising from earlier experience. The main overarching element of surprise across any of its taxonomies, however, is the degree to which an observation is expected. Thus, independently of the variant definitions across the disciplines that study surprise as a phenomenon, we can safely derive a general definition of surprise that satisfies the key characteristics of that notion. For the purposes of this paper, we define surprise as the *deviation from the expected* and we use the notions *surprise* and *unexpectedness* interchangeably due to their highly interwoven nature (Reisenzein 2000): unexpectedness being the approximate cognitive appraisal cause of surprise.

Inspired by the relevant literature on surprise, we view surprise for computational search as the degree to which expectations about a solution are violated through observation (Grace et al. 2014). Surprise search acts as a variant divergent search algorithm, similar to novelty search described below. While novelty search diverges from previously and currently seen outcomes, surprise search attempts to deviate from expected but unseen outcomes. Our hypothesis is that if modeled appropriately, surprise may enhance divergent search and complement or even surpass the creative capacity of traditional forms of divergent search such as novelty.

Novelty

Novelty and surprise are different notions by definition, as it is possible for a solution to be both novel and/or expected to variant degrees. Following the core principles of Lehman and Stanley (2011a) and Grace et al. (2014), novelty is defined as the degree to which an outcome is *different from prior* outcomes within a particular domain. On the other hand, surprise is the degree to which an outcome is *different from the expected* outcomes in a particular domain.

Expectations are naturally based on inference from past experiences; analogously surprise is built on the temporal model of past outcomes. Surprise is a temporal notion as expectations are temporal by nature. Prior information is required to predict what is expected; hence a *prediction of the expected* (Maher 2010; Macedo and Cardoso 2002) is a necessary component for modeling surprise computationally. By that logic, surprise can be viewed as a *temporal novelty* process. Another interesting temporal metaphor of the relationship between surprise and novelty is that the first can be viewed as the *time derivative* of the latter — e.g. position (novelty) and velocity (surprise). While novelty deviates from positions in the search space, surprise deviates from positions as predicted by a model of earlier positions; the model resembles the trajectory of search.

To exemplify the difference between the notions of novelty and surprise, we will use a simple card memory game. In this game each player is given a stack of unseen cards. Players take turns revealing one card at a time, placing them in a sequence. Players have to predict which card will be revealed next. The winner of the game is the one that correctly

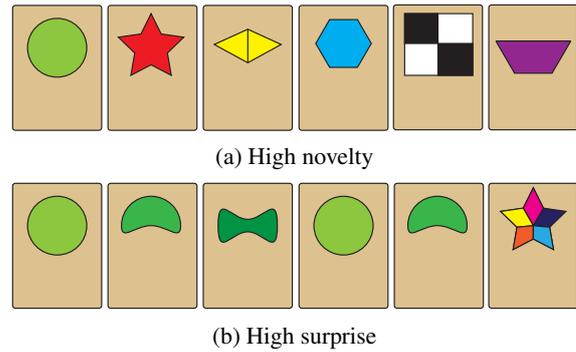


Figure 1: Illustrating the difference between novelty and surprise in a card game example. The cards are drawn in sequence, with the last one placed rightmost. The rightmost card in Fig. 1a does not share the colors or shapes of the other cards, and is thus novel; however, it is not a surprise that a completely different card is revealed (although admittedly, predicting the next revealed card would be quite difficult). The fourth and the rightmost cards of Fig. 1b are surprising, however. In the first case the player expects a new unseen card based on the three cards already revealed but, instead, the first card reappears. In the latter case, due to the repetition of previous card patterns, the player expects another green, curved shape rather than the depiction in the last card.

predicts the next card. In this example the novelty of the game outcome (i.e. next card) is the highest possible if all cards revealed in the past are different. The surprise value of the game outcome in that case is low as the player has gradually internalized a model of expectedness of a new, unseen, card every time. On the other hand, the novelty of the game outcome decreases if seen cards are revealed after a while. In that scenario surprise is increased as the game deviates from the expected outcome which calls for a new card every time. Clearly both surprise and novelty depend on the amount of cards revealed (i.e. history of instances) and the amount of cards available (i.e. how large is the domain). Figure 1 illustrates the difference between the notions of novelty and surprise in the card game example discussed above.

As a guide for evolutionary search, the concept of novelty has primarily been integrated in *novelty search*, which explicitly ignores the objective (or value) of the problem it attempts to solve. Novelty search performs divergent evolutionary search in order to handle deceptive fitness landscapes (Whitley 1991) and premature convergence to local optima. Earlier divergent search methods (e.g. (Angeline and Pollock 1994; Wessing, Preuss, and Rudolph 2013)) provide control mechanisms, modifiers or alternate objectives which complement the gradient search towards better solutions. In contrast, novelty search motivates exploration of the search space by rewarding individuals which are different without considering whether they are objectively ‘better’ than others. Novelty search is different than a random walk, however, as it explicitly provides higher rewards to more diverse solutions and also because it maintains a memory of the areas of

the search space that it has previously explored. The latter is achieved with a *novel archive* of past novel individuals, with individuals with a high novelty score being constantly added to this archive. Each individual's novelty score is the average distance from a number of closest neighbors in the problem space; neighbors can be members of the current population or the novel archive. The distance measure is problem-dependent: examples include the distance between agents' final positions in a two-dimensional maze, or the distance in the position of a robot's center of mass (Lehman and Stanley 2011a). Novelty search has also been integrated for adjusting properties of images such as brightness and symmetry (Lehman and Stanley 2012).

Value

Value has been defined as the degree to which a generated outcome is of high quality within its domain (Ritchie 2007). While in computational art and aesthetics value is largely a subjective notion — which is often measured via the valued ratings of domain experts — in creative problem solving the notion of value is clearly objective. In particular, the quality of any solution or output is determined by its distance to a predetermined goal within a set of constraints imposed by the domain per se. The notion of value can be directly linked to the notion of *objective* in optimization. More specifically, within metaheuristic search the value of an evolved solution is naturally assessed by its fitness value to a given problem. While it is natural to think that measuring progress in terms of fitness (Goldberg and Holland 1988; Michalski, Carbonell, and Mitchell 2013) is the most appropriate approach towards finding a high-fit solution, recent findings from evolutionary divergent search (Lehman and Stanley 2011a; Lehman, Stanley, and Miikkulainen 2013) suggest that explicit objective (fitness) design can be detrimental to evolutionary search, e.g. when the problem is deceptive (Whitley 1991) or open-ended (e.g. in the case of autotelic creative tasks).

While as concepts surprise and novelty have common characteristics (see earlier discussion), value can be seen as an orthogonal concept in the search for good quality outcomes. Value clearly distinguishes from novelty and surprise as it is the degree of outcome quality rather than the degree to which an outcome differs from other outcomes (novelty) or the degree to which an outcome differs from expected outcomes (surprise) in its class. Value, if used as a direction for search, points to a direct assessment of the outcome's quality whereas both novelty and surprise imply an indirect and divergent way of traversing the search space for obtaining an outcome of high quality.

Since value is orthogonal to novelty or surprise, there are ways of integrating it in divergent search e.g. via constraints that accepted artifacts should have a minimum value (i.e. fitness score) while individuals satisfying these constraints can evolve towards divergence. Examples of constrained novelty search, in particular, have been proposed by Liapis, Yannakakis, and Togelius (2015) and Lehman and Stanley (2010) for problem solving tasks (level generation and maze navigation respectively), as well as Vinhas et al. (2016) for evolutionary art and Liapis et al. (2013) for novel game ob-

ject generation. Surprise search can similarly be combined with minimal value constraints, since feasible and infeasible individuals can be evolved in separate populations (Liapis, Yannakakis, and Togelius 2015) towards different goals.

The Surprise Search Algorithm

Based on the above discussion, *surprise* as a driver of evolutionary search can be summarized as a mechanism for rewarding individuals which exhibit behaviors which diverge from the *expected behaviors* of the current population based on *prior observed behaviors*. Like novelty search, surprise search operates exclusively in the behavioral (or phenotypic) space¹: both predictions and prior evolutionary trends refer to the phenotypic space (e.g. behaviors of an evolving agent or output of an artificial painter).

Two components are therefore necessary for surprise search: a *predictive model* which creates the expected behaviors based on past and current outputs, and a *deviation formula* which assesses whether (and to what degree) the actual behaviors deviate from the predicted behaviors. To a certain extent, both of these components are domain-specific and problem-dependent; this section presents certain core properties of each component, while the specific parameters can be tweaked depending on the problem at hand.

Predictive Model

There are multiple ways to predict future outcomes, from simple extrapolation to machine learning. At its core, the set of predicted outcomes \mathbf{p} is derived from the formula in Eq. (1), where m is the predictive model, h is the history (i.e. how far in the past the model has to look to estimate the future) and k is the locality (i.e. how many data points the model has to consider per generation and, as a result, how many predictions it must make).

$$\mathbf{p} = m(h, k) \quad (1)$$

History (h) refers to how far into the past the predictive model observes when making predictions into the future. At the absolute minimum, the two previous generations must be considered, in order to assess a degree of behavioral (outcome) change which can be expected in the current generation. Earlier information can also be used, by looking at previous generations further in the past, or by considering an archive of important past predictions. The latter concept is similar to the rationale of the novel archive (Lehman and Stanley 2011a) in novelty search, where the most novel individuals from past generations are stored. Deviation from behaviors expected currently can be viewed as a form of *passive* surprise (Ortony and Partridge 1987; Grace et al. 2014) as they are assumptions which have not been actively considered. Deviation from a surprise archive can be viewed as a form of *active* surprise (Ortony and Partridge 1987) in that predictions in the archive have been “entertained” (to use the term of Ortony and Partridge) in the past. We consider h to be a problem-dependent parameter for the algorithm.

¹Behavior and phenotype of e.g. an artificial evolutionary process are terms used interchangeably in this paper.

Locality (k) refers to the granularity in which the trends of past populations are observed. Locality can stretch from global (i.e. each generation predicts a single descriptive feature of the population of size P in the current generation) to individual (i.e. each individual traces its own lineage of parents, grandparents etc. and attempts to surprise itself). A parameter k determines the level of *prediction locality* which can vary from 1 (global) to P (individual) or anything in-between. The level of prediction locality (k) in the outcome space is a problem-dependent parameter that can be derived empirically.

Predictive model (m) refers to a model which can calculate a future outcome from current and past data as collected based on k and h . As noted above, any modeling approach can be used for such purposes: from a simple linear regression of points in the outcome space, to non-linear extrapolations, or machine learned models (e.g. artificial neural networks or support vector machines). Depending on the locality of the prediction (k), the model may derive a vector of expected outcomes to deviate from. Again, we consider the employed predictive model, m , to be problem-dependent.

Deviation Formula

We are primarily inspired by the calculation of novelty (Lehman and Stanley 2011a) in the design of a deviation formula for surprise. The formula in Eq. (2) calculates the surprise score of individual i as the average distance of the n closest predictions ($p_{i,j}$) made using the predictive model \mathbf{p} . The formula assumes that the more divergent an observed behavior is from predicted behaviors, the more surprising it is. Just like with novelty search, the distance metric (d_s) is domain-dependent and can affect what is considered surprising (and therefore the evolutionary search itself). The examples in the next section demonstrate optimization for divergence from the expected in maze navigation and generative art tasks; the same examples show different ways of calculating dissimilarity.

$$s(i) = \frac{1}{n} \sum_{j=0}^n d_s(i, p_{i,j}) \quad (2)$$

It should be noted that the deviation formula is purposefully simple, as it is an intuitive way in which humans consider divergence. However, there is potential in exploring different formulas, e.g. so that results which are not too similar but yet not too dissimilar are prioritized versus too dissimilar outputs which can be perceived as atypical, alien or random (Grace et al. 2014). This can be achieved in the distance function itself, or by applying a normalizing function on $d_s(i, p_{i,j})$ (e.g. a Gaussian function). Following the surprise model of Itti and Baldi (2005), d_s can alternatively be formulated as the difference between posterior and prior beliefs of an observer.

Examples of Surprise Search

To illustrate how surprise search can work (and ways in which it differs from search for value or novelty), this paper uses two test beds: a maze navigation task, and a generative art activity. These two exemplar tasks are tackled by

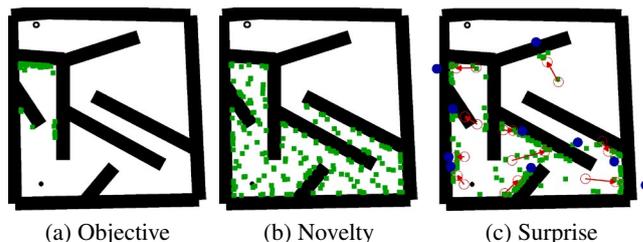


Figure 2: The process of different types of search for maze navigation on the “hard” maze of Lehman and Stanley (2011a). The solid black circle (bottom left) is the starting position, and the empty black circle (top left) is the goal; green dots represent the agents. For surprise search (Fig. 2c), red circles represent the population’s centroids of the two prior generations; the red arrow is the direction of the centroid, from the earliest generation to the latest generation, while the blue circles are the predictions for the clusters’ centroids in the current generation. Surprise search rewards the individuals in the current population (green) which deviate from the closest prediction point (blue).

evolution and constitute computational creativity domains. On one hand maze navigation focuses on *problem solving*: there is an end-state and a clear performance measure, which is whether the maze has been solved. For that purpose, we use the maze example of Lehman and Stanley (2011a) for its appropriateness in testing novelty due to the deceptiveness of the problem. On the other hand, a generative art task represents *autotelic creativity*, where there is no stopping condition and arguably no measure of success (Compton and Mateas 2015). Indeed, defining an objective for generative art would be subjective and ultimately ad-hoc; therefore we do not focus on objective-driven search for this task. We briefly present the problems, the representations used, and how the different strategies (objective-driven, novelty search, and surprise search) explore the space of each problem’s possible solutions.

Maze Navigation

In the maze navigation task, an agent controlled by an artificial neural network (ANN) enters a maze enclosed by walls at the start position of the maze, and has a specific timeframe (i.e. simulation steps) to find the goal position of the maze (see Fig. 2). The agent has 6 line trace sensors along its perimeter, measuring the distance to the nearest wall, and 4 “radars” which inform it on which side of the agent the goal is (if within range). These 10 inputs, along with a bias, are used as input to an ANN, which outputs the change in speed and the change in direction of the agent (2 outputs). The ANN is evolved using neuroevolution of augmenting topologies (NEAT) which adds complexity to initially simple networks during the course of evolution (Stanley and Miikkulainen 2002). A population of 500 ANN-controlled agents is tested in every generation: their final position at the end of the allocated timeframe in one generation can be seen in Fig. 2. This paper discusses the general princi-

ples of the search process, highlighting the differences between surprise, novelty and objective search; Gravina et al. (2016) provide an in-depth analysis of the differences in efficiency and robustness between the three algorithms in the maze navigation task.

When using objective-driven search, Lehman and Stanley (2011a) calculate the fitness of each individual based on how close it is to the goal. This is a “reasonable” performance metric, which however falls short as it does not consider walls and can cause individuals to get stuck in the dead-end at the top left corner of the maze in Fig. 2a, which acts as a local optimum. In order to find the global optimum and solve the maze, the agents must explore areas of the maze with the lowest fitness (i.e. the bottom right corner); therein lies the deceptiveness of the problem under the current objective function.

When using novelty search, Lehman and Stanley (2011a) calculate the fitness (or *novelty score*) of each individual based on its final position, and its distance from the closest final positions of other individuals in the population or in a novel archive. This drives individuals to explore more of the space, and separate themselves from current and past discovered locations. This process eventually pushes some of the most novel individuals to the goal (see Fig. 2b).

When using surprise search, instead, we suggest grouping individuals into k clusters; each predicted point is calculated based on the linear interpolation of a cluster in the population of the two previous generations (see Fig. 2c). In other words, the prediction locality of surprise search in this problem is determined by the number of clusters (k) chosen (k is 10 in this example), h involves two subsequent generations of the population and m is a linear regression function. The fitness (or *surprise score*; s in Eq. (2)) of individuals in the current population is calculated based on the deviation (d_s is Euclidean distance) of each individual from the closest predicted point. This rewards agents who diverge from an expected behavior. Note that while novelty search deviates from points of the maze that have been previously explored, surprise search deviates from predicted points which may have not been reached yet by the agents, or may never will (such as points outside of the maze in Fig. 2c).

Generative Art

In the generative art example, colorful images are generated via evolving compositional pattern-producing neural networks (CPPNs). These neural networks can have different activation functions (e.g. sigmoid or Gaussian curves) which produce symmetries and repetitions in the output (Stanley 2006). In that regard, the CPPN-based artwork is similar to the output of *PicBreeder* (Secretan et al. 2011), although this example uses a simplified representation and evolutionary strategy. Each pixel in the colored image is represented as a HSB (hue, saturation, brightness) triplet, and the CPPN produces these three output values using the x, y coordinates of the pixel as input. Fig. 3 shows how an outcome (Fig. 3a) produces a mutated offspring in the next generation (Fig. 3b). In this example, we predict one expected outcome per individual in the population, taking into account their own parent (i.e. we predict based on genotypic history,

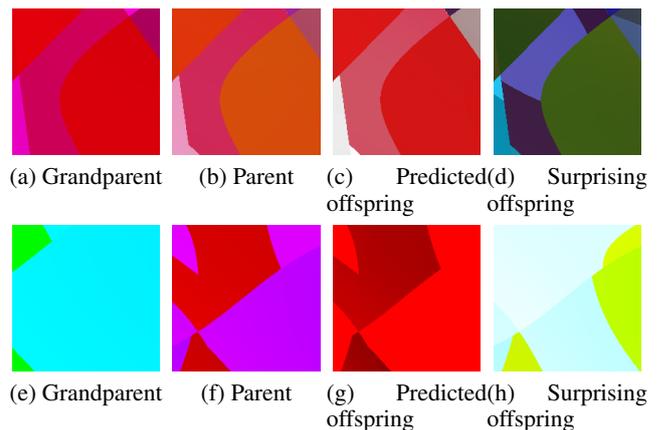


Figure 3: Two examples of surprise search in generative art, illustrating how surprise and predictions can be computed. Images depict potential outcomes of the process.

rather than phenotypic similarity as in the maze example). The predicted outcome is based on the differences in HSB values between the parent and grandparent ($h = 2$) of the evaluated individual, applied on the parent. Similarly to the maze navigation surprise search setup, m is a simple linear regression model and $h = 2$; however, the k value considered here is P (where P is the size of the population), as each image has an individual prediction. Fig 3c provides an example prediction: as the bottom left area becomes lighter (Fig. 3b) compared to the earlier image (Fig. 3a), it is predicted that the same area will become even brighter in the current generation. Another example is in Fig 3g: as the entire canvas from the grandparent (Fig. 3e) to the parent (Fig. 3f) shifts the hue towards warmer colors, it is predicted that the offspring canvas will consist entirely of red colors; moreover, the left-most darker region in the parent’s canvas is expected to become even darker in the offspring’s canvas.

A surprising outcome, in these examples, can be measured based on the per-pixel difference (e.g. d_s can be the Euclidean distance) in hue, saturation and brightness between the predicted outcome and the actual offspring. For instance, Fig. 3d has a high surprise score since the hues of the entire image have shifted to greens and blues, and the image is overall darker. Similarly, Fig. 3h is surprising since the image is overall lighter and shifted hues towards colder colors. It should be noted that Fig. 3h is similar to Fig. 3e; as with the card game example, the novelty score of such an image would not necessarily be high as the previous outcome (which likely is stored in the novel archive) is visually close to the evaluated outcome. The surprise score, on the other hand, takes into account the “trajectory” of evolution and predicts the expected outcome assuming a direction from previous to current outcomes.

Discussion and Conclusions

This paper introduced the notion of surprise for computational search, provided a general algorithm that follows the

principles of searching for surprise and presented two examples implementing the core idea of *deviation from expected*: a maze navigation problem and a generative visual art task. We argue that surprise search may show advantages over other forms of evolutionary divergent search such as novelty search. Based on the advantages of novelty over objective search, we can safely assume that a divergent search-based algorithm like surprise will manage to outperform traditional fitness-based evolution (i.e. objective search) in highly deceptive problems. Our hypothesis is that, similarly to novelty search, deviation from expected outcomes in the search space may result in higher exploratory capacity and diversity; both of which are beneficial properties for computational (evolutionary) search.

Surprise search operates similarly to novelty search with respect to evolutionary dynamics. As surprise search makes predictions for the current generation based on a set of observed behaviors in prior generations, it maintains a temporal window of where search has been. However, surprise search operates differently to novelty search with respect to the goal: surprise maximizes deviation from the expected outcomes whereas novelty maximizes deviation from previous and current outcomes. This evidently creates a new form of divergent search that considers prior behaviors *indirectly* to make predictions to deviate from. The comparative envisaged advantages of surprise search over other forms of divergent search are inherent to the way the algorithm searches, attempting to deviate from predicted *unseen* behaviors instead of prior *seen* behaviors.

As surprise search ignores objectives, a concern could be whether it is merely a version of random walk. Surprise search is not a random walk as it explicitly maximizes unexpectedness. Surprise search allows for a temporal archive of outcomes that accumulates a record of earlier positions in the problem space. Gravina, Liapis, and Yannakakis (2016) compared the behavior of surprise search versus random search (random fitness values) in the maze navigation experiment, demonstrating the differences in both performance and behavior between the two.

This position paper introduced a general form of the surprise search algorithm and examples of its implementation; extensive empirical studies need to be performed to provide evidence for the advantages of surprise as a form of divergent search. The two examples used in this paper are indicative types of test beds for surprise search. For problem solving tasks (such as maze navigation), the algorithm's effectiveness needs to be tested through tasks of varying degrees of deception and complexity. Initial experiments with surprise search in the maze navigation domain indicate that it is as efficient as novelty search, and tends to find solutions faster and more often than both traditional objective and novelty search (Gravina, Liapis, and Yannakakis 2016). For computational art, the algorithm's expressivity and creative capacity can be assessed, based on its ability to deviate from expected outcomes or based on creativity assessment models such as FACE or IDEA (Pease and Colton 2011). Finally, the surprise score introduced can be used to complement any computational creativity assessment method considered.

Acknowledgments

This work has been supported in part by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665).

References

- Adami, C.; Ofria, C.; and Collier, T. C. 2000. Evolution of biological complexity. *Proceedings of the National Academy of Sciences* 97(9).
- Angeline, P. J., and Pollack, J. B. 1994. Competitive environments evolve better solutions for complex tasks. In *Proceedings of the International Conference on Genetic Algorithms*.
- Boden, M. 1995. Creativity and unpredictability. *Constructions of the Mind: Artificial Intelligence and the Humanities. Stanford Electronic Humanities Review* 4(2).
- Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. Routledge.
- Channon, A. 2001. Passing the alife test: Activity statistics classify evolution in geb as unbounded. In *Advances in Artificial Life*. Springer.
- Compton, K., and Mateas, M. 2015. Casual creators. In *Proceedings of the International Conference on Computational Creativity*.
- Donchin, E. 1981. Surprise! surprise? *Psychophysiology* 18(5):493–513.
- Ekman, P. 1992. An argument for basic emotions. *Cognition & emotion* 6(3-4).
- Goldberg, D. E., and Holland, J. H. 1988. Genetic algorithms and machine learning. *Machine learning* 3(2).
- Grace, K., and Maher, M. L. 2015. Specific curiosity as a cause and consequence of transformational creativity. *Proceedings of the International Conference on Computational Creativity June*.
- Grace, K.; Maher, M. L.; Fisher, D.; and Brady, K. 2014. Modeling expectation for evaluating surprise in design creativity. In *Design Computing and Cognition*.
- Gravina, D.; Liapis, A.; and Yannakakis, G. N. 2016. Surprise search: Beyond objectives and novelty. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM.
- Horvitz, E. J.; Apacible, J.; Sarin, R.; and Liao, L. 2005. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. In *Proceedings of the 2005 Conference on Uncertainty and Artificial Intelligence*. AUA Press.
- Itti, L., and Baldi, P. F. 2005. Bayesian surprise attracts human attention. In *Advances in neural information processing systems*, 547–554.
- Kaplan, F., and Hafner, V. V. 2006. Information-theoretic framework for unsupervised activity classification. *Advanced Robotics* 20(10).
- Kulkarni, D., and Simon, H. A. 1988. The processes of scientific discovery: The strategy of experimentation. *Cognitive science* 12(2):139–175.

- Lehman, J., and Stanley, K. O. 2010. Revising the evolutionary computation abstraction: Minimal criteria novelty search. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Lehman, J., and Stanley, K. O. 2011a. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation* 19(2).
- Lehman, J., and Stanley, K. O. 2011b. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Lehman, J., and Stanley, K. O. 2012. Beyond open-endedness: Quantifying impressiveness. In *Proceedings of the International Conference on Artificial Life*.
- Lehman, J.; Stanley, K. O.; and Miikkulainen, R. 2013. Effective diversity maintenance in deceptive domains. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Liapis, A.; Martínez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the International Conference on Computational Creativity*.
- Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2015. Constrained novelty search: A study on game content generation. *Evolutionary Computation* 23(1):101–129.
- Lorini, E., and Castelfranchi, C. 2007. The cognitive structure of surprise: looking for basic principles. *Topoi* 26(1).
- Macedo, L., and Cardoso, A. 2001. Modeling forms of surprise in an artificial agent. In *Proceedings of the annual Conference of the Cognitive Science Society*.
- Macedo, L., and Cardoso, A. 2002. Assessing creativity: the importance of unexpected novelty. *Structure* 1(C2):C3.
- Macedo, L.; Cardoso, A.; Reizenzein, R.; Lorini, E.; and Castelfranchi, C. 2009. Artificial surprise. *Handbook of research on synthetic emotions and sociable robotics: New applications in affective computing and artificial intelligence* 267–291.
- Maher, M. L.; Brady, K.; and Fisher, D. H. 2013. Computational models of surprise in evaluating creative design. In *Proceedings of the fourth international conference on computational creativity*.
- Maher, M. L.; Fisher, D. H.; et al. 2012. Using AI to evaluate creative designs. In *2nd international conference on design creativity, Glasgow, UK*, 45–54.
- Maher, M. L. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design*.
- Merrick, K. E., and Maher, M. L. 2009. *Motivated reinforcement learning: curious characters for multiuser games*. Springer Science & Business Media.
- Meyer, W.-U.; Reizenzein, R.; and Schützwohl, A. 1997. Toward a process analysis of emotions: The case of surprise. *Motivation and Emotion* 21(3).
- Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M. 2013. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.
- Ortony, A., and Partridge, D. 1987. Surprisingness and expectation failure: what's the difference? In *Proceedings of the 10th international joint conference on Artificial intelligence-Volume 1*, 106–108. Morgan Kaufmann Publishers Inc.
- Oudeyer, P.-Y.; Kaplan, F.; and Hafner, V. V. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation* 11(2).
- Pease, A., and Colton, S. 2011. Computational creativity theory: Inspirations behind the face and the idea models. In *Proceedings of the Second International Conference on Computational Creativity*.
- Reizenzein, R. 2000. The subjective experience of surprise. *The message within: The role of subjective experience in social cognition and behavior* 262–279.
- Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1).
- Saunders, R., and Gero, J. S. 2004. Curious agents and situated design evaluations. *AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 18(02):153–161.
- Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2(3).
- Secretan, J.; Beato, N.; D'Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; Folsom-Kovarik, J. T.; and Stanley, K. O. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation* 19(3):373–403.
- Stanley, K. O., and Miikkulainen, R. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10(2).
- Stanley, K. O. 2006. Exploiting regularity without development. In *Proceedings of the 2006 AAAI Fall Symposium on Developmental Systems*.
- Vinhas, A.; Assuncao, F.; Correia, J.; Machado, P.; and Ekárt, A. 2016. Fitness and novelty in evolutionary art. In *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt)*. Springer.
- Wessing, S.; Preuss, M.; and Rudolph, G. 2013. Nicheing by multiobjectivization with neighbor information: Trade-offs and benefits. In *Proceedings of the Evolutionary Computation Congress*.
- Whitley, L. D. 1991. Fundamental principles of deception in genetic search. In *Foundations of Genetic Algorithms*. Morgan Kaufmann.
- Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7).
- Yaeger, L. 1994. Poly world: Life in a new context. *Proc. Artificial Life* 3.

Role of Simplicity in Creative Behaviour: The Case of the Poietic Generator

Antoine Saillenfest*, Jean-Louis Dessalles*, Olivier Auber**

* Telecom ParisTech, LTCI, Université Paris-Saclay, Paris, France

** ECCO, Free University of Brussels (VUB) & The Global Brain Institute, Brussels, Belgium
{antoine.saillenfest, jean-louis.dessalles}@telecom-paristech.fr - olivier.auber@vub.ac.be

Abstract

We propose to apply Simplicity Theory (ST) to model interest in creative situations. ST has been designed to describe and predict interest in communication. Here we use ST to derive a decision rule that we apply to a simplified version of a creative game, the Poietic Generator. The decision rule produces what can be regarded as an elementary form of creativity. This study is meant as a proof of principle. It suggests that some creative actions may be motivated by the search for unexpected simplicity.

Introduction

Can human creativity be captured by equations or algorithms? The idea seems contradictory. Most creative acts are by essence unexpected and cannot be predicted. But if unexpectedness is the hallmark of creativity, couldn't we use it as a proxy? Our hypothesis is that *creative processes should maximize unexpectedness*.

To test the hypothesis, we considered a situation that is sufficiently constrained to offer a limited range of possibilities, but that is still rich enough to give rise to creative behaviour. We used a simplified version of the "Poietic Generator" for that purpose. Our point is to offer a proof of principle by showing that a program implementing the principle of maximum unexpectedness may mimic creative behaviour.

This study relies on a formal definition of unexpectedness provided by Simplicity Theory. To be unexpected, creative acts must generate some complexity drop for an observer. This principle proves sufficient, in the constrained situation of the Poietic Generator, to produce non trivial patterns of actions that can be regarded as creative.

In what follows, we briefly introduce Simplicity Theory and the notion of unexpectedness. We then describe the simplified version of the Poietic Generator that we have been using for our experiments. We then explain how we implemented the principle of maximum unexpectedness in that game and show our results. Lastly, we discuss how this basic form of creativity can be used as a basis to analyse more complex creative behaviour.

Unexpectedness and Simplicity

Our hypothesis is that to appear creative, actions should involve unexpected aspects (Bonnardel, 2006; Maher, 2010). In some situations such as the one analysed here, the set of available actions is so limited that a good way of achieving creativity consists in adopting the following principle:

Principle of maximum unexpectedness in creativity:

Select actions
that will maximize
unexpectedness.

There are few formal definitions of surprise or unexpectedness. Schmidhuber distinguishes between (un)predictability, unexpectedness, surprise and interestingness (Schmidhuber 1997a; 1997b; 2003; 2009). For him, unpredictability implies unexpectedness, but unexpectedness does not imply surprise, which is defined with reference to expectations (Schmidhuber, 2003). He also defines interest as the time-derivative of the best compression an observer can achieve from the situation (Schmidhuber 2009). This means that interest is raised when the observer is making more sense of the current situation.

The framework of Simplicity Theory¹ (ST) also makes use of a difference in complexity. ST was introduced to explain why some events are unexpected and newsworthy (Dessalles, 2006; 2008). *Unexpectedness* is defined as the difference between expected complexity and observed complexity.

$$U = C_{exp} - C_{obs}. \quad (1)$$

The term 'complexity', also known as Kolmogorov complexity, refers to its theoretical definition, namely the size of the shortest summary. We do not consider objective complexity, which is not computable (Li & Vitányi, 1994), but a resource-bounded version of it (Buhrman *et al.*, 2002). ST introduces a difference between C_{exp} and C_{obs} . The former is generally assessed through the complexity of a *causal pro-*

¹ See www.simplicitytheory.science.

cedure, whereas the latter is free from this constraint and matches the usual definition of (resource-bounded) complexity. This difference between generation and observation is parallel (though not identical) to the difference between ‘generation’ and ‘distinction’ (Buhrman *et al.*, 2002). ST designates causal complexity by C_w . Since C_{obs} corresponds to a minimal description, it can be noted C_d . The unexpectedness of an observed event can be rewritten as:

$$U = C_w - C_d. \quad (2)$$

This definition explains why the content of a blank page is not unexpected ($C_w = C_d = 0$) and why a random binary string of size n is not unexpected either ($C_w = C_d = 2^n$). A remarkable lottery draw like 1-2-3-4-5-6, on the other hand, would appear extremely unexpected: all draws have same causal complexity, as they require the generation of 6 numbers ($C_w \approx 20$ bits) ; most of them require the enumeration of 6 numbers as well to be unambiguously determined ($C_d \approx 20$ bits), but not the consecutive draw which can be described with much less ($C_d \leq 3$) (Dessalles, 2006).

In toppling domino challenges, flicking one single domino leads millions of them to fall down. The global result is spectacular; particular moments revealing a well-known image or triggering some mechanical device such as a tiny catapult are spectacular as well. Does our definition of unexpectedness account for such effects? The intended results are certainly chosen to be mostly simple (*i.e.* they require a small amount of information to be described): all dominoes down, a well-known image revealed, a world record to break. But the highly complex causality leading to these events plays a crucial role as well. Even when the process is going on, one can measure the number of failure opportunities (any domino may fail to fall down) that make C_w quite huge. Unexpectedness, and therefore interest, comes from the contrast between both complexity values.

We tested the idea that creative acts appear all the more creative as observers are able to see them as unexpected. In other words, the end result of a creative act must be both simple and seemingly hard to reach.

The Poietic generator

To test the role of unexpectedness on creativity, we had to find a situation in which the machine may explore a limited, but still rich, gamut of actions. We also wanted to stay close to a situation of artistic creativity, where no predefined task is to be fulfilled. The Poietic Generator, created in 1986 by Olivier Auber, offered us an ideal framework.

The Poietic Generator (PG) is a game with no rule. All players see the same matrix, displayed on their screen, but they control only one cell of the matrix. In the real game (which is ongoing: anyone can connect to <http://poietic-generator.net/> and play), players have a rich control of their portion of the screen, in which they can draw coloured

shapes. In the absence of any instructions, players start creating what might look like a random pattern from a distance. But the collective tends to self-organize somewhat, with structures emerging from time to time, either locally or globally, as shown in Figure 1 (see also animated recordings at <http://tinyurl.com/pgen1>).

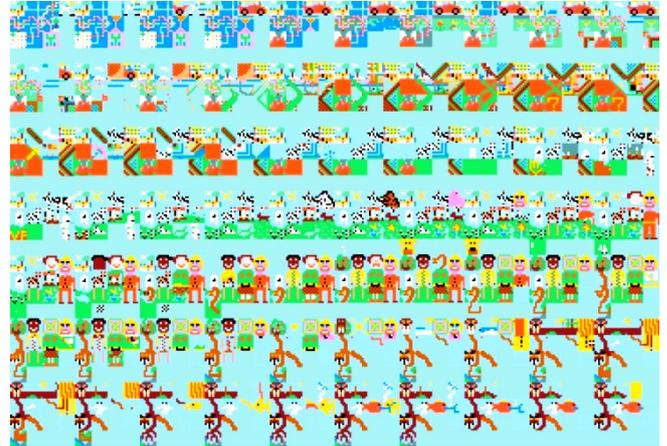


Figure 1: Successive states in a PG session observed in 1996 at Telecom ParisTech (9 participants, ~30 min).

We decided to study how programs would behave on the Poietic Generator if they followed the “principle of maximum unexpectedness.” The point is to see whether the machine appears to be creative and in what sense. This poses several challenges. First, we must define what the machine observes and how it computes generation and description complexity. Second, we must define what we would consider as ‘being creative’. And third, we should compare the productions of the machine with what humans do.

To make the three challenges manageable, we had to simplify the game significantly. In our simplified version of the Poietic Generator (SPG), each cell of the matrix consists in one pixel. Each player, as a result, can only control one among K colours. Moreover, all players are instantiations of the same decision rule. Even this way, the SPG remains rich enough to offer the opportunity of being creative. It is not easy to predict what will happen, and it is not easy to tell in advance what creative actions would be. However, it might be easier to tell after the fact that reaching such or such state was (somewhat) creative.

Coding representations

Any artificial creative device must rely on a model of aesthetic preference. In our approach, unexpectedness provides such a hierarchy. The computation of U , however, presupposes a cognitive model from which complexity can be computed, for instance a neo-Gestaltist theory in which simple patterns are group invariant (Leyton, 2006). To keep things simple, we decided to use a set of pre-computed

simple patterns that we call *basic patterns*. This is, of course, only acceptable for the SPG, and even for the SPG, for small matrix sizes.

Figure 2 shows a rudimentary set of monochrome basic patterns that we used to evaluate the SPG. Our implementation of the SPG, however, accepts coloured basic patterns. For instance, the definition of a trichromatic pattern relies on three colours (c_1, c_2, c_3) that are not instantiated. The distance $H(s_c, p)$ from a given state s_c to a pattern p counts all differing cells between s_c and p for each possible colour instantiation of the pattern and keeps the minimum value. In the monochromatic case, computing $H(s_c, p)$ amounts to taking the minimum between two Hamming distances.

		Shapes									
		None	Diagonal		Triangle		Line				
Background color	black										
	white										

Figure 2: Example of basic 5×5 monochrome pattern set.

The originality of our model is not only that it is based on the notion of complexity, but also to distinguish generation from description. Generation complexity C_w depends on players' actions. In the SPG, the minimal causal history leading from a reference state s_r to a target state s_t consists in indicating the location of each differing cell and how it should be switched. In a SPG of size $n \times n$, one needs $2 \times \log_2(n)$ bits to designate a cell in the matrix. The number of differing cells between s_r and s_t is $H(s_r, s_t)$. For each differing cell, one needs to indicate the target colour. If K is the number of available colours, we need $\log_2(K)$ bits to designate the correct colour among the K alternative decisions. The complexity of generating the transition $s_r \rightarrow s_t$ can be written as the minimum amount of information needed to transform s_r into s_t .

$$C_w(s_r \rightarrow s_t) = H(s_r, s_t) \times (2 \log_2(n) + \log_2(K)). \quad (3)$$

The decision rule consists in searching a maximally unexpected pattern. If we write $\alpha = 2 \log_2(n) + \log_2(K)$, equation (2) now reads:

$$U(s_t) = \alpha H(s_r, s_t) - C_d(s_t). \quad (4)$$

Equation (4) provides a hierarchy of attractiveness for the set of target patterns $\{s_t\}$. At the beginning of the game, s_r is set to the initial configuration of the grid. If the initial state is random, then $H(s_r, s_t)$ has roughly the same value

for all target states s_t . As a consequence, the most attractive targets are the simplest ones: all-white and all-black.

The complexity of reaching targets may however obliterate their attractiveness. ST takes this complexity into account to determine how much actions and targets are wanted (Saillenfest & Dessalles, 2014). We transpose this notion to the SPG by defining the *desirability* of a given target s_t seen from the current state s_c :

$$D(s_c, s_t) = U(s_t) - \alpha H(s_c, s_t). \quad (5)$$

The term $\alpha H(s_c, s_t)$ represents the complexity of generating s_t from the current state s_c . We can see that the most attractive states are not necessarily the most desirable ones. If we put (4) and (5) together, we get:

$$D(s_c, s_t) = \alpha H(s_r, s_t) - C_d(s_t) - \alpha H(s_c, s_t). \quad (6)$$

There is a trade-off between three terms: the difficulty of reaching the target from the reference state, its overall simplicity and the easiness of reaching it from the current state.

The distinction between the reference state and the current state is crucial here. Players are trying to produce an event that will appear unexpected to a hypothetical audience. The audience may be the community of players currently acting on the grid. It may also be anyone connected to the game just for watching in the case of the real Poietic Generator. For this audience, a pattern will constitute an event if it is unexpected as compared to the initial state, or later to subsequent reference states. In the case of the toppling dominoes, the final event: all dominoes having fallen down, is only unexpected in comparison with the initial state.

Our artificial SPG players base their strategy on a similar comparison (equation (4)). When selecting an action to perform, however, they measure the distance from the current state to tentative goals (equation (5)). They begin by selecting a mostly desirable goal s_t° .

$$s_t^\circ = \operatorname{argmax}(D(s_c, s_t)). \quad (7)$$

They change their colour only if it increases desirability, which amounts to saying that $H(s_c, s_t) > H(s_c', s_t)$, where s_c' is the state resulting from their changing colour. Using this strategy, the system is expected to converge on a simple state, not necessarily a simplest one. Once such a target is reached, the reference is set to that new state for all players: $s_r = s_t^\circ$. Due to this change, s_t° is no longer desirable, as it is no longer unexpected. The community of players starts looking for another goal that it may then reach, and so on. The emerging result is that the SPG will visit various simple states in this way. This travel through the state space in search for simplicity generates a basic form of creativity.

Implementing the SPG

The SPG is initialized as an $n \times n$ matrix, where each cell is set to white or, alternatively, assigned a random colour.

Each agent controls one cell of the matrix. It stores the initial global state of the game as its reference state. At each time step, one among the n^2 agents is randomly selected to play. This agent decides either to change the colour of its cell or do nothing, depending on the decision procedure described below. If the system reaches a state s_t that is maximally desired according to (6), then s_t becomes the new reference state for all agents.

Evaluating desirability

When an agent is selected to play, it has to decide whether to change colour or not. To do so, it evaluates the desirability of reachable target states using (6). Figure 3 illustrates how values of $\alpha H(s_c, s_t)$ are compared for different targets s_t . When the target is a multicoloured pattern p , e.g. a trichromatic pattern with variable colours (c_1, c_2, c_3), we add up the three Hamming distances computed only over pixels that are c_i -coloured in the pattern. The same computation is done over all instantiations of (c_1, c_2, c_3); the smallest result is taken as $H(s_c, p)$ and instantiates p .

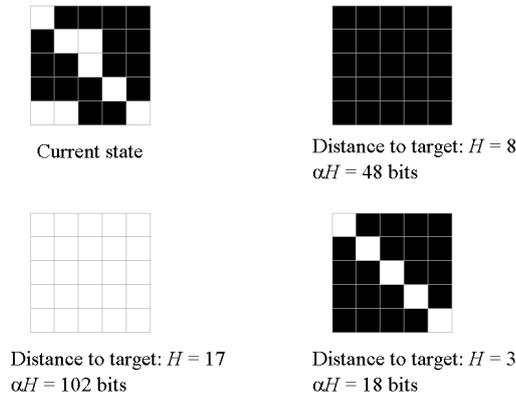


Figure 3: Distance to 3 basic states in a 5x5 monochrome SPG.

It is reasonable to consider that some target states are “too far” from the current state of the SPG to be considered by an agent as candidate target states. We introduce the notion of *horizon*, which is the value of $H(s_c, s_t)$ beyond which agents do not evaluate potential target states. In other words, a state s_t is a candidate target state only if:

$$H(s_c, s_t) \leq \text{horizon}. \quad (8)$$

To compute desirability according to (6), we need to estimate $C_d(s_t)$ for any basic state s_t (figure 2). Details are not crucial here, as long as the computation provides a reasonable hierarchy of forms. The code we chose is based on the following tuple that describes any basic state:

(colours, shape, configuration)

- *colours* is a tuple defining the pattern’s colours. For a q -chromatic pattern, $\log_2(K!/(K-q)!)$ bits are sufficient to determine the *colours* tuple unambiguously.

- We need at most $\log_2(nshape)$ bits to discriminate the pattern’s *shape* among the $nshape$ basic shapes (in the example of figure 3, only three shapes are considered: diagonals, lines and triangles).
- We need at most $\log_2(nconfig)$ bits to determine one configuration among the $nconfig$ configurations that correspond to the same shape. For example, in figure 2, there are 10 possible configurations for the shape ‘line’.

We approximate the description complexity of a basic state s_t using these upper values:

$$C_d(s_t) = C_d(\text{colours}) + C_d(\text{shape}) + C_d(\text{configuration} \mid \text{shape}).$$

We get (logarithms are approximated by their upper integer value):

$$C_d(s_t) = \log_2(K!/(K-q)!) + \log_2(nshape) + \log_2(nconfig). \quad (9)$$

Note that $nshape$ and $nconfig$ depend on s_t . Formula (9) can be used to rank basic states by simplicity (see Table 1).

Table 1: Description complexity of basic states in a monochrome 5x5 matrix with 3 possible shapes (diagonals, lines, triangles).

Basic SPG state	(colours, shape, configuration)
colours: (white, black) or (black, white) shape: None ( or )	Example of code: (all-black) [0, _, _] Length: 1 + 0 + 0 = 1 bit
colours: (white, black) or (black, white) shape: Diagonal (ex:  , )	Example of code: (white rising diagonal) [0, 00, 0] Length: 1 + 2 + 1 = 4 bits
Colours: (white, black) or (black, white) shape: Triangle (ex:  , )	Example of code: (white upper right triangle) [0, 01, 01] Length: 1 + 2 + 2 = 5 bits
Colours: (white, black) or (black, white) shape: Line (ex:  , )	Example of code: (white second horizontal line) [0, 10, 0101] Length: 1 + 2 + 4 = 7 bits

Decision procedure

Once the desirability of candidate target states has been computed by agents, two things may happen: either one candidate state (ore more) is maximally desirable, or none is desirable. When at least one state is maximally desirable, it becomes the agents’ current target. Agents change their colour only if it brings them closer to the target for the $H(s_c, s_t)$ distance. Otherwise, agents perform no action.

At the beginning of the game, desirability is the same for all candidate states s_t (equation (6)): $D(s_c, s_t) = D(s_n, s_t) = -C_d(s_t)$. The same holds when a reference state has been

reached and the new reference is taken to be $s_r = s_c$. In these situations, none of the possible target states is desirable ($D(s_r, s_t) < 0$). A rational agent should not act in the absence of goal. However, such an attitude would be counter-productive in a creative context. In the case of the SPG, the game would freeze, since all agents have the same reference state and the same horizon. We programmed agents to change colour with a certain probability in the absence of desirable state.

Results

Though we implemented SPG for an arbitrary number of colours, we evaluated it only for $K = 2$ (black and white). Our results vary somewhat depending on the value of *horizon*. Figure 4 displays the fraction of the time during which agents are able to find desirable states, as a function of *horizon*. For small values of *horizon*, agents do not “see” any target state and no state is ever desired. At the other end, with a high value of *horizon*, agents can see desired states all the time and are always in goal-oriented search. Moreover, the distribution of desired states shows that the most desired states correspond most of the time to the simplest ones (i.e. the two plain states). For some intermediate value of *horizon*, agents target a broader range of states.

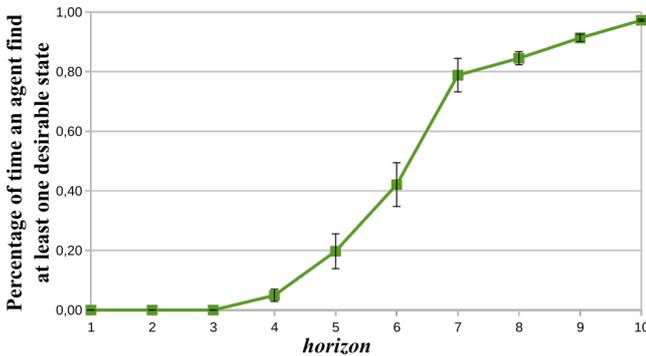


Figure 4: Fraction of time playing agents find at least one state desirable (matrix 5×5).

In the results presented here, parameters are fixed at the following values:

- size of the matrix: 5×5 , two colours, horizon: 7;
- when no state is desirable, agents change colour with probability 0.5.

When the program runs, the state of the SPG changes continuously (see <http://spg.simplicitytheory.science>). From time to time, it reaches a low-complexity state. Figure 5

shows how the complexity of the SPG matrix evolves through time. In this figure, the complexity of the current state s_c is computed in reference to the closest basic state p , by evaluating $\min_p(H(s_c, p) + C_d(p))$. We can observe its “oscillations” as it visits basic states (figure 2) and then moves away from them.

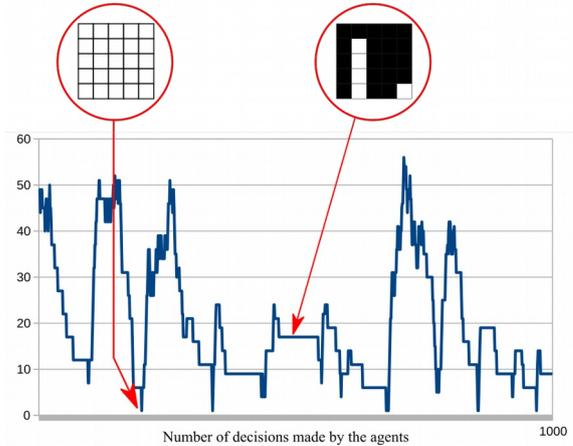


Figure 5: Evolution of the description complexity of the SPG matrix over the first 1000 individual decisions.

Figure 6 corresponds to the same run. It shows the 100 first target states that were successively reached. We can see that the system visits almost all the “basic states” described in figure 2. A simple analysis using periodograms did not reveal any regularity in this sequence.

Note that transitions are not totally random. Figure 6 reveals that when a plain state has been reached, the next basic step is likely to be a diagonal (among the 27 transitions from a plain state to another basic state, 20 lead to a diagonal state with same background). This observation makes sense. Seen from a plain state, diagonals with same background colour lie at Hamming distance 5, which is smaller than *horizon*. For most agents, changing colour would not bring them any closer. Nine of them, located on the diagonals, can get closer by one unit to a diagonal pattern, which becomes more desirable by $\alpha \approx 6$ bits (formula (6)). Since its description complexity amounts to 4 bits (Table 1), its desirability is now 2 bits. If one of those 9 agents is by chance next to play, it will change colour. The diagonal then becomes desirable to all agents. As a result, the probability that the next target will be a diagonal when starting from a plain state must be larger than $9/25 = 0.36$. We measured 0.44. Table 2 shows shape to shape transition frequencies computed after a single run of the program (the number of observed transitions from a given shape to another shape is indicated under the reference shape).

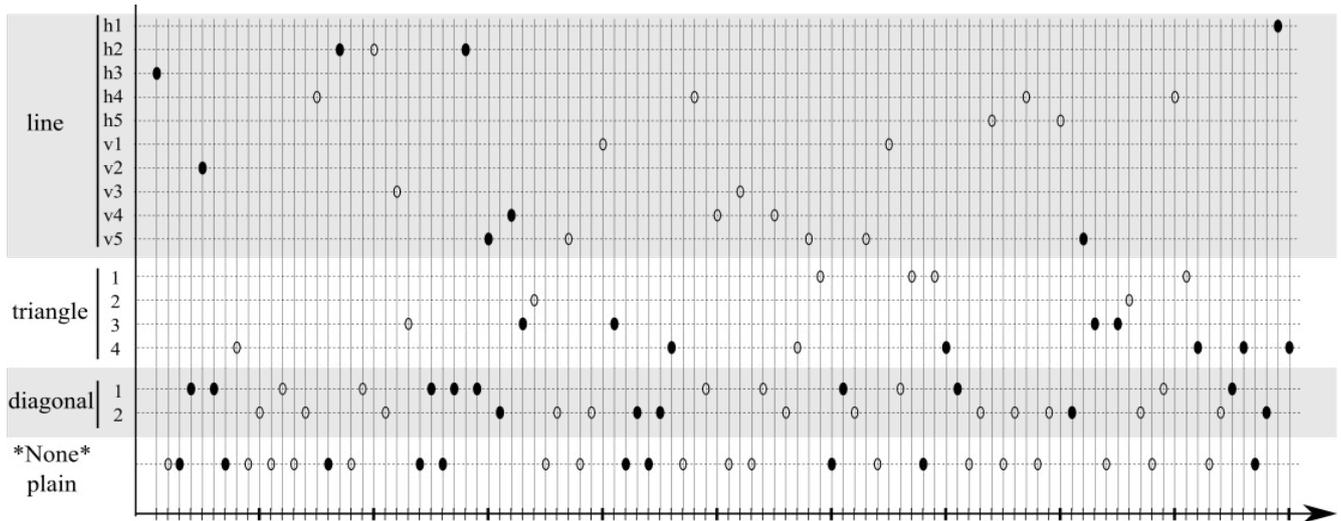


Figure 6: Visited shapes in chronological order. Dot colours (black or white) correspond to shape background.

Table 2: Frequency of transitions between shapes measured in a 5x5 matrix for the shapes of Table 1 during a single run of the program (bc=same background colour; 1-bc = opposite).

		Next shape			
		Plain	Diagonal	Triangle	Line
Shape taken as reference	Plain (1162 transitions)	0.1 bc: 0 1-bc: 0.1	0.44 bc: 0.44 1-bc: 0	0.23 bc: 0.02 1-bc: 0.21	0.24 bc: 0.24 1-bc: 0
	Diagonal (764 transitions)	0.26 bc: 0.23 1-bc: 0.03	0 bc: 0 1-bc: 0	0.12 bc: 0.06 1-bc: 0.06	0.62 bc: 0.6 1-bc: 0.02
	Triangle (999 transitions)	0.50 bc: 0.16 1-bc: 0.34	0.07 bc: 0.07 1-bc: 0	0.26 bc: 0 1-bc: 0.26	0.17 bc: 0.17 1-bc: 0
	Line (906 transitions)	0.35 bc: 0.22 1-bc: 0.13	0.22 bc: 0.21 1-bc: 0.02	0.43 bc: 0.23 1-bc: 0.20	0 bc: 0 1-bc: 0

Discussion

We showed that a simple strategy, the “principle of maximum unexpectedness”, leads to seemingly creative actions. From the definition of unexpectedness as complexity drop between generation and description (formula (2)), we derived the strategy of maximum of desirability (rules (6) and (7)) that continually looks for simple patterns within reach. This strategy, together with the notions of reference state and of horizon, is sufficient to generate interesting behaviour in the SPG. When several instances of the strategy play together the SPG, we observe an emerging behaviour which consists in visiting simple patterns in an unpredictable way (see examples at <http://spg.simplicitytheory.science>).

The walk through simple patterns is the best (in the

sense of most creative) we could get in this simple implementation, at least from a theoretical point of view. Any other emerging result among what the system could have achieved (fixed point, periodic behaviour, random walk) would have been less creative. The program mimics the following features of creativity.

- Search for unexpected simple patterns,
- Co-existence of goal-free and goal-oriented actions,
- On-going goal change,
- Fresh start when a goal is achieved.

Unexpectedly simple patterns are essential to most forms of artistic creativity. One extreme example is offered by the “White on White” painting exhibited by Kazimir Malevich in 1918. Note that further instances of so-called monochrome paintings (*i.e.* uniformly coloured surfaces) can be felt as less creative than the very first one, as they are more complex (more information is needed to discriminate them from each other). More generally, any hidden structure discovered by the observer in a painting makes it more interesting (Leyton, 2006). According to Leyton, the more circles and ellipses our eye can see in Picasso’s “Les demoiselles d’Avignon”, the more beautiful the painting appears. This makes sense within ST’s framework: hidden structure means unexpected simplicity. Any structural component contributes to simpler description, as previously independent components can now be summarized by the structure.

In contrast to routine engineering activity which is goal-driven, some artistic activities are carried out in the absence of definite goal and they are able to invent their own goals on the fly. This is what our implementation of the SPG does, despite its elementary character. This absence of pre-definite goal is perceived by observers. As predicted by (1), the interplay of seemingly random actions and oppor-

tunistic goal generation produces a series of complexity drops that may trigger feelings of beauty (Schmidhuber, 2009). Note, however, that the originality of ST is to define compression through (2) by making a distinction between generation and description.

Our implementation of the SPG has elementary self-observation capabilities. It knows when an interesting creation has been reached. Our program stops for a while when it gets to a target configuration. These moments correspond to local complexity minima. Then, by setting the reference to the former goal state, the system is able to escape from it. As equation (6) shows, the state is no longer desirable once it has become the new reference. The system automatically hunts for new simple states to spot and reach.

When no desirable state is within reach, the decision rule expressed in (6) and (7) does not apply. In such situations of goal-free action, human individuals tend to perform actions anyway, though in a biased manner (Auriol, 1999). Our simulated players are not biased and switch their state randomly. This aspect of our implementation is not governed by any principle and could be improved in more elaborate versions of the SPG.

The desirability of target states expressed by (6) is interesting because it includes two versions of generation complexity C_w . We can write it in a more general form.

$$D(s_c, s_t) = C_w(s_r \rightarrow s_t) - C_d(s_t) - C_w(s_c \rightarrow s_t). \quad (10)$$

The latter term, $C_w(s_c \rightarrow s_t)$, represents a low complexity value that was not anticipated when the system started from the reference state. A target s_t is desirable if $C_w(s_c \rightarrow s_t) \ll C_w(s_r \rightarrow s_t)$, which means that s_t is significantly easier to reach than anticipated. The same phenomenon holds in other forms of creative productions, such as fiction writing. Interest in a narrative may be aroused when some surprising event occurs, but then is perceived as making sense after all, because of some hidden line of reasoning (Saillenfest & Dessalles, 2014; Saillenfest, 2015). Rule (10) offers a similar kind of surprise when the system comes close to a simple state that was initially considered out of reach, but now appears to be just a few moves away.

Limits and perspectives

Our experiment has obvious limits. It is a proof of concept that does not aim at giving an illusion of genuine creativity. We are indeed quite far from what a human observer would regard as truly creative.

Human players in the true PG show significantly more sophisticated behaviour. They may form individual intentions based on their personal history and context; they are able to recognize many shapes and not only geometrical

ones: horses, houses or human figures; on the other hand, they have limited patience and attention span. Differences among individual players may lead to paradoxical situations, as when an idle player becomes a stable anchor around which local activity gets organized, and emerges as a local attractor for this activity.

Human players may collectively produce simple emerging patterns such as a uniform area or a checkerboard. Most emerging patterns, however, consist in recognizable shapes: a cow, a monster, a sea shore, an air strike or a luncheon on the grass. These patterns may occur in one part of the global image and may be inspired by the news.

Mimicking these human capabilities depends on the system's ability to select and recognize elaborate patterns. If the size of the matrix is increased, the set of basic patterns (lines, triangles...) becomes too sparse for the SPG to see any target within the horizon. The situation would become even more complicated if the set of possible actions is increased to bring the SPG closer to the true PG: many colours, more pixels controlled by each player. Populating the set of simple shapes may solve the sparseness problem. However, the problem of computing the complexity C_d of elaborate shapes is not a trivial task (think of recognizing an air strike and its relation to the news).

Another characteristic behaviour exhibited by human players consists in calling attention to themselves whenever possible. For instance, a player controlling a cell in the middle of a uniform region may be tempted to switch to a locally contrasting colour, as in the yin-yang (Taijitu) symbol. This makes sense within ST. The theory indeed predicts that the complexity drop that drives attention to the individual will be larger when her cell is isolated. Its minimal description will be more concise if it is a contrast with its surroundings. A way to improve our model of creativity would be to include a second complexity drop computation at the individual level, so as to allow artificial players to choose between collective creativity and individual signalling.

SPG can be seen as a first step toward a new class of cellular automata that try to mimic some aspects of human creative behavior. One possible perspective for further developments would be to design artificial agents able to play in real PG games. Human players would be asked if they are able to locate them. This experiment might be seen as a visual version of the Turing test. It could become the basis of an 'open science' initiative to study the specificity of human creative behavior. This open experiment could also help to deal with ethical questions about human-computer entanglement in a manner that would be accessible to all (Auber 2016).

All the above mentioned improvements of the SPG keep fundamental principles derived from Simplicity Theory intact. The decision rule expressed by (10) and (7) would remain essentially the same (except for the pattern distance

(3) which cannot remain based on the Hamming distance if more pixels are controlled by each player). Our little experiment with the SPG was designed to be just sufficient to implement the decision rule. This is why it is relevant to study creativity

Conclusion

This study was motivated by the observation that simplicity and complexity drop play a crucial role in creativity. We decided to investigate whether Simplicity Theory could make an interesting contribution to the understanding of creative action. ST was developed to offer a formal definition of interest in human spontaneous communication. Quite naturally, we wanted to explore whether interest in creative situations could be governed by similar mechanisms.

The present study is meant as a proof of principle. We proposed a simplified implementation of the Poietic Generator to verify that a straightforward application of ST's principles could lead to interesting behaviour even in a simplistic setting.

The "principle of maximum unexpectedness" (formula (10)) that we derived from ST makes a trade-off between three values: (1) the simplicity of the target, (2) the difficulty to reach it from the reference situation and (3) the easiness to reach it from the current situation. This decision rule is claimed to apply to a wide range of creative situations. We were able to show that these theoretical principles produce non trivial behaviour even in a simplistic situation. Our suggestion is to consider these principles when designing elaborate creative programs.

Acknowledgments

This study was supported by grants from the programme [Futur&Ruptures](#) and from the "[Chaire Modélisation des Imaginaires, Innovation et Création](#)".

References

- Auber O. (2016). [Refounding legitimacy toward aethogenesis](#). Proceedings of 18th International Research Conference in The Planetary Collegium's Series "Art & consciousness in the post-biological era". *Technoetic Arts Journal, Intellect* (in press).
- Auriol, J.-B. (1999). [Modélisation du sujet humain en situation de résolution de problème, basée sur le couplage d'un formalisme logique et d'un formalisme d'opérateur](#). Paris: Thèse de doctorat – ENST 99-E-049.
- Bonnardel, N. (2006). *Créativité et conception - Approches cognitives et ergonomiques*. Marseille: Solal.
- Buhrman, H., Fortnow, L. & Laplante, S. (2002). [Resource bounded Kolmogorov complexity revisited](#). *SIAM Journal on Computing*, 31 (3), 887-905.
- Dessalles, J.-L. (2006). [A structural model of intuitive probability](#). In D. Fum, F. Del Missier & A. Stocco (Eds.), *Proceedings of the seventh International Conference on Cognitive Modeling*, 86-91. Trieste, IT: Edizioni Goliardiche.
- Dessalles, J.-L. (2008). [La pertinence et ses origines cognitives - Nouvelles théories](#). Paris: Hermes Science.
- Leyton, M. (2006). *The structure of paintings*. New York: Springer Verlag.
- Li, M. & Vitányi, P. (1993). *An introduction to Kolmogorov complexity and its applications (3rd ed.)*. New York: Springer Verlag, ed. 1997.
- Maher, M. L. (2010). [Evaluating creativity in humans, computers, and collectively intelligent systems](#). In T. Taura & Y. Nagai (Eds.), *Design creativity*, 41-47. London: Springer.
- Saillenfest, A. & Dessalles, J.-L. (2014). [Can Believable Characters Act Unexpectedly?](#) *Literary & Linguistic Computing*, 29 (4), 606-620.
- Saillenfest, A. (2015). [Modélisation Cognitive de la Pertinence Narrative en vue de l'Évaluation et de la Génération de Récits](#). Paris: Thèse de doctorat -2015-ENST-0073, ed. 2015.
- Schmidhuber, J. (1997a). [Low complexity art](#). *Journal of the International Society for the Arts, Sciences, and Technology*, 30 (2), 97-103.
- Schmidhuber, J. (1997b). [What's interesting?](#) Lugano, CH: Technical Report IDSIA-35-97.
- Schmidhuber, J. (2003). [Exploring the predictable](#). In T. Taura & Y. Nagai (Eds.), *Design creativity*, 579-612. London: Springer.
- Schmidhuber, J. (2009). [Simple Algorithmic Theory of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes](#). *Journal of SICE*, 48 (1), 21-32.

EVALUATION



Investigating Listener Bias Against Musical Metacreativity

Philippe Pasquier, Adam Burnett,
Nicolas Gonzalez Thomas

School of Interactive
Art + Technology
Simon Fraser University
philippe_pasquier@sfu.ca

James B. Maxwell
Arne Eigenfeldt

School for the
Contemporary Arts
Simon Fraser University
jbmaxwel@sfu.ca

Tom Loughin

Department of Statistics and
Actuarial Science
Simon Fraser University
tloughin@sfu.ca

Abstract

We present an empirical study investigating the hypothesis that listeners hold a bias against computer-composed music. Presented in part as a replication study, the proposed methodology seeks to improve upon weaknesses found in previous studies of the subject. Across two study periods, with approximately 60 subjects each, we failed to find evidence of a significant bias against computer-composed music. We outline potential weaknesses in our design, and propose improvements for future studies.

Introduction: Computational Creativity

A subfield of Artificial Intelligence (AI) that has recently gained significant momentum is the exploration of *computational creativity*. Pasquier et al. define this as “the science of machines addressing creative tasks” (Pasquier et al., 2016). Musical Metacreation (MuMe) is a subfield of computational creativity that addresses specifically musical tasks, like improvisation, composition, and performance. MuMe systems can be classified along a continuum according to their relative levels of autonomy, from completely user-dependent computer music “tools,” to autonomous generative systems (Pasquier et al., 2016).

As an extension of AI, computational creativity presents a new set of philosophical difficulties. In the case of general intelligence, reasoning processes can often be understood through the *a posteriori* analysis of solutions—i.e., a kind of reverse engineering of thought. This phenomenon was noted by Minsky, who pointed out that once we understand how something is done, we no longer regard it as particularly intelligent, but instead see it as a straightforward, mechanical process (Minsky, 1982). Creative products, on the other hand—and perhaps more specifically the products of artistic creativity that we address here—do not necessarily obviate the processes from which they arise. In creative “reasoning,” singularly optimal solutions seldom exist, and detailed knowledge of the technical means involved in producing a given solution cannot always account for the appropriateness of the method chosen.

Evaluation of the products of human artistic creativity is fundamentally subjective. In the field of music, the quality of works is evaluated by the artists themselves, by their peers (fellow composers and musicians), audiences (inferred from concert attendance and album sales), the media (published reviews of critics and journalists), programming requests from ensembles and presenters, and in some cases

by the peer-review committees organized by public funding bodies and arts councils. In the academic world, research is likewise evaluated by the author and her/his peers, by acceptance at academic conferences and the receipt of grants and scholarships, but also via methods that strive for greater scientific objectivity; i.e., the measurement of formalized input/output relationships and empirical studies which record the responses of participants.

Scientific experiments seek to infer the presence of the investigated phenomena by contrasting an experimental manipulation with a baseline control. For metacreations, this control is comprised of human-created artistic works. When dealing with music composition, however, developing the control faces two significant challenges: 1) The problem of selecting a representative work which will contrast against the computer-composed work, while mitigating the introduction of confounding variables, and 2) The interpretational problem of presenting the music to listeners in a way that accurately communicates its essential qualities, while again limiting the introduction of confounding variables.

A Bias Against Musical Metacreation?

Evidence of the bias against computational creativity is perhaps best illustrated anecdotally. When David Cope debuted “Emmy”—Experiments in Musical Intelligence (Cope, 1996)—before a live audience, her compositions were reportedly panned by a critic weeks before the actual concert (Blitstein, 2010). Emmy was not a human composer, but rather a computer program developed by Cope to emulate the styles of famous composers. Poor reviews were not the only obstruction Emmy faced: audiences often reacted with anger to her works, record companies declined to sign recording contracts, and musicians refused to perform her music (Cope, 2004).

Undeterred by this negative response—some of which he described as “racist” for its anti-machine hostility—Cope developed the software further (Blitstein, 2010). This work culminated in a successor to Emmy, called “Emily Howell,” which turned away from style emulation, focusing instead on the creation of novel works in a unique musical style. Some reviewers complained that Emily’s works, though musically pleasing, were hollow, shallow, and lacking depth and heart (Cope, 2004). Cope lamented the bias of his critics, and their complaints that he was taking away one of the remaining things humans could claim was uniquely their own: *creativity* (Cheng, 2009).

Defining “Bias”

For the purposes of the current investigation, it is important to be clear about our definition of “bias.” Explicit biases are consciously held attitudes or preferences, that can be directly reported by individuals. Implicit biases—also referred to as “cognitive” biases (Tversky and Kahneman, 1974)—on the other hand, are unconsciously held beliefs or attitudes that are not directly accessible to subjects, and thus influence behaviours and choices without the individual’s awareness. A typical example might be a CEO who verbally states (and may consciously *believe*) that applicant gender should not influence hiring strategy, yet whose hiring patterns show a strong gender preference.

Through this study we seek to experimentally determine whether there is, or is not, a bias against the notion of computational creativity in music. We do not address explicit/implicit bias directly, but rather attempt to determine whether a general bias exists, such that knowledge of authorship alone—human or computer—is enough to significantly modify preference.

Replication Studies

Reproducibility is an essential principle of scientific research. Statistical convention considers significant results to be those that are unlikely to be attributable to sampling error alone—e.g., the difference between the true population mean and the mean of the sample being tested. The significance value (p) is used to indicate whether the observed effect may be attributed to sampling error alone, such that “significant” results indicate that sampling error should account for the effect $< p * 100\%$ of the time. However, where the responses of human subjects are involved, explicit test methodology, experimental design, and subject selection comprise a complex web of influences and interactions, such that it isn’t always clear that the effect being observed is, in fact, the effect under investigation. This phenomenon is called “confounding,” and it plagues observational studies especially, but also experiments with inadequate controls. Thus, while statistical testing may suggest that an effect is unlikely to be due to sampling error alone, it cannot rule out alternative influences as potential causes. Replication studies help ensure that the results found in one study can be reproduced at a later date, thereby building confidence in the verity of the observed result.

An example of the importance of reproducibility occurred in 2011, when social psychologist Daryl Bem published a study demonstrating so-called “psi” abilities in the *Journal of Personality and Social Psychology* (Bem, 2011). In the study, participants appeared more likely to recall words from a word list if they practiced typing out those words *at a later date*—i.e., they seemed to demonstrate a kind of “premonition” of the *future* rehearsal process that could improve their recall scores in the present. Perhaps unsurprisingly, these results failed to be replicated by subsequent studies (Young, 2012), rekindling conversation about the importance of replicability in the sciences.

Aside from deliberate manipulation, fraud, and statistical chance—such as that due to type-I error-rate (i.e., “false-positive”) inflation arising from multiple comparisons (Garcia-Marques and Azevedo, 1995)—there are a number of reasons for this decline in replication studies. Broadly speaking in the contemporary research milieu, a market-

driven mentality has encouraged an over-emphasis on novel, exciting, and often counter-intuitive results, consequently discouraging both the submission and publication of articles with negative findings; a phenomenon known as the “file drawer effect” (Rosenthal, 1979). In some cases, of course, replication studies that fail to obtain the significant results of an original study do so as a result of methodological imprecision. Some argue, however, that such “conceptual” replications are preferable, as they interrogate the generalizability of the phenomenon being studied (Young, 2012)—i.e., by reframing the intent of the study independently of its precise methodology. Another possibility is whether there has simply been a change in the cultural zeitgeist; i.e., is difficulty replicating behaviour data from past decades attributable to a genuine decline in the studied behaviour, thus indicating a shift in attitudinal norms? For instance, we would not consider the data acquired from an early 20th century study of Western attitudes toward homosexuality to be representative of the population today, and the inferences one could draw from such data would be similarly inapplicable.

The Replicated Study: Moffatt and Kelly

Empirically, (Moffatt and Kelly, 2006) studied the proposed bias against computational creativity in the context of computer-composed music. In this study, participants listened to six one-minute musical excerpts, half of them human-composed and the other half computer-composed. The pieces were in three different “styles”: “free-form jazz”, “strings”, and “Bach.” To minimize effects from participants’ personal preferences, the pieces were presented in three pairs: one human- and one computer-composed for each style. The selection of pieces was determined by their “surface similarity”¹ in an effort to conceal their authorship (Moffatt and Kelly, 2006).

A group of 20 participants were divided into “musician” and “non-musician” groups, based on their level of formal music training and/or experience. Participants listened to each of the compositions and indicated how much they “liked” the composition, on a 5-point Likert scale, and whether they thought it was human- or computer-composed. After this initial round of listening and evaluation, the origins of the pieces were revealed and the participants were asked to evaluate the pieces again. In the second round, the questions were disguised so as to not alert participants to the purpose of the study—in this case asking them how willing they would be to buy, download, or recommend the compositions to someone, and how much they “enjoyed” each piece.

The experimenters noted that participants appeared to demonstrate a prejudice against computer-composed music, generally preferring those pieces that they believed (by their own judgement) to be human-composed. The experimenters dubiously called this the “free prejudice effect”: participants were “prejudiced” in favour of pieces they freely decided were human-composed. However, the experimenters did not find any overt prejudice: there were no statistically significant drops in the evaluations of the computer-composed pieces upon revelation.

They also found that participants were able to identify the

¹Details regarding their definition of surface similarity—i.e., the particular musical features considered—are not provided.

computer-composed pieces as computer-composed, and that non-musicians out-performed musicians at this task. Non-musicians, however, were not statistically successful at identifying human-composed music as human-composed. Participants, both musician and non-musician, also preferred human-composed pieces over computer-composed pieces, regardless of what they guessed their authorship to be, with musicians preferring the human-composed pieces to a greater degree. It is worth noting, however, that Moffat and Kelly draw several unsubstantiated and ill-informed conclusions from their data; an ethical problem we seek to avoid in the present study.

Burnett, Khor, Pasquier, and Eigenfeldt

In a previous study, Burnett et al. (Burnett et al., 2012) addressed similar questions in their evaluation of a system for generating harmonic progressions (Eigenfeldt and Pasquier, 2010). This experiment had certain methodological aspects in common with Moffat and Kelly, dividing participants into musician and non-musician groups, and using a Turing Test-like paradigm to determine whether participants could discriminate between human- and computer-composed musical excerpts. Whereas the Moffat and Kelly study used deception—concealing the purpose of the study from participants—Burnett et al. explicitly informed participants that they would be listening to a mix of human- and computer-generated harmonic progressions and that they would be asked to identify the source of each (thus mirroring the general objective of the original Turing Test). Additionally, Burnett et al. sought to estimate the confidence of participant responses through the use of a 4-point Likert scale: 1) Definitely Human, 2) Probably Human, 3) Probably Computer, and 4) Definitely Computer.

The findings of Burnett et al. echoed those of the earlier study in a variety of ways. For example, like Moffat and Kelly, Burnett et al. discovered that non-musicians outperformed musicians at discriminating between the human- and computer-composed excerpts. However, whereas Moffat and Kelly found that participants more easily identified the origin of the computer-composed works, Burnett et al. found the opposite; participants struggled to identify computer-composed pieces as computer-composed, but generally succeeded at identifying human-composed pieces. With regard to the measure of participant confidence, no significant differences were found between musicians and non-musicians, but participants were generally more confident in their evaluations of pieces that were human-composed. However, it is difficult to make direct comparisons between the studies, given these divergent results, as aesthetic and stylistic differences between the musical materials used in each study throw into question the influence of authorship on listener evaluations.

It is worth noting that both Moffat and Kelly and Burnett et al. received feedback indicating that participants attempted to “outsmart” the experimenters, listening for clues that would reveal the true authorship of the excerpts. In both cases, it was assumed that this effort mislead them into giving incorrect responses.

Experimental Methodology

With the possible exception of (Moffat and Kelly, 2006), there is a lack experimental data corroborating the presence

of a bias against computational creativity in music. Here we describe an experimental attempt to determine whether such a bias exists. This experiment is, in part, a replication study, but it also attempts to improve upon previous studies, taking into account the deficiencies of both Burnett et al. and Moffat and Kelly.

As indicated by participant comments, these previous studies encountered difficulties with participants trying to outsmart the experimenters by listening for “clues” of authorship—a problem we attempt to reconcile with this new procedure. Douglas Hofstadter noted that the potential for a bias against machine creativity might make it necessary to purposely deceive listeners as to the origins of a piece of music (Cope, 2004). The employment of deception in the current experiment has been designed to determine whether this speculation was correct. Efforts have also been made to reduce any practice effects stemming from the non-randomized presentation of the musical excerpts; a problem that affected previous studies. Familiarity and listening fatigue effects were also minimized by including a control evaluation, which allowed us to track changes in participant evaluations in the absence of any experimental manipulation.

To address problems of music selection, we attempted to reduce the perceptual differences between musical pieces by limiting them to a single instrumental timbre: contemporary string quartet. The three computer-composed samples were excerpts from two longer works and exhibited three distinct musical textures: homophony, polyphony, and heterophony (the simultaneous variation of a single melodic line). The generative systems for these works are described in (Eigenfeldt, 2012) and (Eigenfeldt, Burnett, and Pasquier, 2012). Although both generative systems were corpus-based in some way, many of the musical decisions (i.e. voice-leading) were based upon an auto-ethnographic analysis. Eigenfeldt selected two excerpts from his own music that matched the textures and harmonic language in two of the generative works, and composed a new excerpt to match the missing one. We also paired computer- and human-composed pieces based on shared structural aspects, taking into account tempo, rhythm, and dynamics. We believe this approach offers a significant improvement upon Moffat and Kelly’s notion of pairing works by so-called “style”, particularly given the rather unlikely pairings they chose.

To address the “interpretational problem”, all pieces were performed by live musicians, in an effort to normalize the musical percepts and reduce variability. Further, in recording the six excerpts, the musicians did not know which works were human-composed or computer-composed, and each excerpt was allocated an equal 30 minutes for rehearsal and recording.

Methodology

Participants

Participants were recruited from SFU using dissemination emails that were sent out to the School for the Contemporary Arts, SIAT, Cognitive Science, and Psychology programs. Participants were incentivized by informing them that they would be included in a draw for four \$50 cash prizes upon completion of the study.

Unlike previous experiments which tested participants’ ability to discern the origin of a composition in a Turing-

like test, we were now interested in whether participant beliefs about the origins of the pieces had an effect on their evaluations. It was therefore no longer necessary to identify musicians and non-musicians within the pool of participants. However, we believed (as did Moffatt and Kelly) that a participant’s cultural, academic, and musical background might help elucidate the reasons for the extent (or presence) of any bias they might have. Therefore, part of the experiment included a demographic questionnaire requesting that each participant indicate their age, gender, university major, country of birth, number of years residing in Canada, number of years studying/playing music, and number of years experience with computer programming languages (as an indication of their computer literacy).

The experiment was run on two separate occasions (Studies 1 and 2 below), with 60 subjects participating in Study 1, and 62 subjects participating in Study 2. Both studies utilized the same experimental design and online test interface, and were methodologically identical.

Presentation of Musical Examples

The programme of musical works was derived from video recordings of live musicians performing three computer-composed and three human-composed musical works. The pieces presented were composed by, or generated by software designed by, Arne Eigenfeldt. All pieces were performed by the Yaletown String Quartet in Vancouver.

Participants viewed six video recordings, approximately one minute in length each, of the quartet performing each of the pieces. This method of presentation was used, in part, to address a deficiency in previous experiments. Granting the participants the ability to see human musicians performing the compositions was intended to help eliminate some of the listening “strategies” participants had previously employed to determine composition authorship—e.g., believing that subtle variations in the quality of the audio betrayed the composition’s origin. Having all pieces performed by human musicians not only “normalizes” the quality of the recordings, it also presents the excerpts in a more realistic setting, potentially allowing us to more accurately capture participants’ perceptions of the music.

Procedure

Participants were provided with a URL to an online survey. The survey was built using Drupal (drupal.org), with additional modules to enable audio and video playback and time tracking (i.e., to ensure that the participants listened to the musical excerpts in full). Participants were then presented with a consent page indicating that completion of the survey would constitute consent.

Participants were presented with one piece of music at a time. After each musical excerpt, they listened to a 10-second “palette cleansing” recording of the musicians tuning their instruments, to help reduce context effects that would arise from directly following one piece with the next. Practice effect was minimized by randomizing the order of presentation.

After each video presentation, participants were asked to indicate their impressions of each piece, on four different attributes, by ranking them on a bipolar scale with 50 discrete points, labelled only at the left and right extremes. The

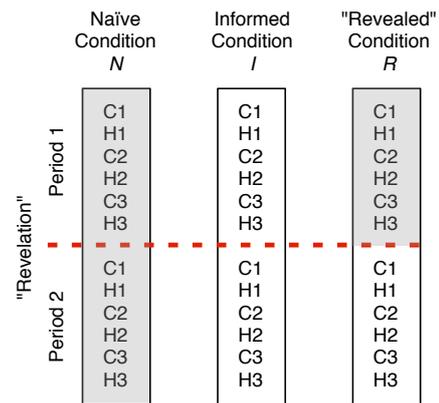


Figure 1: Experiment design using the 3 conditions. Grey shading indicates listening periods during which subjects are unaware of authorship. The horizontal line represents the “revelation” moment, where the second round of evaluations begins. “H” and “C” indicate (un-randomized) human- and computer-composed pieces, respectively.

following dimensions were indicated: **Good-Bad, Like-Dislike, Emotional-Unemotional, and Natural-Artificial.** This evaluation procedure was inspired by that used to evaluate the BeatBender metacreation (Levisohn and Pasquier, 2008). The spreading out of evaluations in this manner (i.e., using a set of bi-polar pairs) was proposed to help identify bias manifesting in a way that could have been obscured were the evaluations condensed into a single rating variable such as “liking” (as in previous experiments). Additionally, past research has indicated that maintaining focused attention (along with discriminative listening and emotional involvement) is critical for the accurate assessment of musical aesthetics (Madsen and Geringer, 2008). In marketing research, predictive validity has been shown to be high when using multiple-item scales over single-item scales (Diamantopoulos et al., 2012). We believe having participants contemplate this variety of dimensions during the listening task is an appropriate way to facilitate the desired level of attention and involvement.

The full set of musical excerpts was presented to participants twice, each time under one of two experimental settings. In the “naïve” setting, participants were not informed of the authorship of any of the pieces, and the experiment was presented as an investigation of the effect that visually witnessing a performance has on one’s aesthetic evaluation of the music performed. In the “informed” setting, however, participants were explicitly told (and reminded) of the authorship of the pieces.

Ideally, the two exposures to the excerpts would have taken place under all possible combinations of settings to eliminate confounding covariates. However, we cannot deinform participants about piece authorship once they have been informed. Therefore there were three different experimental conditions: fully naïve (N), fully informed (I), and “revealed” (R). For the “revealed” condition, subjects start the experiment in a naïve condition, but conclude in an informed condition, with a revelation occurring midway (see Figure 1). This allows us to check for any “reaction” effect, where the shock of the revelation inspires a drastic change in evaluations.

Having these three groups, each broken into two periods, wherein the six pieces are reevaluated, also allows us to control for novelty, exposure, and fatigue effects. If we had conducted the experiment solely with the “revealed” group, as in Moffatt and Kelly, any increase or decrease in evaluations could be attributed to a loss of novelty (i.e., becoming bored), or an increase in familiarity (the “mere exposure” effect, leading to an increase in enjoyment). Listening fatigue could affect the second evaluation in unpredictable ways.

Following the second round of evaluations, participants were thanked for their participation but were asked one final demographic question about their experience with computer programming languages. This was asked at the end of the experience so as to not rouse suspicions and tip participants to the true nature of the study during the initial demographic questions. Participants were then directed to a separate website where they provided their email address so that we could confirm they had finished the survey. Here they were given the option to indicate whether they would like to be contacted about the results of the experience and/or be entered into the prize draw. As this section was separate from the survey-proper, it prevented us from matching survey answers to identifiable e-mail addresses, preserving anonymity.

Despite the differences between our two designs, we believe the design of the present experiment is similar enough to that used in (Moffatt and Kelly, 2006) to allow for easy comparisons: both contain three human and three computer-composed pieces, in our case paired by composer rather than style (all six pieces in the present experiment were composed for string quartet). Our mixed (*R*) condition mimics that used in Moffatt and Kelly, but our addition of the fully naïve and fully informed conditions allowed us to check for timing and fatigue effects as well. Our experiment also had the advantage of asking the same evaluation questions in both rounds: Moffatt and Kelly asked participants how much they “liked” the compositions in the first round and how much they “enjoyed” the pieces in the second. However, the interpretation of these words is entirely subjective and could vary widely across participants (one can “like” a song very much, but depending on their current mood, they may not actually “enjoy” it at that particular moment). This was a concern that Moffatt and Kelly themselves expressed. We attempted to devise our cover story such that we could ask the same questions during both rounds without the similarities cuing the participants to the true nature of our experiment and compromising the validity of the results. The Moffatt and Kelly study also suffered from a lack of randomization of excerpt presentation order, and this too has been addressed for the present study. Finally, the addition of the demographic questionnaire inquires about the age, sex, and culture of the participants, factors which Moffatt and Kelly believed could shed light on the proposed bias we are seeking to identify.

Hypotheses

We anticipated a number of effects, both within and between the three conditions *N*, *I*, and *R*. The null hypothesis is that we should see no significant differences in the evaluation of the pieces among the three groups; the only changes between the initial and subsequent presentations of the stimuli should reflect novelty, exposure, or fatigue effects and have

the same influence in each condition. As for anticipated experimental effects, we formed four main experimental questions:

1) Among those who hear the pieces naively in the first period, does the comparison of human- vs. computer-composed music change their ratings by different amounts when the authorship is revealed before the second hearing than when it is not?

2) In the second hearing only, is the comparison of human- vs. computer-composed music different when the authorship is known than when it is not?

3) Among those who hear the pieces naively in the first period, is the comparison of human- vs. computer-composed music different when the authorship is known than when it is not?

4) Does the comparison of human- vs. computer-composed music change by different amounts when the authorship is revealed partway through than when conditions remain fixed for both hearings?

Statistical Methods

Experimental Design

We created two sets of example pieces; the Human set (*a, c, e*) and the Computer set (*b, d, f*). These pieces were paired, so that each human piece had a corresponding computer piece: (*a, b*)(*c, d*)(*e, f*). Pieces (*a, b*) were denoted as *Pair 1*, pieces (*c, d*) as *Pair 2*, and pieces (*e, f*) as *Pair 3*. These three pairs were always presented in this order, but within each pair the order of human or computer composition was considered both ways, resulting in 8 different versions of the survey.

Subjects were assigned to one of these 8 sequences upon enrolment into the study. The experiment was therefore conducted as a split-split-split plot crossover design. *Group* (*N*, *R*, or *I*)—a fixed effect—was assigned to a subject—a random effect. Within each subject there were 12 hearings arranged in nested groups of decreasing size. The fixed effect *Period* has two levels representing the six hearings before the potential reveal (*Period* = 1) or after (*Period* = 2). Within each *Period*, the factor *Pair*, with levels 1, 2, and 3, is assigned to the two hearings representing the pairs of compositions described above. We treat *Pair* as a fixed effect because the order in which the pairs of pieces are heard is always the same. Finally, within each pair, the fixed effect *Composer*, with levels *H* or *C*, is assigned to a hearing. Note that the key experimental factors, *Group* and *Composer*, are both randomized in this design. The fact that *Pair* was not randomized, but rather presented in sequence for each subject, (1, 2, 3) is unimportant because it represents a combination of the fatigue and other effects due to ordering of hearings and random variation among individual compositions. There is no interest in testing any part of this effect. Importantly, it does not confound with condition or composer.

Statistical Analysis

We analyzed the experiment using mixed-effect linear models according to its design (Milliken and Johnson, 2009), using JMP Version 12 software (2015, SAS Institute). We tested for carryover effects of this factor and found no significance.

The central questions our replication seeks to address are:

1. Whether people enjoy human music implicitly: H vs C in Group N
2. Whether people prefer human music when told: H vs C in Group I
3. Whether the difference of the above reveals a human- vs computer-music bias.

We organized our analyses into four contrasts, each designed to address a specific aspect of these hypotheses as described below. We applied the same contrasts to each response dimension.

Let H represent the model-estimated mean of a given response dimension for human-composed compositions, and C the mean of the same response dimension for computer-composed compositions. We use subscripts “1” or “2” to restrict these means to `Period 1` or `2`, respectively, and “N,” “I,” or “R” to restrict these means to `Group N`, `I`, or `R`, respectively. We use “ Δ ” notation to represent the change in mean responses before vs. after the potential reveal:

$$\Delta H = (H_1 - H_2) \quad (1)$$

$$\Delta C = (C_1 - C_2) \quad (2)$$

We add subscripts to these quantities to refer to these changes under a specific `Group`. All contrasts are tested against a t distribution with 714 degrees of freedom.

Results

In analyzing the data, we operated under the assumption that if results in the naïve condition showed no significant preference for either human or computer-composed music, then discrepancies in the other conditions may indicate the presence of bias. Note, however, that our focus is on the *response to being informed of authorship*, not on the estimation of authorship itself.

Broadly speaking, we considered three main factors: 1) The “pure musical impression” (represented by `Group N`), 2) The influence of knowledge of authorship (`Groups N` and `R`), and 3) The influence of the “reveal” (which takes into account an awareness of the deception).

Four contrasts were designed (series S_1 to S_4), based on the stated hypotheses above:

Series 1 isolates the difference across periods for the naïve (N) and mixed (R) groups:

$$S_1 = [\Delta H_N - \Delta C_N] - [\Delta H_R - \Delta C_R] \quad (3)$$

Since the individual terms represent changes of opinion (Δ), and each bracketed difference isolates the effect of authorship on that change, significant S_1 values could be said to indicate a bias—i.e., when informed of authorship, subjects change their opinions.

Series 2 compares only the second `Period` differences of I and R , to N :

$$S_2 = \frac{(H_2 - C_2)_R + (H_2 - C_2)_I}{2} - (H_2 - C_2)_N \quad (4)$$

Here we attempt to account for repetition effects, by evaluating only the opinions of subjects who have already heard the pieces. The evaluation is again between fully-informed subjects (i.e., since `Group R` in `Period 2` is also informed)

	“Emotional”			
	S_1	S_2	S_3	S_4
t -ratio	-0.57	-0.27	0.63	-0.00
p -value	0.57	0.79	0.53	1.00
Scaled				
Estimate	-1.5	-0.4	1.2	-0.0
Std Error	2.6	1.5	1.9	2.4
<hr/>				
	“Good”			
t -ratio	0.03	-0.52	-0.29	0.33
p -value	0.98	0.61	0.77	0.74
Scaled				
Estimate	0.1	-0.7	-0.4	0.7
Std Error	2.2	1.3	1.6	2.0
<hr/>				
	“Like”			
t -ratio	-0.28	-0.52	-0.25	0.09
p -value	0.78	0.60	0.80	0.93
Scaled				
Estimate	-0.8	-0.8	-0.5	0.2
Std Error	2.7	1.6	1.9	2.5
<hr/>				
	“Natural”			
t -ratio	0.47	-0.17	0.51	1.21
p -value	0.64	0.86	0.61	0.22
Scaled				
Estimate	1.2	-0.2	0.9	2.8
Std Error	2.5	1.5	1.8	2.3

Table 1: Summary of t -ratios, p -values, estimated contrasts, and standard errors for each series, across studies 1 and 2.

and naïve subjects, and compares the average evaluations of all informed subjects against those of the naïve subjects.

Series 3 is similar to series 2, but looking only at R vs N (i.e., excluding I), and thereby contrasting the purely subjective, musical impression with the knowledge of authorship:

$$S_3 = (H_2 - C_2)_R - (H_2 - C_2)_N \quad (5)$$

Series 4 looks again at difference across periods, contrasting the “control” `Groups N` and `I`, against R :

$$S_4 = \frac{[\Delta H_N - \Delta C_N] + [\Delta H_I - \Delta C_I]}{2} - [\Delta H_R - \Delta C_R] \quad (6)$$

Here, we could be said to most directly isolate the deception itself, since we normalize the change across periods for the pure musical impression (N) and the fully-informed evaluation (I), in the absence of any form of deception. These normalized “deception free” evaluations are contrasted with the R case, in which subjects transition not only from the pure musical impression to the knowledge of authorship, but also to an awareness of the deception (i.e., in `Period 2`, they become aware that half of their evaluations have been given with incomplete information). The combined results for both studies are given in Table 1.

We ran a Factorial ANOVA with repeated measures and found no significance, affected primarily by small differences in means (<5 points) relative to scale size (50) and high variability. Additionally, we ran all pairwise comparisons of means for the `Group*Period*Composer` fixed effect, looking for patterns of possible mean differences us-

Group, Period, Composer	“Like”	
	Least Sq Mean	Std Error
<i>N</i> , 1, <i>C</i>	19.5	1.3
<i>N</i> , 1, <i>C</i>	19.5	1.3
<i>N</i> , 1, <i>H</i>	20.0	1.3
<i>N</i> , 2, <i>C</i>	18.7	1.3
<i>N</i> , 2, <i>H</i>	19.5	1.3
<i>R</i> , 1, <i>C</i>	16.9	1.6
<i>R</i> , 1, <i>H</i>	17.6	1.6
<i>R</i> , 2, <i>C</i>	17.6	1.6
<i>R</i> , 2, <i>H</i>	18.0	1.6
<i>I</i> , 1, <i>C</i>	19.7	1.4
<i>I</i> , 1, <i>H</i>	21.0	1.4
<i>I</i> , 2, <i>C</i>	22.3	1.4
<i>I</i> , 2, <i>H</i>	22.0	1.4

Table 2: The Least Square Mean and Standard Error for the “Like” evaluation.

ing Tukey’s Honestly Significant Difference (HSD) method, which controls the type-I error-rate inflation that can occur when multiple hypothesis tests are performed. There were no significant differences among means, indicating that none of the effects of these three factors, or their interactions, were particularly important to this study. The Least Square Mean and Standard Error results for the “Like” comparison are given in Table 2.

Discussion

As with the Moffat and Kelly study, the current study shows that the so-called bias against computational creativity, while observable, is mostly anecdotal and exaggerated. While our results do indicate a negative effect of the knowledge of computer authorship on listener judgements, this effect is not significant.

A clear contributing factor in the failure to find significance is the small differences in means of the evaluations (<5 points on a 50-point scale), and the high variability within that range (see Figure 2 of the Appendix). This would appear to indicate a degree of uncertainty or ambivalence in the listeners, with regard to either the musical content of the examples, or the application of the given criteria to evaluating those examples—participants simply did not have strong opinions, regardless of their knowledge of authorship. Thus, although we did see the anticipated decrease in ratings on the “good” and “like” dimensions for the *R* group (i.e., after the “reveal”), the impact of this change was lost amid the general noisiness of the data.

Unlike Moffat and Kelly, we do not see a significant preference for human-composed music in the naïve group, *N*. We also found no significant decrease in preference for computer-composed music, among fully-informed subjects (group *I*). We did, however, see a slight skewing of scores toward the “bad” dimension in group *R*, after the “reveal”, though the change was not significant. A similarly anticipated change toward “artificial” was noted along the “natural” dimension, though in this case the nature of the evaluation being made is a possible factor. Specifically, it is worth noting that this pair of labels, when compared to the others, fails to denote a clear positive/negative opposition—

i.e., “bad” is clearly opposed to “good”, “dislike” opposed to “like”, and “unemotional” to “emotional.” However, it is not so clear that “artificial” should be opposed to “natural.” Though often used in opposition colloquially, these terms carry strong ideological associations (Žižek, 2011), and thus represent points along an ethical continuum, perhaps more so than extrema in a relationship of contradiction or reversal. The change in response in this case may be an indication of a cultural attitude toward these underlying ideologies—simply put, a human is “natural” and a computer “artificial”; the musical appraisal may be strictly coincidental.

Despite the lack of significant findings, it is worth making a couple of further observations, with reference to Figure 3 in the Appendix. First, there are notable differences between participant responses across the two studies. Looking at the dimensions related most directly subjective preference—“Good” and “Like”—we see that, while in Study 1 the results indicate a positive change in *Period 2* for the *N* group, supporting empirical findings on the relationship between familiarity and musical preference (Szpunar, Schellenberg, and Pliner, 2004; Schubert, 2007), Study 2 opposes this pattern. In fact, Study 2 shows (weakly) correlated negative changes of rating in *Period 2* across all groups for these dimensions, suggesting that perhaps listener fatigue is a contributing factor.

It is also worth noting that, in Study 1, for the “Natural” dimension, participants in the *R* group show a positive change for the computer-composed works, and a negative change for the human-composed works. This appears to suggest that they were pleasantly surprised at the “naturalness” of the works that were revealed to be computer-composed, and perhaps somewhat disappointed at the “unnaturalness” of the works that were revealed to be human-composed; an outcome that may point to the ideological underpinnings of the terms “natural” and “artificial” as a contributing factor, as discussed above.

Future Work

Although the sample sizes for the current study were statistically adequate, the narrow range of ratings, relatively high variability within that range, and overall change in ratings between studies 1 and 2, suggest that perhaps the study should be re-run with a larger number of subjects. Given the randomization of group assignment (*N*, *I*, and *R*), a larger overall population would also help balance the sizes of the different groups. We also note that use of the terms “natural” and “artificial” should be reconsidered, so as to indicate a clearer positive/negative opposition, in keeping with the other dimensions (i.e., “good-bad,” “like-dislike,” and “emotional-unemotional”). Additionally, we recognize that the standardizing choices made in the current study also represent a limitation, in that we have studied only these selected musical conditions. Hence, it cannot be confirmed that the results of this work would apply to other musical genres or other circumstances that differ from the conditions used here.

Further, we are interested in exploring the possibility of using rank-based, as opposed to ratings-based, evaluations. As outlined by Yannakakis and Martínez (Yannakakis and Martínez, 2015), rank-based questionnaires can help eliminate some of the problems associated with ratings-based questionnaires when evaluating subjective, psychological

factors like emotional response, preference, or opinion. In the case of the current study, for example, a rank-based choice—simply ranking the works in each Pair according to how well they represent the dimension under consideration (“good”, “like”, “emotional”, “natural”)—would eliminate the above statistical problem of narrowly distributed ratings, while also obviating use the word “artificial.”

Finally, we are curious whether there may be a correlation between the rating-change across Periods (i.e., Eq. 1 and 2 above) and the musical content of the works under evaluation. Specifically, we are interested in knowing whether the overall complexity of the musical examples has an influence on listeners’ ratings after the “reveal.” We suspect that the degree of change may be correlated with the so-called “inverted-U” pattern, proposed to govern subjective judgments of aesthetic value (Walker, 1981). The inverted-U model suggests that the subjective assignment of aesthetic value follows an “inverted-U” pattern, such that value (or quality) is considered highest for works that match the subject’s preferred complexity level—less complex works are rated lower, as are more complex works. We would like to know whether subjects that potentially hold a bias against computer-composed music provide more sharply contrasting evaluations after the “reveal” for works that match their preferred complexity level. This would suggest that it is not simply the fact of computers composing music that such listeners take exception to, but rather that such systems compose works with a complexity rivalling that of their preferred human-composed works. Such a finding would help establish a more adequately complex understanding of listeners’ attitudes about computational creativity, while at the same time potentially offering a benchmark for selecting works—both human- and computer-composed—for similar studies in the future.

Conclusion

We outlined an experimental design for investigating the proposed bias against musical metacreativity. The experiment was conducted over two studies, on approximately 120 students from Simon Fraser University. Similar to a previous study by (Moffat and Kelly, 2006), we did not find significant support for the presence of such a bias, though our results do suggest that this bias exists, in some listeners. As our results were not significant, we refrained from conjecture based on demographic information. We also outlined a number of possible improvements to our design for future studies.

References

- Bem, D. J. 2011. Feeling the future: experimental evidence for anomalous retroactive influences on cognition and affect. *Journal of personality and social psychology* 100(3):407.
- Blitstein, R. 2010. Triumph of the cyborg composer. *Miller-McCune Magazine*: <http://www.miller-mccune.com/culture-society/triumph-of-the-cyborg-composer> 8507.
- Burnett, A.; Khor, E.; Pasquier, P.; and Eigenfeldt, A. 2012. Validation of harmonic progression generator using classical music. In *Third International Conference on Computational Creativity (ICCC 2012)*, 126–133.
- Cheng, J. 2009. Virtual composer makes beautiful music—and stirs controversy. *Ars Technica* 29.
- Cope, D. 1996. *Experiments in musical intelligence*, volume 12. AR editions Madison, WI.
- Cope, D. 2004. *Virtual music: computer synthesis of musical style*. MIT press.
- Diamantopoulos, A.; Sarstedt, M.; Fuchs, C.; Wilczynski, P.; and Kaiser, S. 2012. Guidelines for choosing between multi-item and single-item scales for construct measurement: a predictive validity perspective. *Journal of the Academy of Marketing Science* 40(3):434–449.
- Eigenfeldt, A.; Burnett, A.; and Pasquier, P. 2012. Evaluating musical metacreation in a live performance context. In *Proceedings of the Third International Conference on Computational Creativity*, 140–144. Citeseer.
- Eigenfeldt, A. 2012. Corpus-based recombinant composition using a genetic algorithm. *Soft Computing* 16(12):2049–2056.
- Garcia-Marques, T., and Azevedo, M. 1995. Multiple comparisons and the problem of alpha inflation. anova as an example. *Psicologia X* 195–220.
- Madsen, C. K., and Geringer, J. M. 2008. Reflections on puccini’s la bohème investigating a model for listening. *Journal of Research in Music Education* 56(1):33–42.
- Milliken, G. A., and Johnson, D. E. 2009. *Analysis of messy data volume 1: designed experiments*, volume 1. CRC Press.
- Minsky, M. L. 1982. Why people think computers can’t. *AI Magazine* 3(4).
- Moffat, D., and Kelly, M. 2006. An investigation into people’s bias against computational creativity in music composition. In *Proceedings of the third joint workshop on Computational Creativity (as part of ECAI 2006), Riva del Garda, Italy*.
- Pasquier, P.; Eigenfeldt, E.; Brown, O.; and Dubnov, S. 2016. An introduction to musical metacreation. *ACM Computers in Entertainment*.
- Rosenthal, R. 1979. The file drawer problem and tolerance for null results. *Psychological bulletin* 86(3):638.
- Schubert, E. 2007. The influence of emotion, locus of emotion and familiarity upon preference in music. *Psychology of Music* 35(3):499–515.
- Szpunar, K. K.; Schellenberg, E. G.; and Pliner, P. 2004. Liking and memory for musical stimuli as a function of exposure. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 30(2):370.
- Tversky, A., and Kahneman, D. 1974. Judgment under uncertainty: Heuristics and biases. *Science*. 185(4157):1124–1131.
- Walker, E. L. 1981. The quest for the inverted U. In *Advances in intrinsic motivation and aesthetics*. Springer. 39–70.
- Yannakakis, G. N., and Martínez, H. P. 2015. Ratings are overrated! *Frontiers in ICT* 2:13.
- Young, E. 2012. Nobel laureate challenges psychologists to clean up their act. *Nature News*.
- Žižek, S. 2011. *Living in the end times*. Verso.

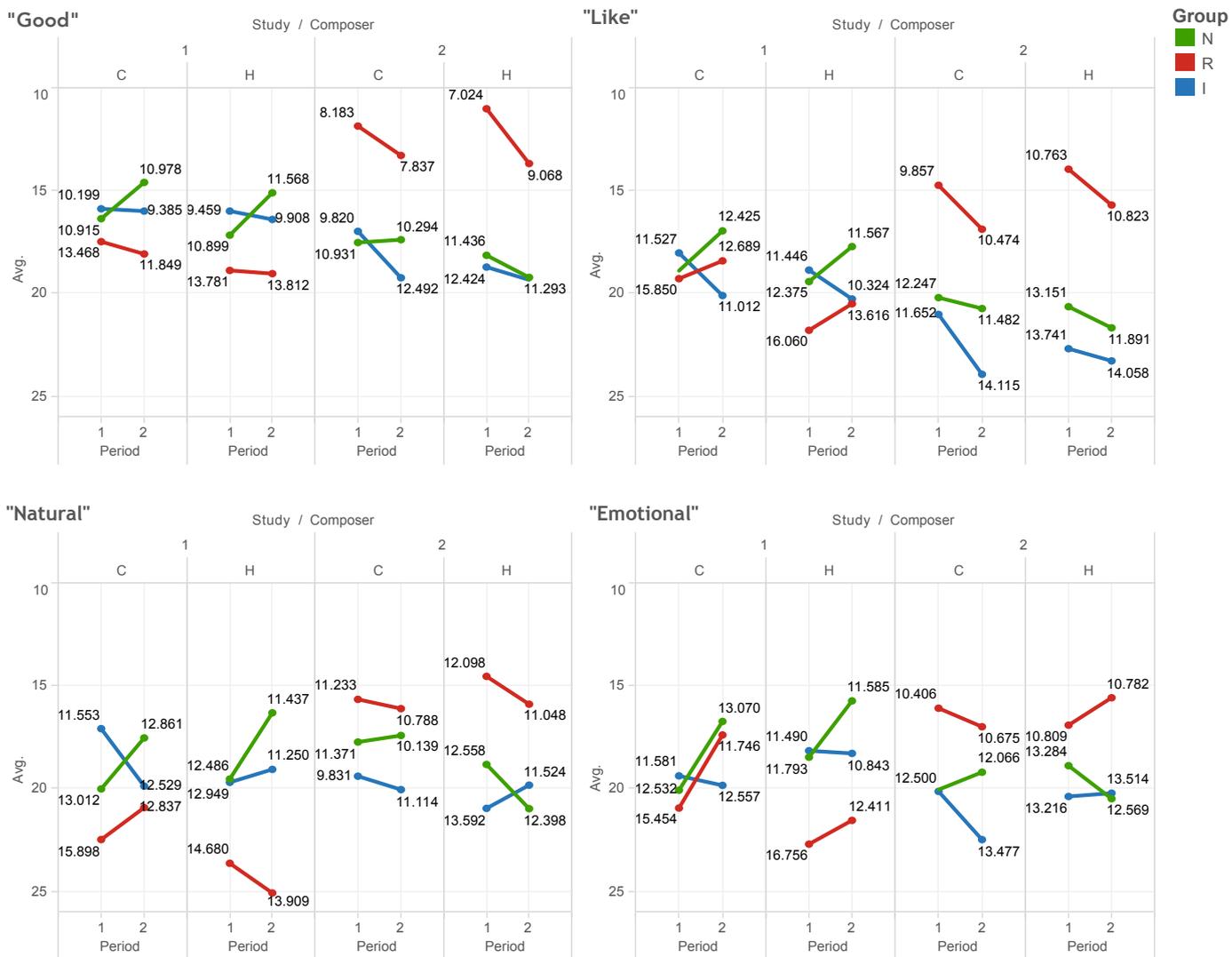


Figure 3: Mean subject ratings for all dimensions. Note that the y-axis is scaled to focus on the range of subject responses. The actual scale range for the study was 0 to 50, with 0 on the left and 50 on the right—i.e., 0=left="good", 50=right="bad".

Preference Models for Creative Artifacts and Systems

Debarun Bhattacharjya
Cognitive Computing Research
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598 USA
debarunb@us.ibm.com

Abstract

Although there is vigorous debate around definitions of creativity, there is general consensus that creativity i) has multiple facets, and ii) inherently involves a subjective value judgment by an evaluator. In this paper, we present evaluation of creative artifacts and computational creativity systems through a multiattribute preference modeling lens. Specifically, we introduce the use of multiattribute value functions for creativity evaluation and argue that there are significant benefits to explicitly representing creativity judgments as subjective preferences using formal mathematical models. Various implications are illustrated with the help of examples from and inspired by the creativity literature.

Introduction

Computational creativity (henceforth CC) is an interdisciplinary field that studies the role of computers in the creative process. Several success stories in domains such as visual art, music, literature, humor, science and mathematics have already been noted (Buchanan 2001; Cardoso, Veale, and Wiggins 2009; Colton and Wiggins 2012). One of the issues that has plagued the research community, analogous to the broader field of artificial intelligence, is around pinning down what it means to be creative. This is of course not surprising as creativity is often considered the pinnacle of human intelligence. Taylor (1988) summarizes several early definitions of creativity from the psychology literature.

Although there is significant debate around defining creativity, there appears to be general agreement about at least two aspects. First, creativity seems to involve ‘novelty and more’, i.e. there are multiple facets to creative artifacts and it is not enough to only be original to be considered creative. Second, creativity is a subjective judgment that is not meaningful without an evaluator of creative value.

In this paper, we frame evaluation in CC and creativity studies in general through a **multiattribute** preference modeling lens that embraces the subjectivity that is inherent in judging creative value, while effectively capturing the notion of creativity involving multiple facets. According to this framework, a (human or machine) evaluator’s preferences are modeled using **preference functions**. There is a rich literature in formal mathematical models of preferences, mainly in fields that pertain to prescriptive decision making,

and we believe there are significant benefits from applying these techniques to the field of CC.

The preference modeling framework is applied to evaluate artifacts as well as CC systems, and we explore the implications of various modeling assumptions through illustrative examples. For instance, creativity researchers and practitioners should be aware of the implications of taking weighted averages of scores along multiple criteria – we argue that such an approach need not always be appropriate in CC, or at the very least, the underlying assumptions should be appreciated. We begin by motivating our research effort.

Evaluating Creative Value

Evaluation, evaluation, evaluation! Evaluating creative artifacts and systems that produce them is to the field of CC what location is to real estate. We distinguish between evaluation of creative artifacts vs. CC systems, like in Pease and Colton (2011). In this section, we provide some background to motivate a multiattribute preference view of creativity.

Novelty and More

Attempts at defining creativity or towards identifying its properties can be seen in the early literature on artificial intelligence. Newell et al. (1958) opined that creative products needed to have novelty and value. Boden (1990) echoed this thought, suggesting that creativity involves generating ideas that are both novel and valuable. Mayer (1999) refers to these two facets as originality and usefulness, citing several alternatives for the latter term, including utility, adaptiveness, appropriateness and significance.

We will avoid the terms ‘value’ and ‘utility’ in how they have been used in the creativity literature because we reserve them for specific concepts from the preference modeling domain. (Our notion of creative value includes novelty.) Instead, we will use the term ‘quality’ to refer to non-novelty related aspects of creative artifacts (Ritchie 2001; Pease, Winterstein, and Colton 2001). Evaluation could in general involve several (> 2) attributes that sufficiently span novelty/originality as well as quality/usefulness.

Subjectivity in Evaluation

Various definitions of creativity explicitly acknowledge the relationship between the creator/creation and an observer (Wiggins 2006) and how a creative artifact must be

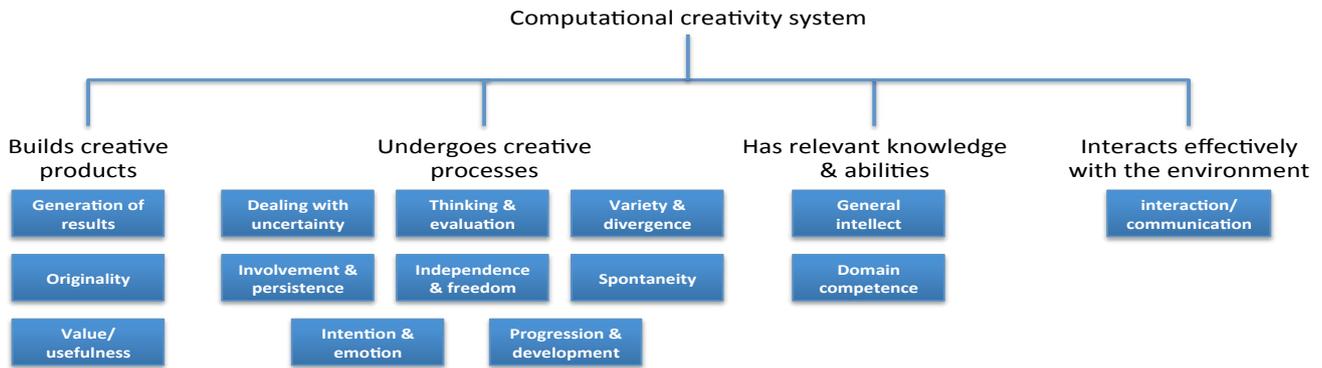


Figure 1: An example of objectives for a computational creativity system, with attributes from Jordanous (2012).

judged or deemed to be creative (Sawyer 2012). The line of research that deals with creativity assessment goes back at least around a hundred years (Cattell, Glascock, and Washburn 1918) and includes popular methods like the consensual assessment technique where artifacts are rated by two or more experts in the field (Amabile 1982).

Assessing a creative artifact is necessarily a cognitive task that involves several complex processes (Varshney et al. 2013). In this paper, we attempt to model an evaluator’s (subjective) preferences for a creative artifact – one approach to understanding these preferences is directly through an overall creativity rating; another is to break down the assessment along multiple attributes. We consider the second approach and adopt the view that the evaluator’s preferences can be captured, at least approximately, through some abstract mathematical representation.

Multiattribute Preference Models: A Review

Decision analysis is an approach to prescriptive decision making that applies the norms of decision theory to practical decision situations, tracing its roots to stalwarts such as Bernoulli, Laplace and Pascal. The field explicitly models decision makers’ subjective beliefs and preferences; the latter aspect has spawned the overlapping fields of multiattribute utility theory (MAUT) and multi-criteria decision making (MCDM). Here we provide a brief summary of relevant concepts, mainly using terminology from the seminal Keeney and Raiffa (1976), but the reader is encouraged to peruse related work (Belton and Stewart 2002; Wallenius et al. 2008).

Preliminaries

A formalization of preferences is useful in the spirit of breaking a larger unmanageable problem into smaller pieces. An **objective** indicates the ‘direction’ in which one strives to do better. It is often convenient to generate objectives by organizing them in a hierarchy, thereby meaningfully structuring them. **Attributes** (or criteria) are measures that adequately determine how well an objective has been met. Figure 1 shows an example objectives hierarchy (with only one level) for a CC system: four high-level objectives are spanned by fourteen attributes from Jordanous (2012).

There are several desirable properties of attributes: they should be *complete* (sufficiently capture degree to which objectives are met), *operational* (meaningful and understandable), *nonredundant* (double counting should be avoided) and *minimal* (with manageable problem dimension).

Making a judgment about how much to give up on an objective for another is the essence of a trade-off, and this is represented by a functional form over attributes. There are two types of preference functions: **utility functions** and **value functions**, also referred to as cardinal and ordinal utility functions: the distinction between them is whether uncertainty is involved or not, respectively. A utility function thus captures both a decision maker’s strength of preference as well as their attitude towards risk. In this paper, we focus solely on value functions as all of the situations that are studied are those of certainty, but the concepts apply broadly. Also, we do not expound upon how preference functions are elicited/assessed here – instead, we refer the reader to the literature (Keeney and Raiffa 1976; von Winterfeldt and Edwards 1986).

Value functions

Let X_1, \dots, X_M be a set of M attributes where lower case x_i denotes the score/consequence along attribute X_i . We use the notation \bar{X}_i to refer to the complement set of attributes to X_i , and x_i^* and x_i^0 for the best and worst scores of attribute X_i . A **preference structure** is defined over the domain of attributes if all points in the domain are comparable and no intransitivities exist. In that case, if $\mathbf{x} = \{x_1, \dots, x_M\}$ and $\mathbf{y} = \{y_1, \dots, y_M\}$ are two alternatives, then a (measurable) value function $v(\cdot)$ (Dyer and Sarin 1979) is one such that $v(\mathbf{x}) \geq v(\mathbf{y})$ if and only if $\mathbf{x} \succeq \mathbf{y}$, where the symbol \succeq reads ‘preferred or indifferent to’.

A preference structure over attributes is determined by **indifference curves**, i.e. complete sets of points in the domain of attributes where the decision maker is indifferent. It can be shown that monotonic transformations of value functions do not change the preference structure, and it is often convenient to normalize value functions between most and least preferred alternatives such that $v(\mathbf{x}^0) = 0$ and $v(\mathbf{x}^*) = 1$.

Additive value functions The simplest and most widely used value function is the **additive** function, of the form:

$$v(x_1, \dots, x_M) = \sum_{i=1}^M \lambda_i v_i(x_i), \quad (1)$$

where $\lambda_i \geq 0 \forall i$ are the weights, $\sum_{i=1}^M \lambda_i = 1$, and $v_i(\cdot)$ are marginal (one-dimensional) value functions bounded between 0 and 1. The additive value function is thought to be fairly robust (Stewart 1996) and is extremely popular in practice, but unfortunately it is often misused. This is because its proponents often forget or are unaware that it applies if and only if there is **mutual preferential independence** among attributes (for $M > 2$). This condition holds if every $\mathbf{Y} \subset \{X_1, \dots, X_M\}$ is preferentially independent of its complement, i.e. the preference structure over \mathbf{Y} does not depend on the scores of the attributes in the complement set. The following example illustrates the implications.

Example 1. [Evaluating cartoon captions]

Sternberg et al. (2006) performed creativity assessments on captions provided by students for cartoons from the New Yorker. These were adjudicated based on three attributes: cleverness, humor and originality, all scored on a 5 point scale, and the total creativity score was computed by summing up the three scores. From a preference modeling perspective, this procedure implicitly assumes that preferences for these cartoon captions, when viewed as creative artifacts, follow an additive value function. This in turn implies mutual preferential independence among attributes. We pose the question – is this condition appropriate?

Consider indifference curves for humor and originality, given a cleverness score. If the score on cleverness is high, then it seems reasonable that the evaluator may generally be willing to give up a large score of humor for some originality. The rationale behind this claim is that the evaluator may view cleverness and humor as two attributes for a higher level objective (quality), and may be willing to compensate one for the other. On the other hand, if the cleverness score is low, the evaluator may no longer be willing to give up as much humor for the same score increment on originality.

These (hypothetical) preferences are clearly inconsistent with mutual preferential independence, which in a three attribute problem enforces identical indifference curves for every pair of attributes, conditional on the score of the third attribute. This would make the additive value function inappropriate in this case. Our conjecture is that preferences of this sort are probably commonplace for creative artifacts. Understanding the preferential assumptions behind implicit functional forms could be broadly beneficial to the creativity community. □

Value copulas The value function $v(\cdot)$ can take any form, including one that does not need to subscribe to independence assumptions. A recent approach to modeling potentially complex preference functions is that of **copulas**, and although they were introduced to model utility functions (Abbas 2009), they can also be used for value functions. A copula $C_\lambda(z_1, \dots, z_M)$ is a multivariate function that is a continuous mapping from the hypercube $[0, 1]^M$

to the interval $[0, 1]$, normalized such that $C(\mathbf{0}) = 0$ and $C(\mathbf{1}) = 1$ (Sklar 1959). It is non-decreasing in each of its arguments z_i , and for each argument, there exists some reference scores of the complement attributes for which it is an affine function. A value function can be constructed from a copula as follows:

$$v(x_1, \dots, x_M) = C_\lambda \left(v_1(x_1 | \bar{x}_1^{\lambda_1}), \dots, v_M(x_M | \bar{x}_M^{\lambda_M}) \right), \quad (2)$$

where $C_\lambda(\cdot)$ is a copula and $v_i(x_i | \bar{x}_i^{\lambda_i})$ are normalized conditional value functions, defined as:

$$v_i(x_i | \bar{x}_i^{\lambda_i}) = \frac{v_i(x_i, \bar{x}_i^{\lambda_i}) - v_i(x_i^0, \bar{x}_i^{\lambda_i})}{v_i(x_i^*, \bar{x}_i^{\lambda_i}) - v_i(x_i^0, \bar{x}_i^{\lambda_i})}, \quad (3)$$

with $\bar{x}_i^{\lambda_i}$ denoting a particular reference score for the set of complement attributes to X_i . Assessing a conditional value function therefore entails determining the marginal rate of value for an attribute when all other attributes are set to some reference scores. It is typical to assess this function at the complementary maximum (\bar{x}_i^*) or minimum (\bar{x}_i^0) scores.

The model of equation (2) represents any value function that is continuous, bounded, non-decreasing in each argument, and strictly increasing with each argument for at least one reference score of the complement attributes. The power of the copula is that like the additive function, it models a high-dimensional function by aggregating one-dimensional functions, yet allows for a much wider class of functions.

An example of a copula where conditional value functions are assessed at the maximum scores of the complement of each attribute is the extended Archimedean copula:

$$E(z_1, \dots, z_M) = a\psi^{-1} \left[\prod_{i=1}^M \psi(l_i + (1 - l_i)z_i) \right] + b, \quad (4)$$

where $l_i \in [0, 1)$, $a = 1 / \left(1 - \psi^{-1} \left[\prod_{i=1}^M \psi(l_i) \right] \right)$, $b = 1 - a$, and the **generating function** ψ has the same mathematical properties as a strictly increasing cumulative probability distribution function. A special case occurs when $l_i = 0 \forall i$ and the generating function is linear, $\psi(z_i) = z_i$, resulting in the **multiplicative** form:

$$v(x_1, \dots, x_M) = \prod_{i=1}^M v_i(x_i | \bar{x}_i^*). \quad (5)$$

Both additive and copula value functions will be applied in subsequent sections.

A Two-Attribute Model for Artifacts

As we highlighted earlier, maximizing novelty (or originality) as well as quality (or usefulness) are reasonable high-level objectives for creative artifacts. The simplest preference model therefore involves two attributes: novelty X_N and quality X_Q . In this section, we first discuss some potential value functions over these two attributes and then consider a scenario that explores the implications of potentially mis-characterizing a user's value function.

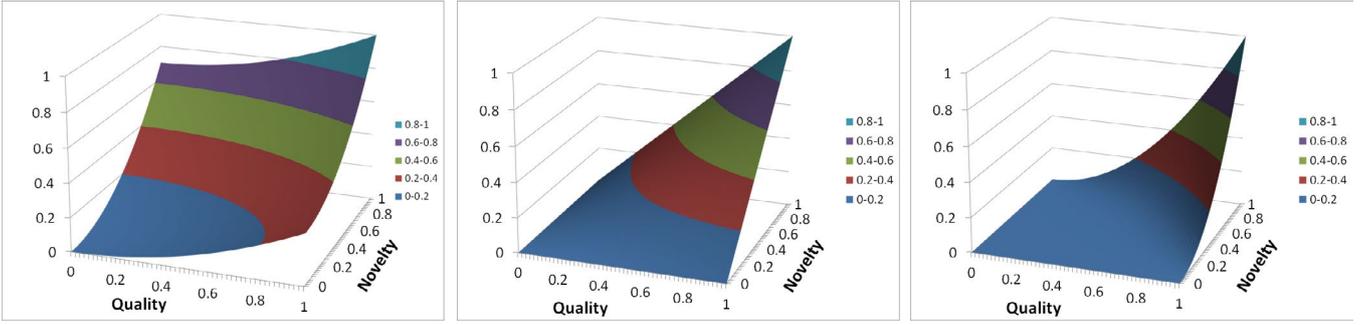


Figure 2: Some example value functions over novelty and quality: Left) additive with $\lambda = 0.7$ and power function marginals, $\beta_N = \beta_Q = 2$; Middle) multiplicative with linear conditionals at maximum reference points; Right) copula with exponential generating function, $\delta = -5$, and power function conditionals at maximum reference points, $\beta_N = \beta_Q = 2$.

Value functions for novelty and quality

An evaluator's preferences for an artifact with novelty x_N and quality x_Q can be represented by $v(x_N, x_Q)$. If the attributes are bounded, then they can be normalized to lie in $[0, 1]$. Assuming that $v(\cdot)$ is bounded, and since value functions are unique up to monotonic transformations, we set $v(0, 0) = 0$ and $v(1, 1) = 1$.

Applying the additive value function from equation (1):

$$v(x_N, x_Q) = \lambda v_N(x_N) + (1 - \lambda) v_Q(x_Q), \quad (6)$$

where $\lambda \in [0, 1]$ is the weight for novelty and $v_N(\cdot)$ and $v_Q(\cdot)$ are marginal value functions bounded between 0 and 1. If more of an attribute is preferred to less, its marginal value function must be increasing. An example is the power function, where $v_j(x_j) = x_j^{\beta_j}$ for attribute j . $\beta_j > 1$ implies marginally increasing value – this seems like a reasonable form for creative artifacts, for instance, the user may deem that increasing novelty from say 0.8 to 0.9 is more valuable than from 0.2 to 0.3. $\beta_j = 1$ represents a linear marginal value function, i.e. $v_j(x_j) = x_j$ for attribute j .

Figure 2 (left) plots an additive value function over the entire domain, with weight on novelty $\lambda = 0.7$ and power marginal value functions with $\beta_N = \beta_Q = 2$. Four indifference curves are highlighted, equally spaced between the worst (0) and best (1) value. Note that when the quality score is 0 and novelty score is 1, the value is as high as 0.7, because the weight on novelty is 0.7.

The evaluator may however deem that there is no creative value (i.e. it equals 0) if either novelty or quality are at their lowest scores. Additive value functions do not support such a condition. Another aspect that additive functions fail to capture sufficiently is that of the **confluence effect**: much like how creativity in people involves more than a simple sum of their level on separate skills/abilities (Sternberg and Lubart 1991), we hypothesize that the value in a creative artifact, as deemed by an evaluator, often arises from the confluence of scores along attributes. Extended Archimedean copulas from equation (4) are examples of copulas that could potentially be used to model both these effects.

Figure 2 (middle) depicts the multiplicative form from equation (5) with linear conditional value functions at maximum reference points. The function is grounded, i.e. equals

0 when either novelty or quality is 0, and increases only when both attributes score high together. The confluence effect is heightened even further in Figure 2 (right), depicting a value copula with an exponential generating function:

$$\psi(z_i) = \frac{1 - e^{-\delta z_i}}{1 - e^{-\delta}}. \quad (7)$$

The parameter δ models value dependence among attributes; $\delta = -5$ was chosen here. The figure indicates a low value for a significant region of the domain, and the value increases only for significantly high scores on both novelty and quality. A value function model should of course reflect the evaluator's preferences as much as possible.

A CC recommender scenario

A CC system should ideally cater to the user's preferences for artifacts/items – but what is the impact of a potential mischaracterization of the user's value function? Let us study this question using an illustrative scenario where the CC system either recommends one artifact or a list of artifacts.

Suppose there are N items produced with novelty x_N^i and quality x_Q^i of the i^{th} item. The standard approach in many creativity studies is to average out the scores to rate artifacts; the implicit value function in such a situation is additive with $\lambda = 0.5$ and marginal value functions that are linear.

If the system provides the user with the top candidate from the N items as determined by the mean rating, and if the user has value function $v(\cdot)$, then the loss in value is:

$$\text{Loss} = \max_i [v(x_N^i, x_Q^i)] - v(x_N^{i*}, x_Q^{i*}), \quad (8)$$

where i^* is the item index with the highest mean rating, or formally, $i^* = \text{argmax}_i \frac{x_N^i + x_Q^i}{2}$.

If the user is instead presented with an ordered ranking of items, then the discrepancy between the optimal rank ordering and the one suggested by the system is:

$$\text{Rank dist.} = D \left[r \left(i : v \left(x_N^i, x_Q^i \right) \right), r \left(i : \frac{x_N^i + x_Q^i}{2} \right) \right], \quad (9)$$

where $r(i : C^i)$ denotes the rank ordering over items i based on condition C^i , and D is some distance metric. The

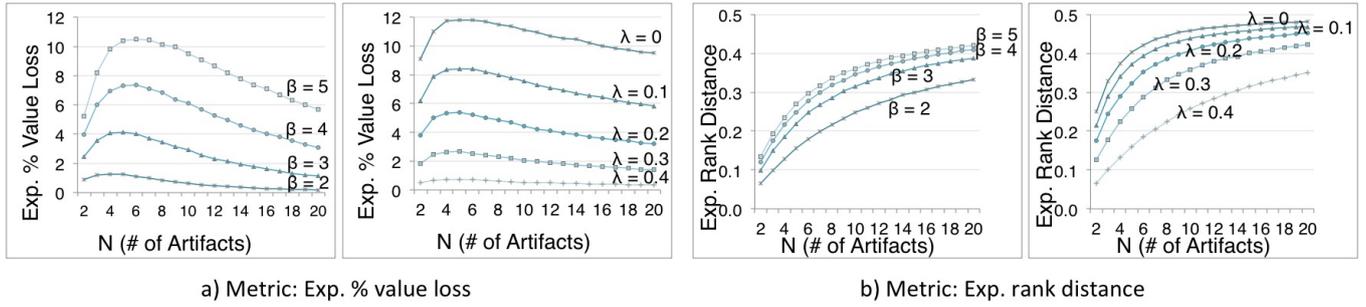


Figure 3: The two metrics in Example 2 for $N = \{2, \dots, 20\}$ as a function of parameters β and λ . Left (both a and b): Sensitivity to β for $\lambda = 0.5$; Right (both a and b): Sensitivity to λ for $\beta = 1$.

following numerical example provides some insights into the implications of a potential discrepancy.

Example 2. [Sensitivity to additive function parameters]

Suppose a CC system draws items independently and uniformly from the unit cube: X_N^i and $X_Q^i \sim U(0, 1) \forall i$. Furthermore, suppose that the user’s value function is additive with weight on novelty λ and where marginal value functions are power functions. For simplicity, consider the case where parameters for the marginal functions are identical, i.e. $\beta_N = \beta_Q = \beta$. Simple probabilistic analysis reveals that the mean ratings of items follow a triangular distribution from 0 to 1 with mode at 0.5. One can compute the expected loss in value and expected rank distance using Monte Carlo simulations over the metrics in equations (8) and (9).

Figure 3(a) plots the % expected loss against the number of artifacts N for various parameter values of β and λ . This metric first increases with N but then decreases, after the mean rating approach has more items from which to select one that is closer in value to the optimal as per $v(\cdot)$. The figure indicates that when a parameter is significantly mischaracterized by the mean rating approach (after fixing the other parameter at its reference value), the potential % loss in value could be around 10 – 12%. The loss in value could be even higher if β and λ are jointly mis-specified. Due to the model assumptions, the results are symmetric around $\lambda = 0.5$ but not around $\beta = 1$.

Figure 3(b) repeats the exercise using a discrepancy in rankings where D is the normalized Kendall tau distance. This distance metric normalizes the number of swaps required to convert one rank order to another, such that identical orders result in 0 distance whereas an order and its reverse have distance 1. The rank distance increases with N and can reach distances of around 0.4 – 0.5, implying that the system could potentially provide a user with a rank order of artifacts significantly different from the optimal order.

In this example, the user’s value function was assumed to be additive. Even in this case, where the functional form is the same as the mean rating approach, not fully appreciating the user’s strength of preferences over attributes could result in CC systems providing lower value artifacts to users. \square

A Three-Attribute Model for Sets of Artifacts

Ritchie (2001; 2007) introduced an evaluation framework for a CC system that assesses the set(s) of artifacts it produces, where each artifact is associated with two measures: typicality $T \in [0, 1]$ and quality $Q \in [0, 1]$. The distinction between these two measures, for example, is how typical a joke is vs. how funny it is. Here we consider a preference model with three attributes based on this framework.

Consider a CC system that produces a large set of artifacts where the T and Q measures of each artifact are generated from a joint probability density function (pdf) $f_{T,Q}(t, q)$. Ritchie proposed several criteria based on these measures. We formulate a model with three attributes of a set of artifacts: novelty X_N and conformance X_C , both properties of the typicalities of the artifacts in the set – the idea is that some artifacts should conform to item type whereas others should be ‘atypical’ and therefore deemed novel – along with the quality X_Q of the set. We define these attributes based on the fraction of artifacts that are less or greater than specified thresholds. For a large enough set, these can be approximated as: $X_N \approx \int_0^{\alpha_N} f_T(t)dt$ (fraction such that $T \leq \alpha_N$), $X_C \approx \int_{\alpha_C}^1 f_T(t)dt$ (fraction such that $T > \alpha_C$), $X_Q \approx \int_{\alpha_Q}^1 f_Q(q)dq$ (fraction such that $Q > \alpha_Q$), where $f_T(t)$ and $f_Q(q)$ are marginal pdfs for each artifact’s T and Q measures. Clearly, only the marginal pdfs of the generating distribution of typicality and quality matter here.

If the system builder can adjust the parameters θ of the generating distribution $f(\cdot)$, and if the user’s value function over the three attributes of the set of artifacts is $v(\cdot)$, then the optimal parameters are:

$$\theta^* = \operatorname{argmax}_{\theta} [v(x_N(\theta), x_C(\theta), x_Q(\theta))]. \quad (10)$$

The reader should note that according to this three-attribute formulation, a large fraction of highly typical artifacts results in low novelty in the set – but other interpretations of Ritchie’s model are possible; for instance, an artifact may be considered typical in its form but novel in its content. Consider the following illustrative numerical example.

Example 3. [System design: Typicality vs. quality]

Suppose the typicality and quality of each artifact of a CC system are generated from independent truncated Gaus-

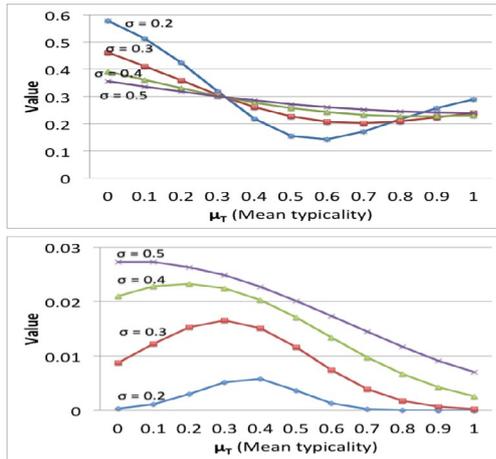


Figure 4: Value of a set of artifacts from Example 3 as a function of parameters μ_T and σ . Top: Mean rating value function; Bottom: Multiplicative value copula.

sian distributions with parameters μ_T , σ_T and μ_Q , σ_Q respectively. To examine a specific simple case, suppose $\sigma_T = \sigma_Q = \sigma$, and thresholds $\alpha_N = 1 - \alpha$, $\alpha_C = \alpha_Q = \alpha$ for $\alpha = 0.7$. Furthermore, suppose the system builder(s) can choose the mean typicality but they would need to sacrifice it for mean quality – formally, they can determine μ_T under the constraint $\mu_T + \mu_Q = 1$. How should they choose μ_T ?

Figure 4 (top) helps analyze this problem for a user with a *mean rating value function*, which is the additive function from equation (1) with equal weights and linear marginal value functions: $v(x_N, x_C, x_Q) = \frac{x_N + x_C + x_Q}{3}$. Clearly, the optimal $\mu_T^* = 0$ for all the displayed σ curves. Although a low mean typicality results in poor conformance, it yields both high novelty and quality, and the additive function willingly sacrifices conformance for the other two attributes.

Figure 4 (bottom) repeats the analysis for a user with a *multiplicative value copula* from equation (5) with linear conditional value functions: $v(x_N, x_C, x_Q) = x_N * x_C * x_Q$. The solution is no longer straightforward because poor performance on any individual attribute needs to be avoided. As σ decreases, a more intermediate μ_T that effectively balances the three attributes should be chosen. This example highlights how the user’s value function can (and should) impact a CC system builders’ decisions.

The bottom figure also reveals that a higher standard deviation improves value to the user, because it increases the fraction of the set of artifacts above or below specified thresholds. In other words, more randomness in the system is preferable; some may view this result as antithetic to the notion of CC. Several cases have been made against focusing solely on the properties of products generated by a CC system, without making other considerations – for instance, there is an argument that creative artifacts will eventually be produced by a random generation system that is “nearly equivalent to the proverbial room full of monkeys pounding on typewriters” (Ventura 2008). \square

Multiple Attributes for CC Systems

The preference modeling techniques discussed for artifacts and sets of artifacts also apply more generally to CC systems. Naturally, the context in which a CC system operates should have an impact on the system builders’ objectives and therefore decisions. For instance, the objectives of a CC system that provides a user with a joke every morning would be substantially different from a culinary CC system attempting a ‘moonshot’ recipe like the next Oreo cookie. In the previous section, we formulated a model that evaluated CC systems based on the set(s) of artifacts produced, disregarding the process by which they were created. Significant research has been pursued on frameworks that also incorporate the processes involved (Pease, Winterstein, and Colton 2001; Colton 2008; Colton, Charnly, and Pease 2011).

It is not our intention here to identify the appropriate attributes for CC systems; there is vibrant discussion in the community on such matters. Instead, we merely highlight that the preference modeling view is entirely consistent with calls for making the criteria of judging CC systems explicit (Jordanous 2012). According to this view, the system builder(s) should first deliberate over their objectives, building a hierarchy as necessary, and then identify a desirable set of attributes that ideally satisfy the properties mentioned earlier. The system builders’ preferences can be represented by a value function over the attributes – this is where the proposed approach goes above and beyond current guidelines for evaluating CC systems. The following numerical example explores the use of value functions to evaluate CC systems. It is intended mainly for illustrative purposes.

Example 4. [Comparing jazz improvisation systems]

Jordanous (2012; 2013) compared three CC systems for jazz improvisation using scores from three judges on a scale of 0 – 10 across fourteen attributes. The attributes are organized by four high-level objectives, inspired by the four Ps (product, process, person, press) model (Rhodes 1961), as shown in Figure 1. A weighted average method was used to compare the systems, but the attribute weights that were determined for the analysis did not really reflect parameters of a user’s additive value function. We will take the liberty of making additional assumptions to craft the data further into a hypothetical example, so as to illustrate some implications of taking a preference modeling approach.

First, we assume that the three CC systems’ scores for each attribute are the mean values of the three judges’ scores, normalized to between 0 and 1. Next, we assume that each high-level objective can be measured by a proxy attribute that aggregates the corresponding attribute scores from the lower level. Specifically, we assume that the value functions for each of the four high-level objectives are additive and equally weighted over the corresponding low-level attributes. The underlying assumption is that there is mutual preferential independence for each of the four high-level objectives. This effectively transforms the original fourteen attribute problem into a four attribute problem. From a modeling perspective, it is often helpful to construct preferences in a hierarchical fashion, but such a dimensional reduction is also useful for simplifying preference assessments.

Attribute	GAmprovising	GenJam	Voyager
'Product'	0.41	0.73	0.48
'Process'	0.34	0.70	0.38
'Person'	0.36	0.72	0.45
'Press'	0.40	0.55	0.57

Table 1: Scores for three jazz improvisation systems.

The data that is generated through this transformation is displayed in Table 1; for the original data, see Jordanous (2013), Ch. 6, Table 6.3. If the goal of the exercise is to determine the best CC system then there is no need to go further – system GenJam dominates system GAmprovising by scoring higher on all four attributes, and almost dominates system Voyager. GenJam is almost surely the most preferred system, but it may be of interest to gauge how valuable it is when compared with others, in which case one needs to assess a value function over the four attributes.

We model a hypothetical value function over the four attributes using a value copula (equation (2)) so as to capture the confluence effect described earlier, where a CC system exhibits higher value only when multiple effects ‘kick in’ together. Specifically, we consider an exponential generating function (equation (7)) for an Archimedean copula (equation (4)) with $l_i = 0 \forall i$. The limiting case of $\delta = 0$ makes the generating function linear, resulting in the multiplicative form (equation (5)). Conditional value functions $v(x_i|\bar{x}_i^*)$ are assumed to be power functions, and for simplicity we assume they are identical, i.e. with the same parameter β .

Figure 5 compares the values of the three systems when parameters β and δ of the value function are varied. The reference case is where conditional value functions are linear ($\beta = 1$) and where the copula is multiplicative ($\delta = 0$). Here, GenJam has value 0.2 but still beats the other systems by a significant margin. Increasing β makes the conditional value functions at the margins more convex and therefore decreases the value; there is not much difference between the three systems for $\beta > 2$. Making the dependence parameter more negative strengthens the confluence effect and decreases the value (compare the middle and right panels of Figure 2 to observe this effect). A positive parameter makes the function concave and increases value. In all cases, GenJam dominates the other systems, and even though this was evident without the need for formulating a value function, the function (and the chosen model) clearly has an impact on the value of the systems. \square

Conclusions

It can be challenging to deliberate over the most pertinent objectives and attributes in many real-world decision situations, and perhaps it is even harder to identify attributes that sufficiently characterize ones preferences for creative artifacts. As Boden (1998) muses: “It is ... difficult to express (verbally or computationally) just what it is that we like about a Bach fugue, or an impressionist painting, ... And to say what it is that we like (or even dislike) about a new, or previously unfamiliar, form of music or painting

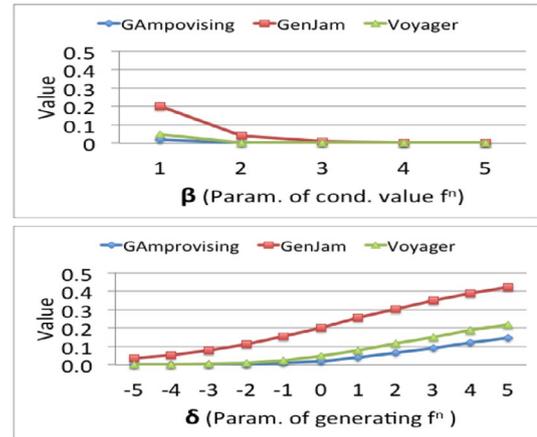


Figure 5: Value of three jazz improvisation systems from Example 4 as a function of parameters β and δ . Top: Sensitivity to β for $\delta = 0$; Bottom: Sensitivity to δ for $\beta = 1$.

is even more challenging.” However, if creativity is inherently subjective and involves a user’s preference judgment, then understanding these preferences is a crucial aspect of CC, regardless of how challenging the task may be and how it is conducted, i.e. whether they are assessed through surveys or estimated through machine learning and related techniques (Fürnkranz and Hüllermeier 2010).

We introduced a preference modeling perspective for evaluating creative artifacts as well as systems – specifically, we formulated various multiattribute value function models. We focused primarily on additive and copula functions, stressing on the importance of the latter family of models and highlighting their potential advantages for creativity evaluation with the help of several illustrative examples. We argue for the explicit study of attributes, including the modeling of preference functions, over ad-hoc analyses that neglects to consider the implications of various assumptions.

There are various benefits to formulating preference models for CC. At an operational level, models that accurately reflect users’ preferences can help in the generation of ideas and artifacts, for instance, they could improve search techniques in the conceptual space. A better understanding of preferences would also result in more effective optimization methods for CC. Furthermore, we have demonstrated that a careful consideration of the objectives of a CC system could help system builders make better strategic decisions. A CC system that could generate new attributes for meeting higher level objectives would be particularly powerful.

There are also potential limitations to using preference models in CC. Although they allow flexibility, more complex models require more parameters, and it can be far from trivial to accurately assess a complicated value function. It remains to be seen how easy or difficult it is for people to respond to preference elicitation schemes that assess multiattribute preference functions for creative artifacts. However, there is little doubt that it is essential for the best empirical research methods to effectively understand how creativity is evaluated in products, processes and ideas.

Acknowledgments

I am grateful to Lav Varshney, Anna Jordanous and four anonymous reviewers for their helpful suggestions.

References

- Abbas, A. E. 2009. Multiattribute utility copulas. *Operations Research* 57(6):1367–1383.
- Amabile, T. M. 1982. Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology* 43:997–1013.
- Belton, V., and Stewart, T. 2002. *Multiple Criteria Decision Analysis: An Integrated Approach*. Springer.
- Boden, M. 1990. *The Creative Mind: Myths and Mechanisms*. London, UK: Weidenfield and Nicolson Ltd.
- Boden, M. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103:347–356.
- Buchanan, B. 2001. Creativity at the metalevel. *AI Magazine* 22(3):13–28.
- Cardoso, A.; Veale, T.; and Wiggins, G. A. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30(3):15–22.
- Cattell, J.; Glascock, J.; and Washburn, M. F. 1918. Experiments on a possible test of aesthetic judgment of pictures. *American Journal of Psychology* 29:333–336.
- Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, 21–26.
- Colton, S.; Charnly, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity (ICCC)*, 90–95.
- Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of AAAI Symposium on Creative Systems*, 14–20.
- Dyer, J. S., and Sarin, R. K. 1979. Measurable multiattribute value functions. *Operations Research* 27(4):810–822.
- Fürnkranz, J., and Hüllermeier, E. 2010. *Preference Learning*. Berlin Heidelberg: Springer-Verlag.
- Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computing* 4:246–279.
- Jordanous, A. 2013. *Evaluating computational creativity: A standardised procedure for evaluating creative systems and its application*. Ph.D. Dissertation, University of Sussex.
- Keeney, R. L., and Raiffa, H. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. J. Wiley, New York.
- Mayer, R. E. 1999. Fifty years of creativity research. In Sternberg, R. J., ed., *Handbook of Creativity*. Cambridge, UK: Cambridge University Press. 449–460.
- Newell, A.; Shaw, J. C.; and Simon, H. 1958. The processes of creative thinking. Technical Report Report P-1320, The RAND Corp., Santa Monica, California.
- Pease, A., and Colton, S. 2011. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB Convention*, 1–8.
- Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In *Proceedings of Workshop Program of ICCBR-Creative Systems: Approaches to Creativity in AI and Cognitive Science*, 129–137.
- Rhodes, M. 1961. An analysis of creativity. *Phi Delta Kappan* 42(7):305–310.
- Ritchie, G. 2001. Assessing creativity. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, 3–11.
- Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.
- Sawyer, R. K. 2012. *Explaining Creativity: The Science of Human Innovation*. USA: Oxford University Press.
- Sklar, A. 1959. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris* 8:229–231.
- Sternberg, R. J., and Lubart, T. I. 1991. An investment theory of creativity and its development. *Human Development* 34(1):1–31.
- Sternberg, R. J., and The Rainbow Project Collaborators. 2006. The Rainbow Project: Enhancing the SAT through assessments of analytical, practical, and creative skills. *Intelligence* 34:321–350.
- Stewart, T. J. 1996. Robustness of additive value function methods in MCDM. *Journal of Multi-Criteria Decision Analysis* 5(4):301–309.
- Taylor, C. W. 1988. Approaches to and definitions of creativity. In Sternberg, R. J., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge, UK: Cambridge University Press. 99–121.
- Varshney, L. R.; Pinel, F.; Varshney, K. R.; Schorgendorfer, A.; and Chee, Y.-M. 2013. Cognition as a part of computational creativity. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC)*, 36–43.
- Ventura, D. 2008. A reductio ad absurdum experiment in sufficiency for evaluating (computational) creative systems. In *Proceedings of the 5th International Joint Workshop on Computational Creativity*, 11–19.
- von Winterfeldt, D., and Edwards, W. 1986. *Decision Analysis and Behavioral Research*. Cambridge, U.K.: Cambridge University Press.
- Wallenius, J.; Dyer, J. S.; Fishburn, P. C.; Steuer, R. E.; Zionts, S.; and Deb, K. 2008. Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science* 54(7):1336–1349.
- Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

Evaluating digital poetry: Insights from the CAT

Carolyn Lamb, Daniel G. Brown, Charles L.A. Clarke
University of Waterloo

Abstract

We test the Consensual Assessment Technique on recent digital poetry, using graduate students in Experimental Digital Media as judges. Our judges display good interrater agreement for the best and worst poems, but disagree on others. The CAT by itself may not be suitable for use on digital poetry; however, the behavior of quasi-expert judges when attempting this task gives us clues towards evidence-based, domain-specific evaluation. We discuss the role of product-base evaluation in digital poetry, and produce a set of desiderata based on our judges' written responses: Reaction, Meaning, Novelty, and Craft.

Introduction

Evaluation is a topic of contention in computational creativity (Jordanous 2013). While various means of evaluating creativity have been proposed (Colton 2008; Jordanous 2013; Ritchie 2007), we are unaware of any rigorous validity tests of these methods. Additionally, while creativity may be domain-specific (Baer 1998), there is little in the way of domain-specific testing for computational creativity, except for ad hoc questionnaires used by individual researchers.

The Consensual Assessment Technique (CAT) (Amabile 1983) is a well-known evaluation technique from psychology, which has not been used before on digital poetry. We use the CAT protocol to ask judges to rate different poems and explain their judgments. Judges display broad agreement on the best and worst poems. Their qualitative responses illuminate the qualities associated with creativity in digital poetry. By analyzing judges' written responses, we identify four major desiderata for digital poetry: Reaction, Meaning, Novelty, and Craft. Evaluating digital poetry may be more complicated than typical uses of the CAT (e.g. children's collages) due to the heterogeneity and experimental nature of digital poetry. Our four desiderata, combined with process-based evaluation, can be used in a more standardized, evidence-based evaluation of a complex field.

We believe that paying attention to the product-based evaluations of experts is a necessary step towards full computational creativity. A truly creative computer system will be able to produce work in a creative field that is taken seriously by others in the field. Process-based evaluation will tell us about the system's techniques, but only product-based

evaluation by experts can tell us if the result is as valuable as intended.

Background

The Consensual Assessment Technique (Amabile 1983) is a method for evaluating human creativity. The idea is to assess creativity as a social phenomenon mediated by experts. A small group of expert judges (for example, visual artists to judge collages), give assessments of a group of artifacts. If the judges broadly agree in their judgments, then the assessment is considered valid. Importantly, judges are not told by what standards they should judge creativity, but are trusted to use their expert judgment.

The CAT and its reliability have been extensively studied. Baer and McKool (Baer and McKool 2009) summarize current best practices:

- Judges must possess expertise in the domain being judged; novice judges have poor interrater reliability. What constitutes expertise is a matter for debate, and can vary depending on medium. Skilled novices can have decent reliability (Kaufman, Baer, and Cole 2009), but some theoretical experts, such as psychologists, do not.
- Judges make their judgment independently, without consulting other judges.
- Judges review the artifacts blindly, without knowing framing information such as the author's identity.
- Judges are not told how to define creativity or asked to explain their ratings.
- Judges rate artifacts on a numerical scale with at least 3 points.
- Judges use the full scale. The most creative artifacts in the group should be at the top of the scale, and the least creative should be at the bottom.
- The number of judges varies from 2 to 40, with an average just over 10.
- Interrater reliability should be measured with Cronbach's coefficient alpha, the Spearman-Brown prediction formula, or the intraclass correlation method.
- An interrater reliability of 0.7 or higher is considered good. Expert judges generally achieve interrater reliabilities between 0.7 and 0.9.

- While the CAT was designed for a homogeneous group of subjects, it works in practice even when the artifacts were made under different conditions.

The CAT has become a gold standard for assessing human creativity. It has good reliability as long as the judges possess sufficient expertise. Obtaining experts is the major bottleneck in performing the CAT. However, the CAT is reliable even with relatively few (Baer and McKool 2009).

The CAT is mentioned frequently in computational creativity, but rarely implemented. Pearce and Wiggins use a modified CAT to assess chorale melodies (Pearce and Wiggins 2007). They ask their judges about the “success” of each melody rather than its creativity. More importantly, they give the judges explicit guidelines about what factors to consider when making their evaluations. Despite this, Pearce and Wiggins’ judges achieve reasonable pairwise inter-judge consistency: mean $r(26) = 0.65$.

Because of the lack of framing information, the CAT can be used only to evaluate a creative product, not the process behind it. Product and Process are two of the four perspectives from which creativity can be judged (Jordanous 2015). We will return to this topic in our Discussion section.

Jordanous’s SPECS model shares features with the CAT—particularly the use of expert raters to rank artifacts—but includes detailed training on theoretical sub-components of creativity (Jordanous 2013).

To our knowledge, CAT-related methods have not been used previously to assess computational poetry.

Method

For our study, we chose graduate students in the Waterloo’s Experimental Digital Media program (XDM) as our judges. XDM includes digital poetry among a variety of other avant-garde, multimedia art forms, and students in the program actively practice producing such art. We judged XDM students likely to understand the demands of poetry as a genre and the challenges of generating poetry with a computer.

Judges were given a set of 30 poems, in randomized order. The poems are listed in Table 1. They evaluated each poem on a scale from 1 to 5, with 1 being “least creative” and 5 being “most creative”. They produced these ratings without group discussion. Judges were told that some poems were written by computers, and others by humans, using computers; they were not told which of the poems were which.

The 30 poems, although not labeled as such, came from three different groups. Ten, group A, were poems in which we judged the authors were trying to create a relatively autonomous creative system; all but one of these poems were taken from published papers in the field of computational creativity. Another ten, group B, were poems in which the human exerted tighter artistic control (for example, by hand-crafting templates), and the computer’s role was relatively limited. The final ten poems, group C, were poems generated using specific source material which remained recognizable in the final product—either “found” poetry or modifications to a well-known human poem. All poems were published between 2010 and 2015. We presented the poems in plain text and without their titles. When the pub-

lished work was a generator producing arbitrarily many poems (for example, a Twitter bot), we provided a single generated poem. In cases where the generated poem was excessively long, a 1-page excerpt was provided. While excerpting may bias judge responses to long poems, this is relatively unimportant to our analysis.

Once all 30 poems had been rated, we began the qualitative portion of the study. Each judge was asked to go back over the poems and, for at least 3 poems, write an explanation for their judgment. We did not present this request until the judges had made all 30 quantitative ratings and did not permit them to change their quantitative answers once the qualitative portion began.

We obtained a total of seven judges, which is within normal bounds for the CAT (Baer and McKool 2009). Participation took 1 hour.

We analyzed our data in three steps. First, we calculated the intraclass correlation between the seven judges. Second, we used the Kruskal Wallis test to see if there was a difference between ratings of poems from groups A, B, and C. Third, we used open coding to determine what major factors were used by our judges in their qualitative evaluations.

Results

Interrater reliability

The intraclass correlation between our judges (a statistic that can range from -1.0 to 1.0) was only 0.18—far below the 0.7 to 0.9 standard for CAT results. A bootstrapped sample of 10,000 permuted versions of the data showed that interrater agreement hovered around zero, with a standard deviation of 0.04, meaning that if each judge gave their ratings by chance, there would be much less correlation between ratings than what our results show. The agreement between our judges, therefore, is statistically significant at $p < 0.01$. However, it is not a strong enough agreement to be used for the usual applications of the CAT, such as judging admissions to academic programs in creative fields.

Looking at the data, some poems were rated very highly by nearly all judges, while others were rated very poorly, but a large mass of poems in the middle had inconsistent or inconclusive results. When the data is reduced to those poems with the highest and lowest average scores, intraclass correlation becomes very high. With the seven best and seven worst poems—nearly half of our original data set—the intraclass correlation is 0.73, and narrowing the number of poems raises that statistic still higher. It is intuitive that poems with the highest and lowest ratings would have relatively good agreement, while poems about which judges disagreed would have average scores closer to the middle. However, when we looked at only the seven best and seven worst in each of our 10,000 bootstrap samples, the mean and standard deviation for intraclass correlation did not rise. Therefore, the agreement on the best and worst poems is not merely a statistical artifact; our judges really were able to agree on these ends of the spectrum.

It appears that our judges agree when selecting the best and worst poems in a group, but cannot reliably rank the group of poems as a whole.

Group	Title	Author	Score	Response
B	“Notes on the Voyage of Owl and Girl”	J.R. Carpenter	4.4	6
B	Excerpt from “Definitions II - Adjectives”	Allison Parrish	4.3	18
C	“Conditionals”	Allison Parrish	4.0	5
B	“trans.mission [a.dialogue]”	J.R. Carpenter	3.9	4
A	Untitled	Unnamed system (Toivanen et al. 2013)	3.6	0
B	“Walks From City Bus Routes”	J.R. Carpenter	3.6	11
B	Excerpt from “[]”	Eric Goddard-Scovel and Gnoetry	3.4	0
C	“St. Louis Blues 2011”	Christopher Funkhouser	3.3	10
A	Untitled	P.O. Eticus (Toivanen, Gross, and Toivonen 2014)	3.1	0
A	“Angry poem about the end”	Unnamed system (Misztal and Indurkha 2014)	3.1	0
B	“Good Sleep”	George Trialonis and Gnoetry	3.1	10
A	“Voicing an Autobot”	Allison Parrish	2.9	4
B	“The Ephemerides”	Allison Parrish	2.9	0
A	Untitled	IDyOt (Hayes and Wiggins)	2.9	3
C	“HaikU”	Nanette Wylde	2.8	5
C	Excerpt from “Dark Side of the Wall”	Bob Bonsall	2.8	0
B	Excerpt from “Exit Ducky?”	Christopher Funkhouser, James Bonnici, and Sonny Rae Tempest	2.7	9
C	“Spine Sonnets”	Jody Zellen	2.6	1
C	“Ezra Pound Sign”	Mark Sample	2.6	13
A	“Blue overalls”	Full-FACE (Colton, Goodwin, and Veale 2012)	2.6	5
C	“Regis Clones (Couplets from ZZT-OOP)”	Allison Parrish	2.6	6
A	“quiet”	MASTER (Kirke and Miranda 2013)	2.4	11
B	Excerpt from “Permutant”	Zach Whalen	2.2	12
A	Untitled limerick	Unnamed system (Rahman and Manurung 2011)	2.2	3
A	“The legalized regime of this marriage”	Stereotrope (Veale 2013)	2.1	7
C	“Times Haiku”	Jacob Harris	2.1	10
C	Untitled	Mobtwit (Hartlová and Nack 2013)	2.1	15
A	Untitled	Unnamed system (Tobing and Manurung 2015)	1.9	4
B	“Rapbot”	Darius Kazemi	1.9	0
C	“The Longest Poem in the World”	Andrei Gheorghe	1.7	7

Table 1: The 30 poems used in our experiment, ranked from highest to lowest average rating. The “Response” column lists how many lines of explanation, in total, were given for judges’ ratings of the poem in part 2 of the study.

Calculating Kendall’s tau between pairs of judges did not result in any useful clusterings of the judges into factions.

Kruskal-Wallis test

While a disproportionate number of the best-ranked poems were from Group B, a Kruskal-Wallis test showed that this difference was not significant ($\chi^2_2 = 3.8, 0.25 > p > 0.1$). Both relatively creative and relatively uncreative poems existed in groups A, B, and C, and no group did systematically better than others.

The individual poems, their group membership, and their average scores are shown in Table 1.

Open coding of qualitative data

We performed open coding by giving each line of written response a content label and a valence (positive, negative, or

neutral), then clustering the content labels into categories. Each category gave insight into the implicit values used by our judges. Overall there were 179 coded lines distributed over 63 explained judgments—an average of 2.8 lines per judgment and 9 judgments per reader. Table 2 shows the proportions of lines of each type and their valences.

One coder performed the initial clustering, while a second repeated the labeling to validate the first coder’s responses. Our two coders agreed on roughly half of lines as to the exact categorization, and for two thirds of lines agreed as to the general category. For the remaining one third, in half of cases, the reviewers agreed that a line could easily be coded as having both reported categories, such as cliché imagery, which has to do with both novelty and imagery. Of the remaining ones, the coders did not initially agree, but were quickly convinced of one or the other categorization.

Grouping	Category	% of comments	% Positive	% Negative
Reaction	Feeling	12%	68%	31%
Reaction	Comparison	10%	44%	44%
Reaction	Base/Other	12%	57%	29%
Meaning	Message	14%	60%	40%
Meaning	Coherence	7%	67%	33%
Meaning	Content	5%	89%	0%
Craft	Technique	12%	67%	29%
Craft	Imagery	7%	67%	33%
Craft	Form	4%	50%	50%
Craft	Skill	3%	20%	80%
Novelty	Novelty	15%	42%	58%

Table 2: Categories derived from our qualitative data.

Below we explain the meanings of each of our labels.

Reaction. 34% of lines described, structured, or contextualized the judge’s affective reaction to the poem.

Feeling. Statements about the emotions evoked in the judge by the poem.

- “This was just super fun to read.”
- “Felt empty.”

Comparison. Lines that compared the poem to something else, including existing poems or poetry movements.

- “I like how this echoes, say, Siri giving instructions.”
- “This reminds me of bad, early 2000s my space poetry... Angsty teens spewing ‘creativity’ on the world wide web.”

Base/Other. Base lines are statements that the poem is or is not creative, without immediate explanation. Often a judgment containing Base lines contained explanation in other lines, so a Base line can be thought of as a topic sentence, not necessarily an unsupported judgment. Lines coded “Other” similarly contain statements more to do with structuring a judgment than with the judgment itself, such as statements that the judge felt conflicted about the poem.

Meaning. 26% of lines described the meaning of the poem, the concepts involved, and their clarity.

Message. Statements about the idea that the judge believes the poet intended to communicate. Most judges made comments with negative valence, stating that a poem was not sufficiently meaningful. A major exception was judge 7, who left long positive comments closely interpreting the meaning of several poems.

- “It would be more creative/interesting if there were a distinct theme or repetition of some sort—some sort of message to the reader.”
- “It begins with an opinion about a campy TV show and ends on gleeful nihilism. The real American Horror Story is the nuclear apocalypse, the end of the world effected by some hideous war games between two self-obsessed nations flexing their muscles at each other. (good twerking). It packs a lot into a very compressed collection of sentences, and also manages to serve as a brutal indictment of contemporary culture.”

Coherence. Nearly all of the lines we coded as Coherence referenced a lack of coherence, or nonsense.

- “It felt too disjointed.”

Content. Statements about the characters, objects, or events in the poem. Judges mentioned this aspect of meaning less frequently than more abstract ideas.

- “It’s a complete narrative in just 3 very short lines.”

Craft. 20% of lines described the way in which the poem’s concept was executed.

Technique. Statements assessing specific literary techniques used in the poem. They include defamiliarization, enjambment, phrasing, repetition, rhyme, rhythm, vocabulary, and voice, as well as more general statements such as “playful use of language”. Poor technique, as displayed by a limited vocabulary or by the poem seeming “forced”, was coded negative.

- “I like this one because I feel like I can hear a distinct voice.”
- “I found the creative intentions - caps, quotation marks, the fragmentive narrative, the asterisks - forced and not really used well.”

Imagery. Statements commenting on the poem’s use of imagery. Imagery is a specific type of content involving direct sensory descriptions, and is important in contemporary poetry (Kao and Jurafsky 2012).

- “Good consistent imagery and figurative language.”
- “The imagery isn’t provocative.”

Form. Only a few poems received comments on their form. Three poems in inventive forms, such as imaginary dictionary entries, were praised for these concepts. Another received a comment that it was too short. In addition, two haikus received negative comments for lacking subtle features of the traditional haiku.

- “If there were another stanza I’d like it more.”

Skill. Statements assessing the poet’s skill or cleverness.

- “Very rudimentary and woe is me.”

Novelty. 15% of lines were statements about the poem’s novelty. Positive valence lines stated that the poem was unusual, unique, or subversive. Negative valence lines stated that the poem—or aspects of the poem—were obvious, derivative, unoriginal, trite, clichéd, banal, failed to push boundaries, or did not sufficiently change their source text.

- “I don’t think this is very creative because it doesn’t push the boundary of poetry in any way.”
- “This is creative because its unique. I’ve never seen a poem like this before.”

Judges frequently disagreed on what traits a poem possessed, and on the valence assigned to those traits. A poem might be described as incoherent by one judge but interestingly disjointed by another, or banal by one judge but unexpected by another. An extreme example is a poem generated by Mobtwit (Hartlová and Nack 2013). The poem was written by arranging tweets to generate emotional contrast. It was described as random and devoid of meaning by Judge 1 (who rated the poem as 1 out of 5), but Judge 7 rated the poem a 5 and gave a long exposition of its meaning (quoted above, under “message”). Restricting the sample to the seven highest and seven lowest rated poems did not remove these qualitative disagreements.

There were slightly more positive (93) than negative (79) lines overall. There was a modest positive correlation between quantitative score and number of positive comments, a similar modest negative correlation between quantitative score and number of negative comments, and no correlation at all between quantitative score and total comments ($r = 0.26, -0.27, \text{ and } 0.008$ respectively).

Discussion

Since we did not achieve the usual inter-rater reliability standard of the CAT, our method is not a finished evaluation. It is possible that the CAT will not provide standardized computational poetry evaluation at all. However, the qualitative portions of our study illuminate how judges with some expertise evaluate computational art, which leads us to a better understanding of what criteria could go into such an evaluation in the future.

Judge selection

Why did our judges disagree about poems in the middle of the set? Should we have chosen a different set of judges? We believe that our judges’ lack of interrater reliability speaks to something more complex than a simple lack of expertise.

The question of who, exactly, has sufficient expertise for the CAT is a difficult one. Kaufman et al. review prior work in the differences between expert CAT judges and novices (Kaufman, Baer, and Cole 2009). Novices lack the interrater reliability of experts and their judgments only moderately correlate with expert judges. However, in many cases, gifted novices (which Kaufman et al. describe as “quasi-experts”) produce judgments that are more in line with those of experts than with the general population. Novices have fewer problems serving as judges when the art form in question is one that the general population encounters

regularly: stories rather than poems, for instance. However, psychologists—even psychologists of creativity—are not experts; they perform as inconsistently as novices from the general population. The expertise necessary for the CAT seems to have more to do with experience in a specific creative field than with knowledge of the theoretics of creativity.

Pearce and Wiggins used both music researchers and music students as judges (Pearce and Wiggins 2007). Why did their experiment achieve close to the recommended interrater reliability while ours did not? One answer is that Pearce and Wiggins’ study was an evaluation of chorale melodies, which are simpler, less diverse, and defined by more well-established rules than computational poetry.

We argue that Experimental Digital Media students should be considered quasi-experts. Even more advanced than Kaufman et al.’s gifted novices, these students are more like experts-in-training, undergoing advanced education in how to produce art in their field. However, the field of digital poetry is too new to be well-defined. It is also possible that the different poets in our study are performing different tasks that ought not to be grouped together. The CAT’s more typical uses revolve around homogeneous products, such as the poetry or collages of elementary school students. Mature artists and researchers, in a new field where a variety of movements, motivations, and techniques are still under development, likely produce a more complex and contentious body of work.

A good idea for future work might be to replicate the CAT with other groups of experts and quasi-experts, or with a more homogeneous group of digital poems. Poets who have been paid for their published work, or participants in events such as the E-Poetry Festival (Glazier 2016), might be appropriate experts.

However, we strongly advise against the use of computational creativity researchers as expert judges unless they themselves are practicing artists in the field being studied. Computer researchers without such artistic experience are likely to have the same problem as psychologists judging human art. They may thoroughly understand the theory, but they are unlikely to have an expert sense of the *artistic* aspects of their work. Moreover, because academic publishing depends heavily on theory and argumentation, and because the field of computational creativity is so new, computer researchers (including ourselves) are likely to be distracted from evaluations of specific products by our beliefs about where we would like the field to go.

Reliance on experts

As noted, novices lack high interrater reliability, and their judgments correlate only modestly with those of efforts. In some areas, novice judgments can be uncorrelated or even negatively correlated with those of experts (Lamb, Brown, and Clarke 2015). However, some researchers have good reasons for setting a goal of popular appeal rather than the approval of experts. For these groups, techniques based on interrater reliability are not suitable, since novices lack it and experts are not the intended audience. Popular appeal should be measured through other methods, such as perhaps Jor-

danous's measurements of community impact (Jordanous, Allington, and Dueck 2015).

Judge bias

Specific to computational creativity is the possibility of judges being biased against computational art, due to pre-existing beliefs about what computers can and can't do, or to a need to connect with the imagined human author. Some researchers suggest providing framing information in order to fix this problem (Charnley, Pease, and Colton 2012). However, this bias does not always empirically appear.

Friedman and Taylor told judges either that musical pieces were composed and performed by humans or that they were composed and performed by computers. Judges' beliefs about who composed the music did not significantly moderate their enjoyment, emotional response, or interest in the music (Friedman and Taylor 2014). This was true regardless of the judges' expertise. Similarly, Norton et al. found that while individual humans can be biased for or against computers, the bias across a group was usually not statistically significant (Norton, Heath, and Ventura 2015).

Our anecdotal experience suggests that XDM students are in little danger of bias against computers. They themselves incorporate computers into their process on a daily basis. If anything the bias was in the other direction. As one judge put it after the experiment, "Sometimes I wanted to say that a poem was childish, but then I thought, 'What if a computer wrote it?' and I didn't want to hurt the computer's feelings."

Even where bias against computers exists, it is not relevant unless computer products are compared with the products of humans and the judges are somehow aware of which products are from which group. All the poems in our study had some involvement from both humans (who wrote a computer program) and computers (which put together words based on the program), but judges were not told what the computer's role was. It is easy to imagine studies where the role of the computer is more homogeneous, or even studies which compare outputs from different versions of a single program.

Desiderata for domain-specific poetry evaluation

Baer argues that creativity is an umbrella term for a variety of independent domain-specific skills (Baer 1998). If this is the case, then evaluations of computational poetry would be expected to contain criteria that apply only to poetry, perhaps only to computational poetry. Studies like ours are a step towards developing these criteria.

Our study suggests a set of desiderata shared by most of our judges for poetry:

- **Reaction.** The poem should provoke feelings of enjoyment and/or interest from the reader.
- **Meaning.** The poem should intentionally convey a specific idea. Even if the poem is difficult to understand, its difficulty should enhance the underlying meaning. (For example, a Dadaist poem uses apparently meaningless text to illustrate ideas about how language and meaning work.)

- **Novelty.** The poem should be unusual or surprising in some way, and not merely repeat familiar tropes.
- **Craft.** The poem should make effective use of poetic techniques in service of the other three criteria. This can include form, imagery, auditory effects such as rhyme, psychological effects such as defamiliarization, visual effects such as enjambment, and verbal effects such as voice. Effective use of these techniques requires skill.

These desiderata are not straightforward. In particular, some of the literary techniques praised by our judges oppose each other. At least one poem received positive comments for its detailed imagery, while another received a positive comment for simplicity. Requiring detail and simplicity at the same time is a contradiction!

We suggest viewing literary techniques as a toolbox of strategies for poetic success. Some may be more appropriate to a particular goal than others. The question asked to a judge about craft should not be, "How many times are literary techniques used?" It should be something more like, "What techniques are used, and how effective are they?" It should be assumed that such questions can only be answered by expert or quasi-expert judges.

Relations between our desiderata and existing theories

Our desiderata have overlap with other evaluation theories, but are not identical to them. For example, Novelty and Value are frequently used to evaluate creativity. Our judges did emphasize Novelty, but Value either did not appear or was divided into many sub-criteria.

Van der Velde et al. use word association to define creativity criteria (van der Velde et al. 2015). Our Novelty criterion corresponds to their Original and Novelty/Innovation, while their Skill and Craftsmanship correspond to our Craft. Van der Velde's other criteria are Emotion and Intelligence.

Ritchie suggests Typicality and Quality criteria (Ritchie 2007). A hint of Typicality can be seen in Comparison judgments. It appears that for a positive typicality judgment, a poem must strike the judge as not merely typical of poetry, but typical of *good* poetry. The "Dadaist dictionary" was rated highly, but poems typical of "the scrawl of a high school senior" were not. Groundedness in relevant poetic movements led to a positive response, but so did poems seen as entirely novel. Typicality as Ritchie conceives it may be neither necessary nor sufficient for computational poetry.

The Creative Tripod (Colton 2008) consists of Skill, Imagination, and Appreciation. While Skill as such was a minor category for us, everything under the Craft grouping presumably requires skill. Imagination was rarely mentioned, but it could be argued, as by Smith et al., that Imagination is the underlying trait which allows for Novelty (Smith, Hintze, and Ventura 2014). This reading is supported by Van der Velde et al., who group "Imagination" under Novelty/Innovation (van der Velde et al. 2015). Appreciation is difficult to read into any of the coded comments. However, the highly fluid definitions of traits in the Tripod make it difficult to definitively state if they are present or not.

Manurung et al.'s criteria of Meaningfulness, Poeticness, and Grammaticality (Manurung, Ritchie, and Thompson 2012) overlap with our desiderata. Meaningfulness and Meaning are synonymous; Poeticness and Craft are similar concepts. However, Manurung et al. operationalize Poeticness as meter and rhyme, while judges in our study had a more expansive view of Craft. Grammaticality was not emphasized; some poems received positive comments despite being quite ungrammatical.

Our Reaction criterion does not appear in many existing models, since most models focus only on qualities imputed to the poem or poet. However, it bears some resemblance to the Wellbeing and Cognitive Effort criteria of the IDEA model (Colton, Pease, and Charnley 2011), which could perhaps be used to break judge reactions down more finely.

The product or the process?

We believe that product-based evaluation is an important part of the creative process. Nevertheless, it has a major drawback: it cannot differentiate between the creativity of the computer system and the creativity of the human who programmed it.

Some examples from our data set illustrate this problem. "Notes on the Voyage of Owl and Girl", our most highly rated poem, is based on a tightly handcrafted template. The human author provides a narrative structure which does not alter, and the computer selects details (from a human-curated list) to fill it in. In its original form, "Owl and Girl" exists on a web page and is periodically re-generated before the viewer's eyes. "Owl and Girl" is interesting artistically, but its high ratings refer mostly to the creativity of the human author.

Conversely, "The legalized regime of this marriage", created by Stereotrope, is among the most poorly rated. Stereotrope is an experiment in computational linguistics. The system mines existing text for similes, produces a common-sense knowledge base using these similes, and uses the knowledge base to generate similes and metaphors of its own. The new similes and metaphors are then used to fill in templates and construct a poem. Our judges disliked this poem, calling it obvious, clichéd, unskilled, and uncreative. However, Stereotrope is doing something more *computationally* interesting than "Owl and Girl".

We must ask what the goal is for a system like Stereotrope. Do we wish to construct a system whose use of simile and metaphor is artistically successful? Then Stereotrope—in its current form—fails. But if we wish to construct a system using *humanlike* simile and metaphor, then it is easy to argue that Stereotrope succeeds: its metaphors feel obvious *because* they are humanlike. Such a system might not be artistically creative, but it might be a good model of the everyday creativity of non-artist humans expressing themselves. We will not know if a system has succeeded unless we know which of these goals it was aiming for. (Other goals than these are, of course, possible.)

We believe that ultimately, computational creativity must succeed on both fronts. To set a goal of artistic success while ignoring process is to abandon comparison to human creativity. But to set a goal of process while ignoring product is

to fail to take seriously the very medium in which the computer is working. A system which fails to take art seriously can have value as a cognitive model, but that model will not represent the cognition of skilled human artists, nor will the system's output be taken seriously by such artists.

One could argue that a computer must first establish a humanlike process before refining that process to be more artistic. This is reasonable, but debatable. It is also possible that producing good art and using a humanlike process are two tasks at which the computer can progress simultaneously. The learning process of an initially-uncreative computer may or may not look like the learning process of an initially-unskilled human, and setting a goal of behaving like an unskilled human may in some circumstances be counter-productive.

An interesting idea for future work would be to replicate the CAT study and present information about the specific tasks assigned to the computer, in a standardized form such as the diagrams in (Colton et al. 2014). CAT judges would then be asked how creative they believed *the computer* had been. An alternative would be to use one evaluation technique for product, and another for process.

Conclusion

The Consensual Assessment Technique is an established product-based creativity evaluation. We used the CAT to examine the opinions of Experimental Digital Media students, whom we consider quasi-experts, on computational poetry. The students agreed on the best and worst poems, but were divided about the ones in the middle. There was no significant bias for or against poems from the computational creativity research community.

Based on qualitative comments, we identified several evidence-based criteria through which our judges made their evaluative decisions: Reaction, Meaning, Novelty, and Craft. These might be refined for use in future evaluations.

Despite modest overall inter-rater reliability, we argue that Experimental Digital Media students are appropriate quasi-experts. Achieving consistency may be difficult for computational poetry due to its experimental and diverse nature. It is important for evaluation to take into account both product and process. Our present study does not include process components, but process could be added, either separately, or by including process information in some systematic way. We believe there is more knowledge to be gained by investigating the workings of the CAT and other expert judgment procedures.

References

- Amabile, T. M. 1983. A consensual technique for creativity assessment. In *The Social Psychology of Creativity*. Springer. 37–63.
- Baer, J., and McKool, S. S. 2009. Assessing creativity using the consensual assessment technique. *Handbook of Assessment Technologies, Methods and Applications in Higher Education* 65–77.
- Baer, J. 1998. The case for domain specificity of creativity. *Creativity Research Journal* 11(2):173–177.

- Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the Third International Conference on Computational Creativity*, 77–82.
- Colton, S.; Pease, A.; Corneli, J.; Cook, M.; and Llano, T. 2014. Assessing progress in building autonomously creative systems. In *Proceedings of the Fifth International Conference on Computational Creativity*, 137–145.
- Colton, S.; Goodwin, J.; and Veale, T. 2012. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.
- Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The face and idea descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.
- Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.
- Friedman, R. S., and Taylor, C. L. 2014. Exploring emotional responses to computationally-created music. *Psychology of Aesthetics, Creativity, and the Arts* 8(1):87–95.
- Glazier, L. P. 2016. E-poetry: An international digital poetry festival. <http://epc.buffalo.edu/e-poetry/archive/>, accessed February 25, 2016.
- Hartlová, E., and Nack, F. 2013. Mobile social poetry with Tweets. Bachelor thesis, University of Amsterdam.
- Hayes, M. D., and Wiggins, G. A. Adding semantics to statistical generation for poetic creativity. Late-breaking abstract at the *Sixth International Conference on Computational Creativity*, 2015.
- Jordanous, A.; Allington, D.; and Dueck, B. 2015. Measuring cultural value using social network analysis: a case study on valuing electronic musicians. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 110.
- Jordanous, A. K. 2013. *Evaluating Computational Creativity: a standardised procedure for evaluating creative systems and its application*. Ph.D. Dissertation, University of Sussex.
- Jordanous, A. 2015. Four PPPPerspectives on Computational Creativity. In *Proceedings of the AISB Symposium on Computational Creativity*. 8 pages.
- Kao, J., and Jurafsky, D. 2012. A computational analysis of style, affect, and imagery in contemporary poetry. In *NAACL Workshop on Computational Linguistics for Literature*, 8–17.
- Kaufman, J. C.; Baer, J.; and Cole, J. C. 2009. Expertise, domains, and the consensual assessment technique. *The Journal of Creative Behavior* 43(4):223–233.
- Kirke, A., and Miranda, E. 2013. Emotional and multi-agent systems in computer-aided writing and poetry. In *Proceedings of the Artificial Intelligence and Poetry Symposium (AISB13)*, 17–22.
- Lamb, C.; Brown, D. G.; and Clarke, C. L. 2015. Human competence in creativity evaluation. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 102.
- Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.
- Misztal, J., and Indurkha, B. 2014. Poetry generation system with an emotional personality. In *Proceedings of the Fifth International Conference on Computational Creativity*, 72–81.
- Norton, D.; Heath, D.; and Ventura, D. 2015. Accounting for bias in the evaluation of creative computational systems: An assessment of DARCI. In *Proceedings of the Sixth International Conference on Computational Creativity*, 31–38.
- Pearce, M. T., and Wiggins, G. A. 2007. Evaluating cognitive models of musical composition. In *Proceedings of the Fourth International Joint Workshop on Computational Creativity*, 73–80.
- Rahman, F., and Manurung, R. 2011. Multiobjective optimization for meaningful metrical poetry. In *Proceedings of the Second International Conference on Computational Creativity*, 4–9.
- Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.
- Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A neologism creator nomen ipsum. In *Proceedings of the International Conference on Computational Creativity*, 173–181.
- Tobing, B. C., and Manurung, R. 2015. A chart generation system for topical metrical poetry. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 308–314.
- Toivanen, J. M.; Järvisalo, M.; Toivonen, H.; et al. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the Fourth International Conference on Computational Creativity*, 160–167.
- Toivanen, J. M.; Gross, O.; and Toivonen, H. 2014. The officer is taller than you, who race yourself! Using document specific word associations in poetry generation. In *Proceedings of the Fifth International Conference on Computational Creativity*, 355–359.
- van der Velde, F.; Wolf, R. A.; Schmettow, M.; and Nazareth, D. S. 2015. A semantic map for evaluating creativity. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 94.
- Veale, T. 2013. Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the Fourth International Conference on Computational Creativity*, 152–159.

Regent-Dependent Creativity: A Domain Independent Metric for the Assessment of Creative Artifacts

Celso França, Luís Fabrício W. Góes, Álvaro Amorim, Rodrigo Rocha and Alysson Ribeiro da Silva

Programa de Pós Graduação em Engenharia Elétrica

Pontifícia Universidade Católica de Minas Gerais

Minas Gerais, Brasil 30535-901

{celso.franca,lfwgoes,alvaro.amorim,rcor,alysson.ribeiro}@pucminas.br

Abstract

Humans are the ultimate judges on how creative is an artifact. In order to be creative, most researchers agree that an artifact has to be at least new and valuable. However, metrics to evaluate novelty and value are often craft for individual studies. Even within the same domain, these metrics commonly differ. Although this variety of metrics extends the spectrum of alternatives to assess creative artifacts, the lack of domain independent metrics makes hard to compare artifacts produced by different studies, which in turn slows down the research progress in the field. In this paper, we propose an domain independent metric, called Regent-Dependent Creativity (RDC), that assesses the creativity of artifacts. This metric requires that artifacts are described within the Regent-Dependent Model, in which artifacts features are represented as dependency pairs. RDC combines the *Bayesian Surprise* and *Synergy* to measure novelty and value, respectively. We show two case studies from different domains (fashion and games) to demonstrate how to model artifacts and assess creativity through RDC. We also propose and make available a simple API to promptly use RDC.

Introduction

The beauty of computational creativity lies in its diversity, which ranges from music, culinary to science. In recent years, this myriad of possibilities has attracted researchers from across many research fields such as computer science, social sciences and arts into a quest to unveil the processes in which creativity emerges. This interaction of various disciplines led to the proposal of a plethora of creative systems, in which most of them have the ultimate goal of producing creative artifacts. The definition of what makes an artifact creative or not is still an evolving discussion, nevertheless researchers tend to agree that an artifact has to be new and valuable on a particular domain to be considered creative (Boden 2015; Colton et al. 2015; van der Velde et al. 2015). Although the concepts of novelty and value are intuitive to humans, they are not easily translated into a computer program, as they depend on individual knowledge, beliefs, tastes and cultural values (Boden 2015).

In order to tackle such a daunting task, researchers on computational creativity have proposed many solutions to assess how novel and valuable is an artifact. A popular approach is to ask what humans think about it (Lamb, Brown, and Clarke 2015; Karampiperis, Koukourikos, and

Koliopoulou 2014), as we are the ultimate judges on creativity. In this way, creative artifacts are indeed evaluated, but the human implicit mechanisms to spot creativity are still left as a black box. A more analytical approach is the use of domain specific metrics to assess novelty, value and any other features that could be related to creativity such as satisfaction, plausibility, faithfulness, generality, etc. This metrics zoo is an expected outcome due to the huge challenges imposed by the complexity that is the assessment of creativity. However, as the sub-fields in computational creativity mature, some have been converging (Góes et al. 2016; Pinel, Varshney, and Bhattacharjya 2015; Maher and Fisher 2012) to standard methods and metrics to evaluate creativity, while others are still proposing new metrics (Tomasic, Znidarsic, and Papa 2014; Schorlemmer et al. 2014; Colton et al. 2014; van der Velde et al. 2015). The latter may still be beneficial on the long run, but it makes hard to compare existing solutions between different works, consequently slowing down the progress in these particular sub-fields.

As an ambitious goal, a domain independent creativity metric would be ideal to boost scientific research on computational creativity. In fact, previous work (Maher 2010; Maher and Fisher 2012) pursued this objective, but some issues remained: i) the proposed metrics lacked implementation details, making it hard to replicate them; ii) very little or no practical examples were provided, weakening their appeal to experimental researchers; and iii) the lack of quantitative results hindered to show the effectiveness of the metric across different domains.

In this paper, we propose the Regent-Dependent model which describes artifacts as a collection of features, represented by dependency pairs. Once represented in this model, artifacts can be evaluated by our proposed Regent-Dependent Creativity (RDC) metric.

The RDC metric combines the *Bayesian Surprise* and *Synergy* to measure novelty and value, respectively. In order to address some aforementioned issues found in previous work, we present: i) two case studies from different domains; ii) a quantitative evaluation of RDC; and iii) propose and make available a simple API to the research community that implements the RDC metric.

Novelty and Value

Creative artifacts have to be novel and valuable (Boden 2004). In order to evaluate novelty, there are few metrics

based on concepts of unexpectedness, expectation and surprise that are commonly used (Grace and Maher 2014). On the other hand, value can be extracted from the associations and rules that bond individual artifacts (Varshney et al. 2013).

A novel artifact may be new only to a particular person or group. Alternatively, it may be entirely new in relation to all human history. The former type of novelty is required to achieve p-creativity (p for psychological), while the latter is concerned to h-creativity (h for historic) (Boden 2015). In practice, psychological creativity is more feasible, since it can be verified for a given dataset of known artifacts. In contrast, historical creativity imposes a dataset to have all existing artifacts, which its completeness is hard to be proved.

However, creativity is not just novelty, a creative artifact must also be valuable. Value is a measure of performance or attractiveness of an artifact which depends on its acceptance by an expert or society (Maher 2010). There are many types of value (e.g. beauty, scientific interest, musical harmony, utility etc.) (Boden 2015), and many of them are difficult to recognize, harder to put into words, and even more difficult to say clearly. For a computational model, however, it must be precisely defined (Boden 2009).

Even in science, values are often transient and sometimes changeable. The meaning of simplicity and elegance, when applied to scientific theories, is something that philosophers of science have long try - and fail - to precisely define (Boden 2004). In addition to it, if a scientific finding or hypothesis is interesting it depends on other current theories of the time and also in the social context. This is where creativity is concerned, the shock of the new can be so great that even for those who are the witnesses, it is difficult to see value in the new idea. This makes the calculation of value specific to a certain context, that is, the value of an artifact depends on the relationships between an artifact and the existing ones, more precisely, the associations of the artifacts features. When evaluating an artifact, its value is determined according to the combination of its features, which in turn are governed by rules that were implicitly created by humans in that context to value certain types of artifacts more than others. Once these rules and associations are expressed in a computer model, the value of an artifact can be determined.

Related Work

The existence of the “islands of creativity” problem, as recently highlighted by Bown (2015), suggests that a significant obstacle for the evaluation of computational creativity resides in the idea that creativity is situated on specific systems without any fluidity between these systems and the rest of the world. In fact, when it is not used a very specific metric, they appeal to random choice.

Some research tackle this problem by employing human computation to assess creativity in computer systems. Joyner et al. (2015) suggest that human computation can be an effective strategy to collect a wide variety of methods for creative tasks. From a set of existing solution methods to the intelligence test Raven’s Progressive Matrices (RPM), they developed other new methods using crowd-sourcing, highlighting those that were most different and achieved significant success. On the other hand, Lamb, Brown, and Clarke (2015) point out that the quality of a creativity metric

relies on the appropriate choice of human judges, which is addressed by the consensual assessment technique (Amabile 1982) from the field of psychology.

As opposed to the idea of using humans as judges, Cook and Colton (2015) proposed an alternative way to enable a software to make significant decisions. With the use of evolutionary algorithms which evolve short pieces of code called *preference functions*, it makes meaningful and justifiable choices between artifacts. As another approach to measure value, Jordanous, Allington, and Dueck (2015) investigate how to measure subjective and cultural value which have been expressed by members of a community towards other members. Focusing on the activity by electronic musicians on the music social network SoundCloud, they combined qualitative and quantitative research to understand and trace significant ‘valuing activities’ in Sound-Cloud data.

Maher (2010) proposed a domain independent metric to evaluate creative artifacts. It is based on novelty, value and unexpectedness. Novelty is measured as the distance from clusters of other artifacts in a conceptual space. In addition to it, value is calculated through a set of performance criteria. Finally, unexpectedness looks for variations in attributes by the use of pattern matching algorithms. Despite this research was the first to propose a domain independent creativity metric, it does not verify its applicability in real world examples.

Maher and Fisher (2012) extend the previous model proposed by Maher (2010), where creativity is evaluated based on novelty, value and surprise. In contrast to the previous unexpectedness metric, they use Bayesian inference based on prior probability for measuring the surprise of a given artifact. They suggest an application regarding the design of laptop computers.

Other work also focused on automatically assessing creativity using a creativity model, focused on aesthetics aspects, based on the probabilistic model for Bayesian inference and Shannon’s measure of entropy (Burns 2006; 2015). Bayesian inference is applied for evaluating the meaning of a given artifact based on prior evidences and the psychological arousal produced by violating expectations is modeled mathematically by Shannon’s measure of surprise (Rigau, Feixas, and Sbert 2007). The aesthetics is finally expressed as the product of Shannon-entropy measure of surprise and the Bayesian-probability measure of meaning. Some previous work have used Bayesian posterior probability or prior probability to model a notion of Bayesian surprise (Itti and Baldi 2009; Baldi and Itti 2010; Maher and Fisher 2012), instead of using the Shannon-entropy.

Similar to those previous work, we also use the Bayesian surprise metric for assessing novelty, which is known to be reasonably effective (Itti and Baldi 2009; Baldi and Itti 2010). However, in contrast to previous work, for modeling and measuring the value of a given artifact, we use concepts of synergy by extracting metrics of a graph-based knowledge representation of the artifact’s properties.

Regent-Dependent Model

A single data model to describe each artifact is imperative to allow the creativity evaluation of artifacts produced by different systems. In this paper, we propose a data model to

describe an artifact as a set of pairs between its features and their modifiers. This dependency relationship is defined by a pair $P(\text{regent}; \text{dependent})$ associated with a numeric value v . A *regent* is a feature that contributes to describe an artifact, and may be an action or attribute, while a *dependent* can change the state of an attribute or connect an action to a target. For example, an artifact *car* can be described by a pair $p_i(\text{color}; \text{blue})$, where *blue* changes the state of the attribute *color*. The same artifact could also be described by another pair $p_i(\text{drive}; \text{home})$, where the dependent *home* connects a target to the action *drive*. In a grammatical example, the famous slogan “*Just do it*” can be described by two pairs: $p_i(\text{do}; \text{just})$ and $p_j(\text{do}; \text{it})$. The first says that the adverb *just* is a modifier of the verb *do*, while the second pair connects the verb *do* to the direct object *it*.

The value v is important because it can be used to represent the intensity of a specific pair in different contexts. Different cultures have different preferences for culinary recipes, music and art. Even the science progress is weighted by social interests. Thus, the pairs can be modeled to these different situations. For example, a car with the *blue* color may be more common in certain countries than others, so the value v can be set higher than other colors.

With the definition of the presented data model, it is possible to build a *dataset* of existing artifacts and a graph of associations between the artifacts pairs which are required in our proposal to calculate novelty and value, respectively, as explained in the next sections.

Bayesian Surprise as a Novelty Metric

Novelty can only be evaluated compared to a group of existing artifacts. In this paper, we propose that novelty is calculated using a well-known metric called the Bayesian surprise (Baldi and Itti 2010), which enables to evaluate how much new is an artifact compared to existing ones in the *knowledge dataset*. This *knowledge dataset* is the description of existing artifacts, organized in instances following the Regent-Dependent model so that each instance is the representation of an artifact described by its pairs. Physically, a *dataset* is a matrix, where the rows are instances of artifacts and the columns its pairs.

The Bayesian surprise stems from the fact that a new artifact is unusual and surprising for the observer. This surprising effect is an interesting novelty detector that can be calculated by the application of Bayes’ theorem, as shown in Equation 1.

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)} \quad (1)$$

According to this theorem, the probabilities represent subjective degrees of belief in hypotheses or models that are updated as new data is acquired. Thus, the degree of conviction of an observer is represented by a subjective probability function $P(h)$ measuring the degree of belief in the observer’s hypothesis h . The term $P(h)$ is called *prior distribution*, and reflects the knowledge before new data are considered, whereas the term $P(h|d)$ is called *posterior distribution*, and as the name suggests, reflects the knowledge after consideration of a new fact d has occurred, and be inserted in the hypothesis h . Similarly, $P(d)$ is the probability that d occurs independently of the hypothesis h , and $P(d|h)$

is the probability that event d occurs given that h is true (Kruschke 2015).

Fundamentally, the effect of a new artifact is to transform an observer previous convictions in posterior convictions. Novelty thus can be quantified by considering the difference between the probability distributions that accurately describes how the world view of the observer has changed.

In fact, as shown empirically in recent studies, this approach is able to capture human notions of novelty in different types and levels of abstraction (Itti and Baldi 2009; Baldi and Itti 2010; Varshney et al. 2013). Mathematically, the novelty $n(p_i)$ of a pair p_i , regarding a specific artifact a , is calculated by Equation 2, where σ and \bar{m} are respectively the variance and average of an existing pair in the *dataset* of artifacts, and μ_i is the value associated with p_i . The total novelty N_a of a given artifact a is defined as $N_a = \sum_{p_i \in a} n(p_i)$. Equation 3 computes the normalized novelty in the range $[0,1]$, by means of an exponential normalization, where λ is a smoothing factor.

$$n(p_i) = \frac{1}{2\sigma^2} \left[\sigma^2 + (\mu_i - \bar{m})^2 \right] \quad (2)$$

$$f(N_a, \lambda) = 1 - e^{-\lambda N_a} \quad (3)$$

Synergy as a Value Metric

Strategies for assigning a value to an artifact can be widely distinct from one domain to another. Even experts from the same domain will differ comparing two or more artifacts (Boden 2015). For this reason, there are several metrics to measure value. For instance, pleasantness measures the flavor perception of a recipe (Pinel, Varshney, and Bhattacharjya 2015; Varshney et al. 2013), an aggregation metric defines the quality of a slogan (Tomasic, Znidarsic, and Papa 2014) etc. However, these metrics are designed for specific domains.

On the other hand, artifacts, even in different domains, are composed of a set of elements that have actions and attributes. This set of elements and the interaction among them is what gives value to a certain artifact. For example, a recipe consists of ingredients, each with its own taste, its texture, and its aroma, the final flavor of the recipe, however, is the result of the combined actions of its ingredients (Corning 2012). This feature takes place in other areas, such as in music in which harmony occurs when two or more pitches are combined to produce a chord (Cope 2015), or in some turn-based strategy games, where players perform individual moves with a set of elements that together implement an efficient strategy (Millington 2009).

Moreover, there are plenty of information publicly available that describes artifacts and the elements that constitute them like in *fooDB*¹. In particular, it is also available how these elements interact and what interactions are most valued in a given context or group of people, which is key to compose a valuable artifact. These facts give evidence that the relationship between the elements of an artifact can be used as a measure of value.

Our *Regent-Dependent Model* allows to represent an artifact by its elements, which in turn are described by regent-dependent pairs. This model also allows to build a graph

¹Available at <http://foodb.ca>

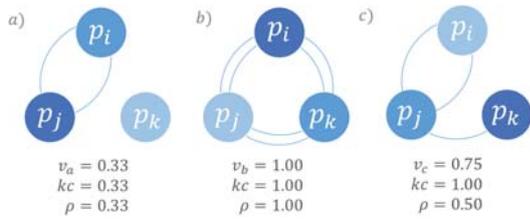


Figure 1: Relationships between pairs used to describe three different artifacts with isolated pairs (a), completely interconnected pairs (b) and reasonably associated pairs (c).

connected by pairs in which their relationship are valued in a particular context. Consequently, the most valuable artifacts in this graph are the ones which have pairs that are more interconnected among themselves.

A metric that captures this cooperative nature is synergy (Corning 2012). It measures the effect produced by various elements (forces, particles, parts or individuals) acting cooperatively in the development of emergent behaviors found in many real-world systems, such as the brain and other neural systems, stock markets, the Internet and social networking systems. The center of gravity of an object, for instance, is actually a synergistic effect, it depends upon how the combined weight of all its parts is distributed. Therefore, the value of an artifact can be measured by the synergy of the elements that exist within it.

To measure synergy, an artifact a is modeled as a graph $G_a = (V, E)$ where the vertex set V consists of pairs belonging to a and there exists an edge between two vertices p_i and p_j if and only if they belong to the same set of synergy. Each set of synergy is defined by two or more pairs that have complementary effects, i.e., they exhibit a better effect together than separately.

The graph is providential because it presents a series of metrics to measure the relationship between vertices. In particular, the *connectedness* and the *density* of the graph define fairly well a measure of synergy and the value v_a of an artifact a can then be defined by Equation 4.

$$v(a) = \frac{1}{2}kc(G_a) + \frac{1}{2}\rho(G_a) \quad (4)$$

where:

- G_a : is the graph which represents the artifact a .
- $kc(G_a)$: is the Krackhardt's connectedness of G_a .
- $\rho(G_a)$: is the density of G_a .

The first term of the Equation 4 measures the associativity among the pairs of an artifact. If all pairs are associated, i.e., all pairs are reachable from every other, then $kc(G_a)$ Krackhardt's connectedness of G_a is maximum (Krackhardt 1994). If all pairs are isolated in the artifact, then $kc(G_a)$ is minimum. The second term measures the strength of the connection among the pairs of an artifact. Basically, it measures the average number of connections between two pairs p_i and p_j . Although this is a simple measure of the relationship between vertices, the relationship described in Equation 4 can be more descriptive to contain other graph metrics such as *concentration*, *diameter* and *max flow*.

Figure 1 shows graphs of three different artifacts described by pairs p_i , p_j and p_k . Each of these pairs is a vertex and the edges between a pair and another indicates a synergistic relationship. The values v_a , v_b and v_c of respectively artifact a , b and c , calculated by Equation 4, in the range $[0,1]$, is greater when the pairs are fully connected and smaller when less associated the pairs are.

Regent-Dependent Creativity Metric

The proposed *Regent-Dependent Creativity (RDC)* metric, for calculating the creativity of an artifact a , is defined in Equation 5 as the sum of the normalized novelty n_a and value v_a plus an extra penalty term. This penalization is needed to avoid that high novelty artifacts with low value (different but useless artifacts), and high valuable artifacts with low novelty (useful but already known artifacts) are considered creative.

$$rdc(a) = n_a + v_a - p(n_a, v_a) \quad (5)$$

$$p(n_a, v_a) = s_a(1 - e^{-kd_a}) \quad (6)$$

where:

- s_a : is the sum of n_a and v_a .
- d_a : is the absolute difference of n_a and v_a .

Equation 6 penalizes an artifact depending on the difference among its novelty and its value. The greater the difference between novelty and value of an artifact, its creativity is more penalized. The penalty is more intense as the variable k is higher, however, no artifacts are penalized more than the sum of its novelty and its value. Therefore creativity is in the range $[0,2]$.

Case Studies

In this section, we show two case studies to demonstrate how to model artifacts and assess their creativity using RDC. The first case study is a simplified example from the fashion domain to evaluate apparels. The second one is from the game domain. It is based on a real and large problem to create card combos in the game HearthStone.

Fashion it: Evaluating Creative Apparels

The creation of fashion artifacts is challenging given the diversity of factors such as style, color, patterns, materials, etc (Jagmohan et al. 2014). The challenge lies both in the combination of various elements of clothing with different styles and purposes for creating a complete apparel, and in subsequently ranking them based on certain criteria.

Consider a hypothetical case where in the space of clothing items available to compose an apparel, there is only one type of shoes, one type of pants and one type of shirt, varying only the color as shown in Figure 2(a). Given this space, the process of generating creative apparels reduces to combining the colors of the clothing items available to form a complete apparel (*shoes + shirt + pants*).

Figure 2(b) shows some existing apparels that are considered interesting combinations for a fashion consultant and provides our prior knowledge.

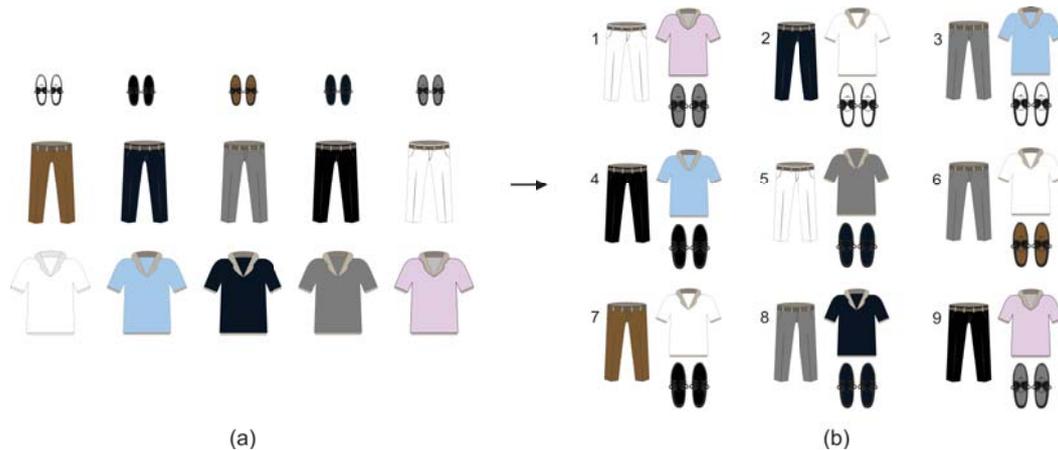


Figure 2: (a) Space of clothing items available to compose an apparel. (b) Existing apparels used to define the knowledge dataset.

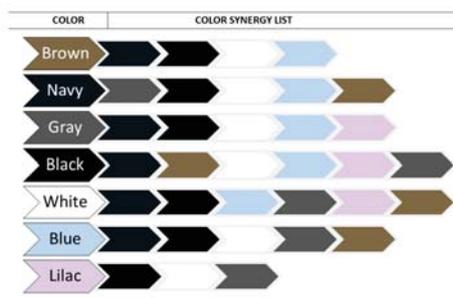


Figure 3: Color Synergy List

As the color is the only feature to be described and there are seven different color options (*white, black, navy, gray, blue, brown, lilac*), we can use the *regent* to represent the clothing item while the *dependent* is one of the available colors. Thus, the set of *regents* has three elements (*shirt, pants, shoes*) and the set of *dependents* has seven elements. These definitions guide the construction of the *dataset* where each instance is an apparel and the attributes are elements p_i of set P of all pairs used to describe all the clothing item:

$$P = \{shirt:white, shirt:navy, shirt:gray, shirt:blue, shirt:lilac, pant:white, pant:black, pant:navy, pant:gray, pant:brow, shoes:white, shoes:black, shoes:navy, shoes:gray, shoes:brow\}$$

For example, the first apparel of Figure 2(b), would be described by the pairs $(shirt, lilac) = 1$; $(pant, white) = 1$, $(shoes, gray) = 1$. There are nine known apparels in the *dataset* as described in Table 1. Note that values are set to 1, since all pairs are equally important in this example.

The creativity assessment of an artifact is made according to its novelty and value, as presented in Equation 5. The novelty of an apparel can be calculated by Equation 2, using the apparel dataset.

On the other hand, to calculate value we need to know about the synergy of colors, i.e., what color combinations

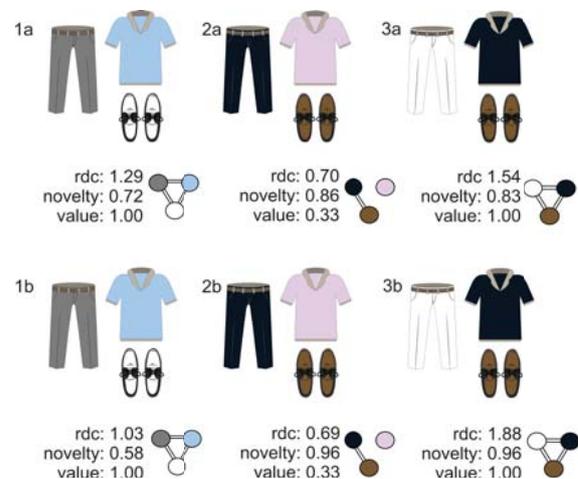


Figure 4: Behavior of the proposal metric for evaluation of different apparel.

are most valued. There are some techniques for combining colors based on color wheels, wherein a set of colors are harmonious when they fit some analogous, triad or pattern. Figure 3 illustrates a synergy list for each color based on the color wheels. The *brown* color for example, has synergy with the colors: *navy, black, white and blue*. Then, according to our proposal for the metric, an apparel would be modeled as a graph where the vertices are a clothing item and there is an edge between one clothing item and another if they have synergistic colors. With the complete graph, the value of the apparel can then be calculated by Equation 4.

Figure 4 shows the behavior of the proposed *RDC metric* in different scenarios. In apparel 1a, for instance, the colors are all synergistic, so that the graph representing that apparel is completely connected and the application of the Equation 4 returns the maximum value. The novelty, however, is penalized because it is an apparel with an existing pattern in the *dataset*.

Apparel 2a has a non existent pattern in the *dataset*, consequently achieving high novelty. On the other hand, the synergy of colors occurs only between pants and shoes,

APPAREL	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
A_1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0
A_2	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
A_3	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0
A_4	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0
A_5	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0
A_6	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
A_7	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0
A_8	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
A_9	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0

Table 1: *Dataset* of existing apparels. Each instance is an apparel and the attributes are the values v_i of the pairs used to describe each apparel. A nonexistent pair has value 0.

making the shirt a isolated vertex in the graph. The effect of this isolation is to reduce the value of this apparel to 0.33 and thereby creativity to 0.7.

Apparel 3a, in addition to exhibit a non existent pattern in the *dataset*, has synergistic colors for shirt, pants and shoes, so it has a maximum score of creativity.

In order to further explore the RDC metric, the apparels 1b, 2b and 3b of Figure 4 show the metric behavior after 10 additional inserts of the apparel 1a into the *dataset*. The apparel 1a becomes more common, reducing its novelty. However, as the *dataset* increased towards apparel 1a, the apparel patterns 2b and 3b become even more novel.

HoningStone: Evaluating Creative Combos for HearthStone

*Hearthstone*², by Blizzard Entertainment, is a DCCG in which human players compete in one-versus-one matches in alternating turns, until a player is defeated. On each turn, a player can play any cards from his hand, use his hero power or minions to attack characters (minions or hero) and particularly combining cards, that is, playing *combos*. Thus, a combo is a group of related cards played in the same turn. In (Góes et al. 2016), a computational creativity system, called HoningStone, was proposed. It automatically generates creative card combos for *Hearthstone* based on the Honing theory of creativity (Gabora 2010). HoningStone used a creativity metric based on surprise and efficiency to generate and evaluate combos. These metrics used a dataset of 31000 distinct combos extracted from real game logs from 10000 decks played in more than 3 million matches obtained from the various public websites.

In this paper, we use the same knowledge dataset to model and evaluate a few card combos generated by HoningStone using RDC. We show how to use synergy as a value metric instead of efficiency.

Each combo is composed of cards, which in turn has effects. Each effect, described in the card's text, is modeled as a pair $P(ability, target)$ which has a value v . In a card which the text is "destroy 2 minions", for instance, it is represented as $P(destroy, minion) = 2$. *Hearthstone* produces 190 distinct pairs when combining all abilities and targets from the existing card set (Góes et al. 2016). Thus, the prior knowledge is composed by 31000 combos, each one represented by those effect pairs extracted from each card. A card c_i is synergistic to another card c_j when they have complementary pairs, i.e., the combined effect of the comple-

²Available at <http://us.battle.net/hearthstone/en/>

Card	Pairs
C_1	a(plusAttack, alliedMinion) = 2 b(giveCharge, alliedMinion) = 1
C_2	c(enrage, self) = 1 d(plusAttack, self) = 1
C_3	e(dealDamage, character) = 1
C_4	f(dealDamage, minion) = 1 g(plusAttack, minion) = 2
C_5	g(enrage, self) = 1 i(plusAttack, self) = 1
C_6	j(plusAttack, self) = 1 k(plusHealth, self) = 1 l(drawCard, ownHand) = 1

Table 2: A subset of six cards and their respective pairs.

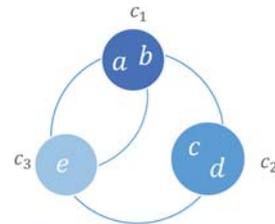


Figure 5: Associations between the effects pairs of cards c_1 , c_2 and c_3 .

mentary pairs produces greater advantages than when played separately. Figure 5 shows the pairs' relationship for the cards c_1 , c_2 and c_3 , while Table 2 lists pairs of each of these cards.

For example, card c_2 and c_3 are synergistic as c_2 enrage effect, represented by pair c , is activated only when this card takes damage. In addition to it, pair e of card c_3 works as trigger that deals damage to card c_2 , binding c_2 and c_3 . This combination of cards and their complementary effects that makes a combo stronger. The same type of associations can be made to all other cards and effects. The more associations a card has to another, higher is the synergy.

Figure 6 shows the novelty, value and RDC for three combo examples, generated with HoningStone, using cards c_1 , c_2 , c_3 , c_4 , c_5 and c_6 . Novelty is calculated using the

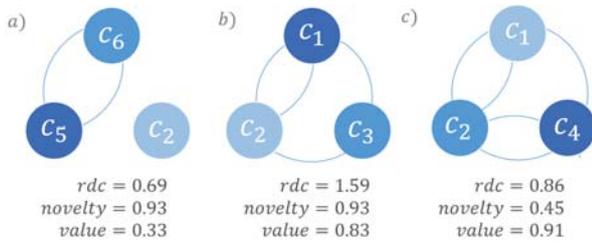


Figure 6: Behavior of the RDC metric for evaluation of different combos.

knowledge dataset of 31000 combos, and synergy uses a simplified set of associations which covers all the effects in cards from c_1 to c_6 . Combo b is novel according to the knowledge database and also has a high synergy, leading to a high RDC. On the other hand, combo a presents low value and high novelty. This gap penalizes creativity since it is a new combo but not very effective.

Regent-Dependent Creativity API

In order to complement the presented case studies, an API for evaluating artifacts was developed³. Only two inputs are required to evaluate an artifact: a knowledge database that contains existing artifacts of a particular application domain, and a set of relations that represent the synergy among the artifacts' attributes. The knowledge database must contain artifacts encoded in JSON format. In the first example, where clothing items are combined to form an apparel, the knowledge database has the following format:

```
[
  {
    "clothingItems": [
      {
        "type": "SHIRT",
        "color": "LILAC"
      },
      {
        "type": "PANTS",
        "color": "WHITE"
      },
      {
        "type": "SHOES",
        "color": "GRAY"
      }
    ]
  },
  ...
]
```

A specialized parser is responsible for converting the encoded knowledge database into a collection of instances of artifact objects. The decoded collection of artifact objects is depicted below:

```
{
  "1": [0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0],
  "2": [1,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0],
  ...
  "9": [0,0,0,0,1,0,1,0,0,0,0,0,0,0,1,0]
}
```

Relations representing the synergy of the artifacts are structured as a map between each attribute and its respective synergistic attributes. These relations are illustrated in figure 3, describing the synergy among the colors and their clothing items. The API supports the synergistic relations to be represented as follows:

³Source-code for the Regent Dependent Creativity API: <https://github.com/CreaPar/rd-creativity-metric-api>

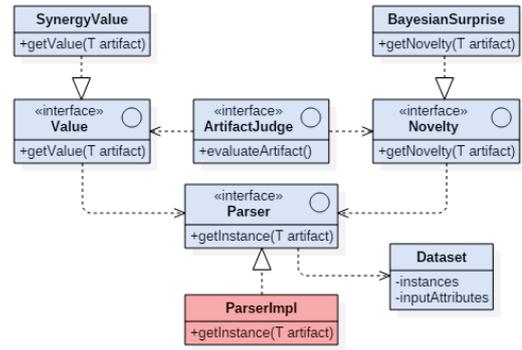


Figure 7: Class diagram

```
{
  "WHITE": ["NAVY", "BLACK", "BLUE", "GRAY", "LILAC", "BROWN"],
  "BLACK": ["NAVY", "BROWN", "WHITE", "BLUE", "LILAC", "GRAY"],
  "NAVY": ["GRAY", "BLACK", "WHITE", "BLUE", "BROWN"],
  "BLUE": ["NAVY", "BLACK", "WHITE", "GRAY", "BROWN"],
  "GRAY": ["NAVY", "BLACK", "WHITE", "BLUE", "LILAC"],
  "BROWN": ["NAVY", "BLACK", "WHITE", "BLUE"],
  "LILAC": ["BLACK", "WHITE", "GRAY"]
}
```

When the parser loads the knowledge database, it computes the mean and variance of each attribute among all loaded artifacts. This information is useful for the calculation of the RDC metric. The two main classes responsible for the Regent-Dependent creativity metric are: the *SynergyValue* class, responsible for calculating the value metric, in which the method *getValue(T artifact)* will return the synergistic value of the artifact given as parameter; and the *BayesianSurprise* class, responsible for calculating the novelty metric, by using the method *getNovelty(artifact T)*. With a measure of novelty and value, the *evaluateArtifact()* method in *ArtifactJudge* class, judges how creative is an artifact according to Equation 5 using RDC. Figure 7 shows the implementation details of the API.

Conclusion

Despite the proposal of several metrics to assess the creativity of artifacts, still computational creativity lacks metrics that can be used across different domains. This paper addresses this issue by proposing the Regent-Dependent Creativity (RDC) metric, based on the *Bayesian surprise* and *synergy* to measure novelty and value. The presented results show the use of RDC in two different domains: fashion and games. The fashion case study is simplified but is a throughout example to show each step to model and use RDC. The second one is a real world example to show that the model is applicable to larger problems. This paper also presented an API, which is available online, with a full example so researchers can promptly use RDC to evaluate artifacts.

As future work, RDC can be used into several other domains, such as culinary, arts, music etc. RDC can also be used as a creativity metric to guide the generation of artifacts. The API can be extended to accommodate other metrics and filled up with more examples. In addition to it, we can validate RDC using human experts to assess creativity through techniques such as Consensual Assessment and human computation. We hope that RDC helps the computational creativity community to boost progress in this challenging research field.

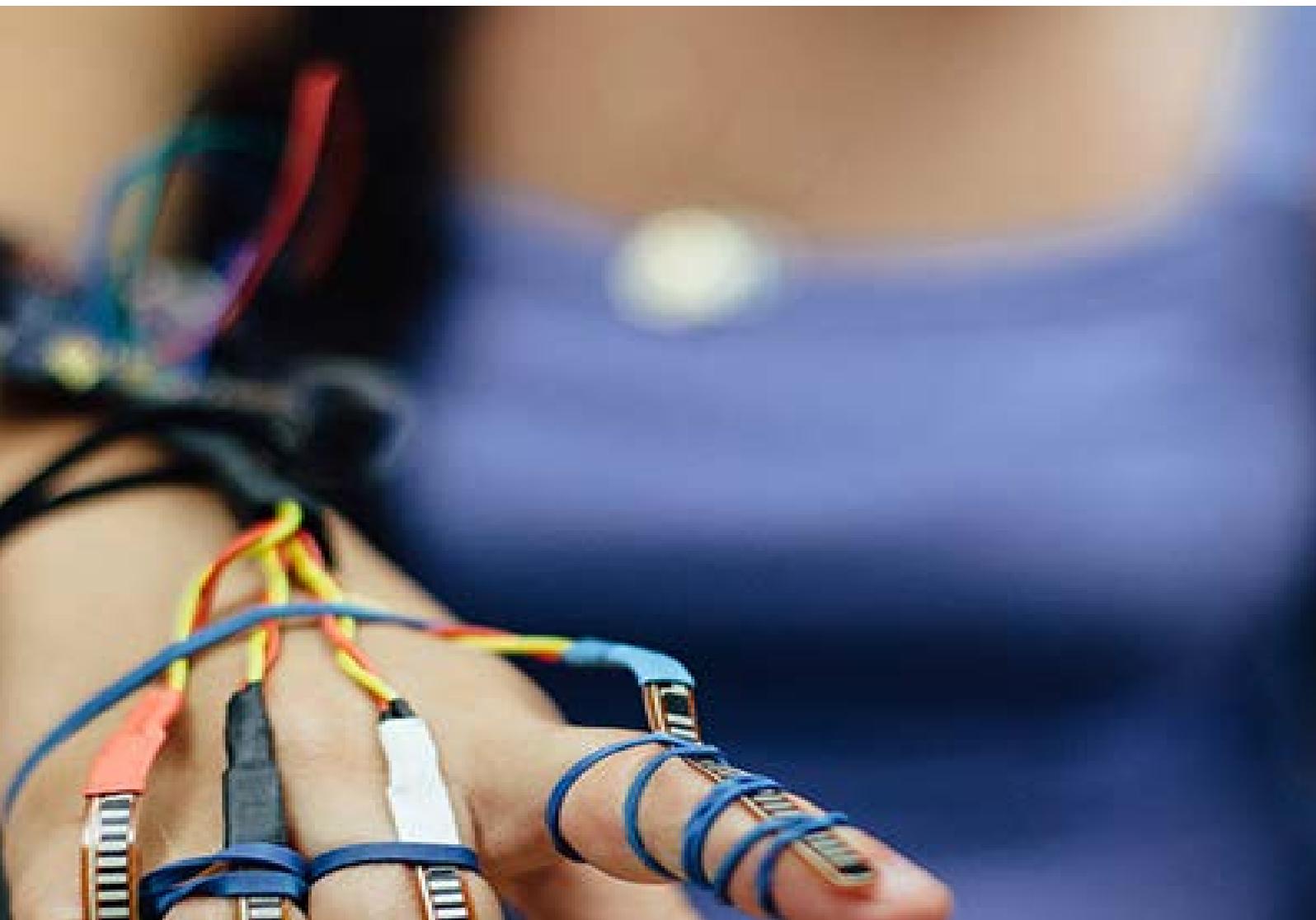
Acknowledgment

We would like to thank FIP PUC Minas, FAPEMIG, CNPq and CAPES to support this work.

References

- Amabile, T. M. 1982. The Social Psychology of Creativity: A Consensual Assessment Technique. *Journal of Personality and Social Psychology* 43(5):997–1013.
- Baldi, P., and Itti, L. 2010. Of bits and wows: A bayesian theory of surprise with applications to attention. *Neural Networks* 23(5):649–666.
- Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. London: Routledge, 2 edition.
- Boden, M. a. 2009. Computer models of creativity. *AI MAGAZINE* 13(2):72–76.
- Boden, M. A. 2015. Creativity and ALife. *Artificial Life* 21(3):354–365.
- Bown, O. 2015. Attributing creative agency: Are we doing it right? In *Int. Conf. on Computational Creativity*, 17–22.
- Burns, K. 2006. Atoms of eve’: A bayesian basis for esthetic analysis of style in sketching. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 20(03):185–199.
- Burns, K. 2015. Computing the creativeness of amusing advertisements: A bayesian model of burma-shave’s muse. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 29:109–128.
- Colton, S.; Pease, A.; Corneli, J.; Cook, M.; and Llano, T. 2014. Assessing progress in building autonomously creative systems. In *Int. Conf. on Computational Creativity*, 137–145.
- Colton, S.; Pease, A.; Corneli, J.; Cook, M.; Hepworth, R.; and Ventura, D. 2015. Stakeholder groups in computational creativity research and practice. In *Computational Creativity Research: Towards Creative Machines*. Springer. 3–36.
- Cook, M., and Colton, S. 2015. Generating code for expressing simple preferences: Moving on from hardcoding and randomness. In *Int. Conf. on Computational Creativity*, 8–16.
- Cope, D. 2015. Computational creativity and music. In *Computational Creativity Research: Towards Creative Machines*. Springer. 309–326.
- Corning, P. 2012. *Nature’s Magic: Synergy in Evolution and the Fate of Humankind*. Cambridge University Press.
- Gabora, L. 2010. Revenge of the neurds: characterizing creative thought in terms of the structure and dynamics of memory. *Creativity Research Journal* 22(1):1–13.
- Góes, L. F. W.; da Silva, A. R.; Rezende, J.; Amorim, A.; França, C.; Zaidan, T.; Olímpio, B.; Ranieri, L.; Morais, H.; Luana, S.; and Martins, C. A. P. S. 2016. Honingstone: Building creative combos with honing theory for a digital card game. *IEEE Transactions on Computational Intelligence and AI in Games* PP(99):1–1.
- Grace, K., and Maher, M. L. 2014. What to expect when you’re expecting: the role of unexpectedness in computationally evaluating creativity. In *Int. Conf. on Computational Creativity*, 120–128.
- Itti, L., and Baldi, P. 2009. Bayesian surprise attracts human attention. *Vision Research* 49(10):1295–1306.
- Jagmohan, A.; Li, Y.; Shao, N.; Sheopuri, A.; Wang, D.; and Varshney, L. R. 2014. Exploring Application Domains for Computational Creativity. In *Int. Conf. on Computational Creativity*, 328–331.
- Jordanous, A.; Allington, D.; and Dueck, B. 2015. Measuring cultural value using social network analysis: a case study on valuing electronic musicians. In *Int. Conf. on Computational Creativity*, 110–117.
- Joyner, D. A.; Bedwell, D.; Graham, C.; Lemmon, W.; Martinez, O.; and Goel, A. K. 2015. Using human computation to acquire novel methods for addressing visual analogy problems on intelligence tests. In *Int. Conf. on Computational Creativity*, 23–30.
- Karampiperis, P.; Koukourikos, A.; and Koliopoulou, E. 2014. Towards machines for measuring creativity: The use of computational tools in storytelling activities. In *Int. Conf. on Advanced Learning Technologies*, 508–512.
- Krackhardt, D. 1994. Graph theoretical dimensions of informal organizations. In Carley, K. M., and Prietula, M. J., eds., *Computational Organization Theory*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc. 89–111.
- Kruschke, J. 2015. *Doing Bayesian data analysis : a tutorial with R, JAGS, and Stan*. Boston: Academic Press.
- Lamb, C.; Brown, D. G.; and Clarke, C. L. 2015. Human competence in creativity evaluation. In *Int. Conf. on Computational Creativity*, 102–109.
- Maher, M. L., and Fisher, D. H. 2012. Using AI to evaluate creative designs. In *Int. Conf. on Design Creativity*, 45–54.
- Maher, M. L. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *Network Conference on Creativity and Innovation in Design*, 22–28.
- Millington, I. 2009. *Artificial Intelligence for Games*. CRC Press.
- Pinel, F.; Varshney, L. R.; and Bhattacharjya, D. 2015. A culinary computational creativity system. In *Computational Creativity Research: Towards Creative Machines*. Springer. 327–346.
- Rigau, J.; Feixas, M.; and Sbert, M. 2007. Conceptualizing birkhoff’s aesthetic measure using shannon entropy and kolmogorov complexity. In *Computational Aesthetics*, 105–112.
- Schorlemmer, M.; Smaill, A.; Kai-uwe, K.; Kutz, O.; Colton, S.; Cambouropoulos, E.; and Pease, A. 2014. COINVENT : Towards a Computational Concept Invention Theory. In *Int. Conf. on Computational Creativity*, 288–296.
- Tomasic, P.; Znidarsic, M.; and Papa, G. 2014. Implementation of a Slogan Generator. In *Int. Conf. on Computational Creativity*, 340–343.
- van der Velde, F.; Wolf, R. A.; Schmettow, M.; and Nazareth, D. S. 2015. A semantic map for evaluating creativity. In *Int. Conf. on Computational Creativity*, 94–101.
- Varshney, L. R.; Pinel, F.; Varshney, K. R.; Schorgendorfer, A.; and Chee, Y.-M. 2013. Cognition as a part of computational creativity. In *IEEE Int. Conf. on Cognitive Informatics and Cognitive Computing*, 36–43.

INTERACTION



Modes for Creative Human-Computer Collaboration: Alternating and Task-Divided Co-Creativity

Anna Kantosalo and Hannu Toivonen

Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
anna.kantosalo@helsinki.fi, hannu.toivonen@cs.helsinki.fi

Abstract

The analysis of human-computer co-creative systems in current literature is focused on a human perspective, highlighting the benefits of co-creative systems for human users. This study paper examines different styles of human-computer co-creation from a more computational perspective, presenting new concepts for analysis of computational agents in human-computer co-creation. Our perspective is based on Wiggins' formalization of creativity as a search. We formalize for co-creative scenarios involving an alternating, iterative approach to co-creation, which we call *alternating* co-creativity and briefly discuss its non-alternating counterpart, *task-divided* co-creativity. With focus on alternating co-creativity, we analyze the co-creative process and discuss new modes and roles for the creative agents within it. Finally, we illustrate our theoretical findings in the context of current co-creative systems and discuss their relation to the roles and expectations presented in current literature.

Introduction

Human-computer co-creativity, a form of collaborative creativity between a human and a computational agent is a topic gaining more and more interest in various domains. Especially interaction designers have been interested in human-computer co-creativity, in order to develop better creativity support systems. In these systems, computational agents are often seen as mere tools (see e.g. Lubart (2005), Maher (2012), McCormack (2008)). As computational creativity researchers we are interested in how the computer can take the role of a more equal partner in the co-creative process.

To be able to facilitate the study of this partnership from a computational perspective, we need concepts and language to discuss the properties of computationally creative agents and frameworks to analyze them further. In this paper, we first look at human-computer co-creativity from a human-centered perspective common in current literature. We see what kind of roles have been commonly taken, or shared, by humans and computational agents and how creative responsibility has been shared in previous projects. We then assume a more computationally oriented perspective, and revisit Wiggins' framework of creativity as a search.

We propose a means for extending Wiggins' framework to human-computer co-creativity that allows for both sys-

tem and agent level analysis of co-creativity. On the system level, we focus on what we call *alternating co-creativity*, an iterative setting, where a human and a computational agent take turns in constructing and modifying a single creative artifact, or concept. We also briefly consider an alternative scenario, in which the human and the computational agent perform specific creative sub-tasks. We call this *task-divided* co-creativity. On the agent level, we focus on *complete* agents, which themselves form complete systems under Wiggins' formalization and are thus capable of *alternating* co-creativity as opposed to *incomplete* agents, which are only capable of *task-divided* co-creativity.

Our formalization of *alternating* co-creativity focuses on a pairwise case involving only one human and one computational agent, although the setting generalizes to more than two participants. The collaboration typically starts from scratch and the aim of the participants is to create and converge into a result that satisfies both parties. With the framework we analyze a number of potentially challenging scenarios in *alternating* co-creativity to achieve a more balanced human-computer co-creative partnership. Finally, we illustrate the framework in the context of some current co-creative systems, highlighting different modes and roles in *alternating* and *task-divided* co-creativity.

Human-Computer Co-Creation from a Human-Centric Perspective

Current literature on human-computer co-creativity is focused on a human perspective and on how computational agents can support human creativity. This is a noble goal, but often seems to reduce the computational agent into the role of a tool as opposed to an individual creator.

The concept of computational agents as a tool is well illustrated by Lubart's (2005) classification of creative computational partners into four roles:

1. Computer as a Nanny: The computer manages user's work and time spent on creative tasks and takes on routine tasks such as saving and information presentation.
2. Computer as a Pen-Pal: The computer facilitates information flow between the artist and the audience, or other human co-authors.
3. Computer as a Coach: The computer can advice the user

of creativity-inducing techniques to stimulate the user's creative process.

4. Computer as a Colleague: The computer can be creative in itself, "or contribute new ideas in a dialogue with humans".

The same focus on computational agents as tools can also be seen in a more recent article by Maher (2012). She examined the question "Who's Being Creative?" in the context of co-creative ideation and described three roles for the computers: support, enhance and generate. Computers in the support role provide the human with tools and techniques for supporting creativity. The computer as an enhancer extends the creative abilities of the human user by providing knowledge or encouraging creative cognition. Finally, the computer as a generator will provide the user with creative ideas to interpret, evaluate and integrate as creative products.

There is a great overlap between the roles suggested by Maher and Lubart, although the exact equivalence seems to depend on the skill level of the human participant. If we assume a naive human creator, Maher's support role is similar to Lubart's Nanny or Pen-Pal roles. The enhance-role becomes parallel to Lubart's coach role, and the generator role is similar to the role of a colleague.

The two classifications differ most in the role of the human. Maher defines two roles for the human: to model and to generate. The first role describes a human who defines the computational models an processes of the computational agent, while in the second the human is facilitated or enhanced by a computer. Lubart does not explicitly define any roles for the human, but the human is seen as essential for evaluation and fine tuning of creative ideas, while Maher allows also for a more audience-like role, where the human only interprets the final artifact. This allows for an interpretation in which Maher's computational generator can be slightly more independent compared to Lubart's colleague.

Similarly to Lubart, McCormack (2008) implicitly represents the human in the role of a final evaluator in his article. He envisions a future where machines will enable "modes of creative thought and activity currently unattainable" while the human is still an essential part of the creative process. His vision describes creative systems fulfilling the role of an instrument; again the computational agent is seen as an interactive tool with creative potential for the human to master. Burlison (2005) talks about a more balanced relationship between the human and the computational partner, and considers that a hybrid human-computer system may enhance both human and computer capabilities.

Where the strength of the human is seen to lie in evaluation, the strength of the computer is seen in performing mundane tasks fast: Yannakakis et al. (2014) consider that in mixed-initiative game level co-creation, computational agents can improve human creativity by offering lateral thinking aids (fresh stimuli), diagrammatic reasoning aids (pictorial presentations for aiding the creative process) and searching massive search spaces quickly for novel and useful concepts.

Both Maher's and Lubart's classifications also clearly show how a creativity support system does not necessarily

need to be creative in order to be able to offer valuable support to the creative process. It is easy to imagine systems fulfilling multiple roles of either classification without any system components designed specifically to contribute creatively. Similarly, McCormack's vision of new instruments and the tasks presented by Yannakakis et al. do not necessarily require autonomous creative capability from the system.

The role of the computer is also defined by the needs of the user of the co-creative system: Lubart (2005) refers to earlier work by Bonnardel and Marmche, who concluded that the user's level of expertise affects what kind of computer support is most helpful for the user. Nakakoji (2006) has similar considerations, as he classifies the role of computational creativity support systems to "dumbbells", "running shoes" and "skis" based on whether the user needs to develop her creative capability, create faster, or if she needs new ways to create that go beyond her own capabilities.

Finally, human-computer co-creativity can take place in multiple configurations: According to Maher (2012), both humans and computers may participate in co-creation as individuals, in groups of humans vs. teams of computational agents, or as a part of the human society vs. a computational multi-agent society. However she notes that most interaction in current systems seems to happen between an individual human and a computer, whereas interactions between societies of humans and agents are nonexistent.

Formalization of Alternating Co-creativity

We focus on cases where one human and one computational agent collaborate in co-creation, as this is currently the most common case presented in literature. We define *alternating co-creativity* as co-creativity in which the co-creative partners take turns in creating a new concept satisfying the requirements of both parties. As a sister term, we define *task-divided co-creativity* as co-creativity in which the co-creative partners take specific roles within the co-creative process, producing new concepts satisfying the requirements of one party. We focus on the first which we consider more interesting as it puts the human and the computational agent in a more equal position.

Under the surface, the goals of the participants in *alternating co-creativity* are much deeper than just generating an artifact. Only in trivial cases will both parties agree from the outset on what is relevant and interesting. Instead, in the interesting cases, to reach an agreement they will need to modify their views and opinions.

For the human participant, this is a chance to get new inspirations and reach artifacts she could not have reached otherwise, potentially expanding her capabilities. For the computational participant, the setting offers both motivation and resources for transformational creativity (see below for more): transformational creativity is needed in order to reach a result that satisfies both the human and the computational agent and input from the user can be used to guide the transformations.

We will build our description and analysis of the two modes, *alternating* and *task-divided*, on Wiggins' (2006) formulation of creativity as search.

The Creative Systems Framework

Wiggins (2006) gives a generic framework for describing creative systems as search; we give a brief overview of the concepts and notation here, with our interpretation, as well as some simplifying notation. For full details, we refer to Wiggins (2006).

A creative system operates in some space \mathcal{U} of concepts or artifacts. For instance, for a poetry writing system, this universe \mathcal{U} could consist of all possible sequences of words. (Wiggins' formulation of the universe can be understood more broadly, but we find it useful that \mathcal{U} specifically denotes the space where the system can technically operate.) Our example poetry system can deal with sequences of words even if they are not poem-like, but it cannot handle melodies or pictures even if they had poetical properties.

A set \mathcal{R} of rules defines the actual search space within the universe by specifying which artifacts are valid in the system's view. A poetical system imposes constraints on the structure and grammaticality of word sequences and the system aims to find sequences that are considered valid poems. An interpretation function $[[\cdot]]$ applies the rules on concepts, yielding real numbers between $[0, 1]$. Assuming a threshold for admitting valid concepts, we denote the valid subspace of \mathcal{U} by

$$R \equiv [[\mathcal{R}]](\mathcal{U}). \quad (1)$$

(Wiggins denotes the same set by \mathcal{C} .)

Another set \mathcal{E} of rules evaluates concepts in the universe for their quality or value. In the case of poetry, the quality could be related e.g. to the contents and meaning of the poetry. We denote the subspace of \mathcal{U} evaluated favorably by

$$E \equiv [[\mathcal{E}]](\mathcal{U}). \quad (2)$$

The goal of the system can now be stated simply as creating—or finding, using the search metaphor—concepts in $R \cap E$.

How the system searches the space is defined by a set \mathcal{T} of traversal rules. Another interpretation function $\langle\langle\cdot\rangle\rangle$ applies the traversal rules \mathcal{T} to move from a point (concept) c^i in \mathcal{U} to a new point c^{i+1} :

$$c^{i+1} \equiv \langle\langle\mathcal{R}, \mathcal{T}, \mathcal{E}\rangle\rangle(c^i). \quad (3)$$

Since the aim is to satisfy \mathcal{R} and \mathcal{E} , the actual traversal is naturally informed by them. In general, the input c^i and output c^{i+1} can be sets of concepts.

To ease discussion of concept sets reached from a particular concept c^i , we use $T^n(c^i)$ to denote the set of concepts reachable in at most n recursive applications of the traversal step of Equation 3:

$$T^n(c^i) \equiv \bigcup_{j=0}^n \langle\langle\mathcal{R}, \mathcal{T}, \mathcal{E}\rangle\rangle^j(c^i). \quad (4)$$

Let c_\emptyset denote the empty concept and assume that it is always a member of \mathcal{U} . When a system has no existing concepts to start from, the space reachable to it is now denoted by $T^\infty(c_\emptyset)$. The set of valid and valued concepts that the system can generate can now be expressed simply as $T^\infty(c_\emptyset) \cap R \cap E$.

In our setting, the human and computational agent take turns in modifying a single concept. We use $t(\cdot)$ to denote a single traversal step, taken according to \mathcal{T} , \mathcal{R} and \mathcal{E} , omitted from the notation for simplicity, and returning a single concept:

$$t(c^i) \equiv \max_{\mathcal{R}, \mathcal{E}}(\langle\langle\mathcal{R}, \mathcal{T}, \mathcal{E}\rangle\rangle(c^i)),$$

where $\max_{\mathcal{R}, \mathcal{E}}(\cdot)$ denotes selection of a single item according to how high it evaluates on \mathcal{R} and \mathcal{E} .

Transformational creativity (Boden 1992) takes place when the system changes its own conception of which concepts are valid (by modifying \mathcal{R} and thereby R) or its method of traversing the space (by modifying \mathcal{T}) (Wiggins 2006), or its standards (by modifying \mathcal{E} and thereby E).

Alternating and Task-Divided Human-Computer Co-Creativity

Interpreted through Wiggins' framework, a creative system consisting of two collaborating parties, a human and a computational agent, aims in principle to create artifacts in the intersection $R \cap E$ of its R and E just like any other system.

We use Wiggins' creative systems framework, however, to describe each agent separately. This allows us to analyze the capabilities and roles of the agents, and to characterize various issues in co-creation. We use subindices h and c to denote the human and computer parts as follows:

\mathcal{U}_h	\mathcal{U}_c	Sets of all possible concepts that the human and the computer can process
R_h	R_c	Sets of valid concepts
E_h	E_c	Sets of appreciated concepts
$T_h^n(c)$	$T_c^n(c)$	Sets of concepts reachable in n steps from c
$t_h(c)$	$t_c(c)$	The concept produced after c .

The above sets are defined by the respective rules $\mathcal{R}_h, \mathcal{E}_h, \mathcal{T}_h, \mathcal{R}_c, \mathcal{E}_c, \mathcal{T}_c$. Note also that the traversal (e.g. $T_c^n(c)$ and $t_c(c)$) depends in practice also on the history of already generated/seen concepts, not only the most recent one, c .

Alternating Co-creation In *alternating* co-creation we assume that each party takes turns in co-authoring a single joint concept. Using the above notation, alternating co-creation can be described as cycles of

$$c_c^i = t_c(c_h^{i-1}) \quad \text{and} \quad c_h^{i+1} = t_h(c_c^i) \quad (5)$$

where the subscripts h and c are used to denote the concept creator and superscript i the relative order of the created concepts c .

The universe where both parties can operate is $\mathcal{U}_h \cap \mathcal{U}_c$. The goal of alternating co-creation is to produce concepts that satisfy both parties, thus the valid and appreciated sets of artifacts of the system are also characterized by the respective intersections, $R_h \cap R_c$ and $E_h \cap E_c$, respectively.

In the interesting cases, the goal is not simply to find concepts in the intersection of the mutual initial sets of valid and appreciated concepts $R_h \cap R_c \cap E_h \cap E_c$. For instance, some of the pairwise intersections may be empty. E.g., if

$E_c \cap R_h = \emptyset$ then the computer does not appreciate any concepts considered valid by the human, and the task has no solution. Therefore, the parties will need to be able to transform their rule sets so that they can find a solution that has become acceptable to both, leading to transformational creativity. We will return to this in the later sections.

We define two further modes of alternating co-creativity: *symmetric* and *asymmetric*. When the computational agent encounters a situation where its rules do not allow it to operate further, there are two ways to continue co-creation: transform the rules to adapt to the new situation, or skip turns during the process. If an agent uses transformational creativity to solve conflicts instead of skipping turns, we say it is capable of *symmetric alternating* co-creativity. If it uses skipping instead, we say it is only capable of *asymmetric alternating* co-creativity.

Even in the case where transformations are not needed, alternating co-creation may help either party reach areas they could not have reached otherwise. For instance, if the new concept $c_c = t_c(c)$ returned by the computer was not reachable for the human, i.e., $c_c \notin T_h^n(c)$ for any reasonable n , then the human user gains access to new concepts.

Task-Divided Co-creation In *task-divided* co-creativity, the human and the computational agent do not take equal turns in creating a concept together, but instead the task of creating a concept is divided into specific subtasks within Wiggins' (2006) formulation. These tasks include defining the conceptual space (\mathcal{R}), defining the value of the system (\mathcal{E}), and generating new concepts within the system (\mathcal{T}).

Task-divided co-creativity can be performed by *incomplete* creative agents, restricted systems which are incapable of defining their own concepts (missing \mathcal{R}), evaluating their concepts (missing \mathcal{E}), or generating concepts (missing \mathcal{T}). For instance, some creative systems use genetic algorithms to search a space of possibly interesting concepts, but outsource the evaluation \mathcal{E} (the fitness function) to the user. In contrast, we define a *complete* agent as one which has its own \mathcal{R} , \mathcal{E} and \mathcal{T} and is therefore able to take part in alternating co-creation.

Formally *task-divided* co-creativity can be defined as a search performed in \mathcal{U} by an interpretation function utilizing either \mathcal{R}_h or \mathcal{R}_c for defining concepts, \mathcal{T}_h or \mathcal{T}_c for search within \mathcal{U} , and \mathcal{E}_h or \mathcal{E}_c for evaluation of the concepts depending on the division of tasks between the human and the computational agent. The interpretation function could then take for example the following form:

$$c^{i+1} = \langle\langle \mathcal{R}_c, \mathcal{T}_c, \mathcal{E}_h \rangle\rangle(c^i).$$

Obviously, real systems rarely fall into rigid categories such as alternating or task-divided, or complete or incomplete, just like their rule sets \mathcal{R} , \mathcal{E} rarely produce crisp sets of valid concepts. We believe, however, that the concepts here and the following analysis of challenges are useful for a better understanding of different possible roles, issues and opportunities in human-computer co-creation.

Computational Challenges in Alternating Co-Creativity

With the formalization of *alternating* co-creativity, we can analyze some potential problem situations encountered when trying to achieve a mutually beneficial, symmetric co-creative session for the human and the computational agent. We focus on four main problems: *universal mismatch*, *conceptual mismatch*, *artistic disagreement* and *generative impotence*. These problems bear similarities to the situations addressed by Wiggins (2006) as aberration and uninspiration, as discussed below.

We define and characterize the problems using the turn-taking structure of alternating co-creation. In particular, we consider different cases where the output of one participant is problematic for the other participant when the latter is supposed to use it as its input. Solutions to the problems are suggested from a position striving to better fill the needs of the human participant, a.k.a the user.

Universal Mismatch

In a *universal mismatch*, the human agent or the computational agent produces a concept that is outside the universe of the agent next in line:

$$c_h^i \notin \mathcal{U}_c \quad \text{or} \quad c_c^j \notin \mathcal{U}_h.$$

Such a situation could happen, for example, in poetry co-creation: If in the computational agent's universe, concepts are ordered lists of words, but the human suggests a visual poem which requires the understanding of the shape of a poem, the agent is fundamentally unable to understand the concept and operate on it.

Unfortunately, a *universal mismatch* is a fundamental problem since, by our definition, the computational agent cannot reach outside its universe \mathcal{U}_c .

However, if we allow the computational agent to skip its turn, we may wait until the human proposes another concept that fits within the universe of the agent. This allows for some level of *asymmetric* co-creation without proper alternation between the parties (see above). In the extreme case there is no overlap between the universes \mathcal{U}_h and \mathcal{U}_c (except for the empty concept c_\emptyset). In such a case, not even asymmetric co-creation is possible. With this we can formulate a fundamental requirement for alternating co-creation:

$$\mathcal{U}_c \cap \mathcal{U}_h \neq \{c_\emptyset\}.$$

Since there is no way to correct a *universal mismatch* during co-creation, such issues should be tackled already in the design of the co-creative agent.

Conceptual Mismatch

In a *conceptual mismatch* the computational agent is unable to recognize the concept given by the human as a valid concept or *vice versa*:

$$c_h^i \notin R_c \quad \text{or} \quad c_c^j \notin R_h.$$

Compared to a universal mismatch there is still a possibility to represent the relevant characteristics in the universe of

the computer. For example, if a computer strictly requires a specific poetic meter but the human has a different meter in mind, the computer can still process the poem (as a sequence of words) but it does not consider it as a proper poem.

The problem could again be solved trivially by having the human continue the creative process alone until we find a concept recognized as valid by the computational agent. If we want to achieve *symmetric* co-creation, we must consider transformational strategies instead.

This problem is somewhat similar to Wiggins' formalization of aberration where the single system comes up with a new concept outside its (current) conceptually valid space. Depending on the value of the concepts, Wiggins proposes some strategies for transformational creativity: If the system has found a new set of concepts which are all valued, we can change rules \mathcal{R} to include the new concepts in the conceptual space. If only some of the concepts found are valued, Wiggins suggests in addition to modify \mathcal{T} to avoid the unvalued concepts. If only unvalued concepts are found, he suggests to modify \mathcal{T} in order to avoid unwanted concepts.

In the case of alternating co-creativity, we can solve the *conceptual mismatch* problem by similar means, by transforming \mathcal{R}_c to include the new concept. Depending on the system, we may also need to make changes to \mathcal{T}_c or \mathcal{E}_c , to allow for the search to continue from the new concepts, or to expand the set of valued concepts to cover the new ones (see *Artistic disagreement* and *Generative impotence* below).

If the human participant is unable to understand the computational agent's suggestion, we have two possibilities to adapt to the humans needs: We can transform \mathcal{R}_c to exclude the "wrong" concepts, or modify \mathcal{T}_c to avoid them. This scenario however is difficult to successfully attain for two reasons: The human may be unable to communicate the problem to the computer in a sufficient manner, especially as we assume no other communication means except for the artifact. Also conforming too much to the human's desires may limit the creativity of the computational agent and decrease the overall value of the system for co-creation.

Artistic Disagreement

An *artistic disagreement* takes place when the human and the computational agent disagree on the (aesthetic) value of a concept produced by the other:

$$c_h^i \notin E_c \quad \text{or} \quad c_c^j \notin E_h$$

Artistic disagreement may seem like a trivial problem as the evaluation of the previously produced concept is not computationally necessary for continuing the search. However, from the perspective of co-creation, it is necessary to define this problem, as it may lead to a situation where the system continuously produces concepts that are of no value to the user, or the system is forced to search areas of no artistic interest to itself.

Conceptually, *artistic disagreement* is similar to Wiggins' concept of uninspiration. An uninspired system is unable to find highly evaluated concepts. In "hopeless" uninspiration, we have $E = \emptyset$, in "conceptual" uninspiration we have $E \cap R = \emptyset$ and in "generative" uninspiration we have $E \cap T^\infty(c_\emptyset) = \emptyset$.

Similarly, an *artistic disagreement* may stem from multiple underlying scenarios:

- The human and the computational agent do not value anything in their shared universe:

$$E_c \cap \mathcal{U}_h \cap \mathcal{U}_c = \emptyset \quad \text{or} \quad E_h \cap \mathcal{U}_h \cap \mathcal{U}_c = \emptyset$$

- The human and the computational agent do not value anything in their shared conceptual space:

$$E_c \cap R_h \cap R_c = \emptyset \quad \text{or} \quad E_h \cap R_h \cap R_c = \emptyset$$

- The human and the computational agent do not value anything the other one can produce:

$$E_c \cap T_h^n(c^i) = \emptyset \quad \text{or} \quad E_h \cap T_c^n(c^i) = \emptyset$$

Wiggins considers that "hopelessly uninspired" and "conceptually uninspired" systems are fundamentally ill-defined. Similarly, we consider that if the human and the computational agent are unable to value anything in each other's universes or conceptual spaces, and they are incapable of transformation, the human and the computational agent are fundamentally unsuited to work together in alternating co-creation. This implies that the computational agent is not designed to fit the user's needs.

In the case of systems capable of transformative creativity we have, however, some options for continuing the creative search in an alternating manner: If the computer does not value any objects in the shared universe we need to change \mathcal{E}_c to better fit the human valuation. If the computer does not value any objects in the shared conceptual space we can either change \mathcal{E}_c as previously, or change \mathcal{R}_c to increase the number of potentially valued concepts in the shared conceptual space. The case for handling specific concepts, unvalued by either the human or the computational agent is more nuanced.

If the computer is unable to value the concept provided by the human, the only option is to again change \mathcal{E}_c . However, if the computer produces concepts not valued by the user, we can either again accommodate the user's valuation by modifying \mathcal{E}_c , completely forbid search on uninteresting concepts by removing them from R_c by modifying \mathcal{R}_c , or direct the search towards more interesting concepts by modifying \mathcal{T}_c .

Since evaluation of the offered concept is not required in the formalization, *artistic disagreements* can also be solved by non-transformational means if we allow for the computer to simply trust the user's evaluations. In these situations the computer could simply continue the search despite the evaluative outcomes, but this could imply that the computer gives up, at least partially, its own \mathcal{E}_c and becomes more a servant to the user's goals. The new concepts found in this manner may then be either relevant or irrelevant to the human. Therefore, if the human is similarly trusting the computer, we may soon end up searching areas that are interesting to neither party.

Generative Impotence

Generative impotence occurs if the human or the computational agent is incapable of continuing the creative search from the concept provided by the other:

$$T_c^n(c_h^i) = \emptyset \quad \text{or} \quad T_h^n(c_c^j) = \emptyset.$$

Due to the differences in human and computational creativity, we are much more likely to end up in a situation where the computer is unable to process the current concept.

Trivially the case could be solved either by allowing the computational agent to perform a random search in \mathcal{U}_c , or returning to an earlier state, but these solutions seem unfit for a co-creative scenario. Simple random searches are not deemed very creative, and returning to an earlier state may in the worst case lead the human and the computational agent into an endless loop. Again, if we allow asymmetric co-creation, the computational agent can wait until the human produces a new c_h^i which it can process.

In order to enable the co-creation to continue in a symmetric manner, we will need to change \mathcal{T}_c so that the computer is able to continue its search for new concepts. Similarly, if the human is unable to continue creating from a concept provided by the computer, we can either continue the computational creation, or change the search strategy. However, in this case, it would be again extremely important for the human to be able to communicate to the computer in a relevant manner where the problem lies.

Computer Roles in Alternating and Task-Divided Co-Creativity

Formalizing co-creativity as alternating or task-divided search allows us to discuss the role of the human and the computational agent in co-creation from a computational viewpoint. We argue that *alternating* co-creativity poses more strict requirements to the computational agent than *task-divided* co-creativity. To be able to participate in alternating co-creativity, an agent has to be *complete*, whereas also *incomplete* agents can participate in task-divided co-creativity. This section discusses the roles of computational agents in alternating and task-divided co-creativity. We also give practical examples from literature to show how the formalization can be used to analyze existing systems.

Complete Creative Agents in Co-Creation

Computational creative agents, which are *complete* in the sense that they are capable of identifying (\mathcal{R}_c), generating (\mathcal{T}_c), and evaluating (\mathcal{E}_c) some concepts in a space (\mathcal{U}_c), can take more advanced roles compared to their *incomplete* counterparts. If they are capable of transformational creativity, i.e., of modifying their own behavior by changing (\mathcal{R}_c , \mathcal{T}_c , and \mathcal{E}_c) based on the human input, we can achieve *symmetric alternating* co-creativity at the system level. Complete agents incapable of transformational creativity can participate in *asymmetric alternating* co-creativity by skipping turns when needed. Naturally, complete agents are also capable of participating in *task-divided* co-creativity, if they suppress some of their capabilities.

Instances of *symmetric alternating* co-creativity are very rare in current literature. Many systems based on *complete* computationally creative agents have been transformed to interactive systems exhibiting creatively unbalanced scenarios: For example, in the Poetry Machine system (Kantosalto

et al. 2014) the computational agent works in an environment where it is restricted to provide partial concepts (poetic fragments) only when the human specifically asks for them. On the other hand in the pun generating STANDUP system (Waller et al. 2009), the computational agent seems to be performing the whole creative act alone, based on some minimal human input, such as a word to be included in the pun. These systems are good examples of originally *complete* creative agents participating in *task-divided* co-creation, where the creative responsibility is unevenly distributed to the human and the computational agent.

Among systems described in literature, the game level design system Tanagra (Smith, Whitehead, and Mateas 2010) seems to fit the definition of an *alternating* co-creation system best: In Tanagra, the computational agent and the user take turns in working on the same game level. In addition to generation, Tanagra also participates in evaluating the playability of both human and computationally produced content throughout the creative session.

Pleasing and Provoking Agents The nature of alternating co-creativity and the role of the *complete* creative agent are largely dependent on how it chooses to react to human input. In *symmetric alternating* co-creation, the interaction is defined by how much the computational agent decides to adapt to the user's needs. The agent can either try to *please* the human, by conforming to the human's ideas about concepts and their evaluation or *provoke* the human, by being more willing to challenge the human-provided concepts.

An extreme case of an agent striving to *please* would modify its creative process to better comply with the human's needs and preferences, even to the extent where it effectively reduces its own creativity by limiting \mathcal{R}_c or \mathcal{T}_c , or adjusting \mathcal{E}_c to avoid concepts that seem to be displeasing for the human. Current co-creative systems mainly employ pleasing agents. For example in Tanagra, the user's modifications are given priority over the computational agent's modifications so that the system can not change level components placed by the human. This effectively reduces the search space of Tanagra to accommodate the human.

Provoking computational agents can be thought of having stronger opinions, defending their viewpoints and resisting changes based on human preferences. This may make the agent outright challenging towards the human user's suggestions. Unfortunately, such systems are so far nonexistent, and in fact such a stance seems to be opposed by literature. For example, the creators of Tanagra talk about ensuring "that Tanagra does not push its own agenda on the designer" (Smith, Whitehead, and Mateas 2010).

Both *pleasing* and *provoking* agents have use-cases within co-creative systems. For example, if a user is attempting to produce concepts that convey his or her specific style, a *pleasing* agent which adapts to the user's preferences is more desirable. However, if a user is searching for more varied ideas, a *provoking* agent is a more ideal creative partner.

Naturally, agents do not have to be just *pleasing* or *provoking*, but a more balanced position between these two extreme stances is recommended. An agent balancing between

the two extremes would conform to the user's preferences whenever it would deem the transformation necessary and mutually beneficial. Therefore the agent should not outright accept or refuse transformational changes introduced by the human suggesting a new concept, but evaluate how valuable it would be to add new acceptable concepts, techniques, or value functions to its library. This manner of intentional, human-induced transformational creativity would potentially allow the computational agent to take more creative responsibility and be a better creative partner.

Incomplete Creative Agents in Co-Creation

Task-divided co-creation is unbalanced by nature, so it can take place between an *incomplete* as well as a *complete* creative agent and a human. So far, most examples of co-creative systems seem to be instances of task-divided co-creation, where the computational agent and the human clearly divide the creative responsibility over a concept to distinct subtasks, including generation and evaluation of concepts, and even the definition of the conceptual space.

The conceptual space where an agent operates is usually defined by the author of the program, but here we are more interested in how the human user participating in co-creation can effectively partake in defining the conceptual space in which the program does its generative and evaluative acts. In some systems, such as the pun generating STANDUP-system by Waller et al. (2009), the user can effectively set the conceptual space by controlling the level of word familiarity and joke class before the computationally driven generation of puns starts. In this case, the computational agent does not have a way to explore the search space beyond the user given constraints, nor does it have a chance to transform the conceptual space where it works. The user therefore acts effectively in the role of a "*concept definer*".

The strong generative capability of computational agents is often seen as the largest advantage of human-computer co-creation. For example Yannakakis et al. (2014) promote searching massive spaces as an advantage of computational systems in mixed-initiative co-creation. The role of "*concept generator*" is the de facto role of the computational agent in many systems, including especially many systems utilizing genetic algorithms. For example, the Evolver system (DiPaola et al. 2013) is essentially restricted to generating new populations of artwork candidates for the human user to evaluate and select for the next round of generation.

Where generation is often held as the forte of the computational agent, evaluation then again is very much held as the domain of the human author. Both Lubart (2005) and Maher (2012) assume that even systems of the most autonomous sort (computer colleagues or generative agents) will have a human evaluating their creative outputs. Human as the "*concept evaluator*" is clearly seen also in the previously mentioned Evolver project (DiPaola et al. 2013), where the human hand picks the candidates for each evolutionary round. Of course, some systems seem to share the evaluation responsibility, but on distinct topics: For example the Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014) will do evaluations of playability even for user generated content, but ultimately the human decides which con-

cepts are good. In fact, it could be argued that at least the final evaluation of when to end the search for better concepts is in current co-creative systems done by the human.

Discussion and Conclusions

From a computational perspective, the human and computer roles presented in earlier literature do not seem to be precise enough to categorize and describe the responsibilities of the human and the computational agent within the co-creative setting. First, roles of creativity support systems, including Lubart's (2005) computer as a nanny or pen-pal and Maher's (2012) support role, seem irrelevant from an analysis based on the creative systems framework (Wiggins 2006) since the tasks included in these roles (e.g. facilitation of communication between humans) do not count as creative behavior. Second, the computer as a coach (Lubart 2005) or enhancer (Maher 2012) rely in the computer re-formalizing the human's work by introducing specific creativity techniques, or giving the human fresh stimulus to induce creativity—both tasks again may be done by the computational agent without any creative behavior. Finally the final two categories, computer as a colleague (Lubart 2005) and generator (Maher 2012) actually fit a number of computationally very varied scenarios described in this paper. Therefore the introduction of new terms such as *symmetric alternating*, *asymmetric alternating*, and *task-divided* co-creation for describing the creative process, as well as the introduction of two types of computational agents *complete* and *incomplete* should be useful for the analysis of co-creative systems.

With regard to *alternating* co-creation we defined two modes for the computer to take: *pleasing* or *provoking* the human. It seems that whether a system should take the role of a more adaptive or a more challenging colleague depends on the needs and skill level of the user. This has a direct connection to Nakakoji's (2006) work, which underlines the role of co-creative systems as enabling faster creativity, training creativity, or entirely new areas for creativity for the user, and is also supported by the work of Liapis et al. (2013) on designer modeling. Indeed, choosing between a pleasing and a provoking stance will require further work on user modeling. In the future, systems taking a more *provoking* stance may be of particular interest for co-creativity research, as Maher (2012) points out that "successful examples of [human] collective creativity encourage diversity but do not require that everyone understand others' perspectives or even necessarily to reach consensus".

With regard to *task-divided* co-creation we were able to define three distinctive roles which can be taken either by the human or the computational agent: *concept definer*, *concept generator*, and *concept evaluator*. All of these roles can be clearly justified from the point of view of co-creativity as search, as all of them immediately relate to the capabilities in Wiggins' (2006) creative systems framework. The evaluator and generator roles are also implicitly defined in literature. However, in the formal categorizations by Maher and Lubart the systems again have little differences.

In our formalization, we have focused on the responsibilities of the human and the computational agent mostly within an iterative co-creative scenario. However, it is important to

note that human influence on the co-creative agent is not limited only to how the computational agent chooses to conform to human needs during a co-creative session, instead the design of co-creative systems is from early on influenced by user needs (Kantosalo et al. 2014), and they can be encoded in such fundamental aspects of the system that limit the universe of concepts the system can work on.

The co-creative session is also characterized by other factors besides the viewpoints and roles presented in this paper. One of the largest factors characterizing co-creation is interaction. We have omitted interaction entirely from this paper, but we want to note that some form of communication besides sharing the concepts could be valuable. Exchanging information such as descriptions of the creative process or evaluations of the concepts shared might provide significant improvements to the co-creative experiences between the human and the computational agent. Certainly, for the computational agent, such information would facilitate making educated decisions on how to carry out the creative transformations required to achieve *symmetric alternating* co-creation.

For possible communication between agents, we can learn from other frameworks of computationally creative agents, such as the FACE model (Colton, Pease, and Charnley 2011) or from how societies of computational agents work together e.g. in the creative workshop model suggested by Corneli et al. (2015). We could also learn from the perspective of social creativity by having the computational agent model the utility value of concepts to the human user, in order to direct the creative search into mutually more beneficial areas. In the future it would also be interesting to consider scenarios involving multiple computational agents and humans.

For now, the framework can be used to analyze current systems to pinpoint computationally interesting areas for research. Likewise, it can be used in the design of new co-creative systems, as it introduces new terminology for discussing both the goals of co-creation as well as the roles and stance taken by the system towards the human during the co-creative process.

Acknowledgments

This work has been supported by the European Commission under the FET grant 611733 (ConCreTe) and by the Academy of Finland under grant 276897 (CLiC). We would like to thank the anonymous reviewers for their constructive comments.

References

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Burleson, W. 2005. Developing creativity, motivation, and self-actualization with learning systems. *International Journal of Human-Computer Studies* 63(45):436–451.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*. April 27-29, 2011. Mexico City, Mexico, 90–95.

Corneli, J.; Jordanous, A.; Shepperd, R.; Llano, M. T.; Miztal, J.; Colton, S.; and Guckelsberger, C. 2015. Computational poetry workshop: Making sense of work in progress. In *Proceedings of the Sixth International Conference on Computational Creativity*. June 29-July 2, 2015, Park City, Utah, USA, 268–274.

DiPaola, S.; McCaig, G.; Carlson, K.; Salevati, S.; and Sorenson, N. 2013. Adaptation of an autonomous creative evolutionary system for real-world design application based on creative cognition. In *Proceedings of the Fourth International Conference on Computational Creativity*. June 12-14, 2013, Sydney, Australia, 40–47.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the Fifth International Conference on Computational Creativity*. June 10-13, 2014, Ljubljana, Slovenia, 1–8.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Designer modeling for personalized game content creation tools. In *AIIDE Workshop on Artificial Intelligence & Game Aesthetics*. AAAI.

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4):365–369.

Maher, M. L. 2012. Computational and collective creativity: whos being creative? In *Proceedings of the Third International Conference on Computational Creativity*. May 30 - June 1, 2012. Dublin, Ireland, 67–71.

McCormack, J. 2008. Facing the future: Evolutionary possibilities for human-machine creativity. In *The Art of Artificial Evolution*, Natural Computing Series. Springer Berlin Heidelberg. 417–451.

Nakakoji, K. 2006. Meanings of tools, support, and uses for creative design processes. In *International Design Research Symposium '06, Seoul, Korea*, 156–165.

Smith, G.; Whitehead, J.; and Mateas, M. 2010. Tanagra: A mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. June 19-21, 2010. Monterey, California, USA, FDG '10, 209–216. New York, NY, USA: ACM.

Waller, A.; Black, R.; O'Mara, D. A.; Pain, H.; Ritchie, G.; and Manurung, R. 2009. Evaluating the STANDUP pun generating software with children with cerebral palsy. *ACM Transactions on Accessible Computing* 1(3):16:1–16:27.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*. April 3-7, 2014. Ft. Lauderdale, Florida, USA.

Experience Driven Design of Creative Systems

Matthew Yee-King and Mark d’Inverno

Department of Computing, Goldsmiths, London
m.yee-king@gold.ac.uk, dinverno@gold.ac.uk

Abstract

The key contribution of this paper is to describe and demonstrate a novel application of grounded theory to the analysis of a human/machine music performance. Rather than attempting to measure the ‘creativity’ of our machine improviser, we instead proposed an investigation of the *experiences* of humans - in this case the designer, the performer and the listener. We report the design of an AI system chosen to perform in a specific *creative context* - a jazz-inflected musical performance in this case - and explore the specific experiences of these human actors through the performance itself. The performance is one which is a commonplace one where a single human musician interacts and performs with a single autonomous system. We describe this system which improvises by training pitch and event sequence models in real time from a live audio input and then uses a riffing behaviour to generate output in the form of note sequences with varying timbre. However, the main thrust of this paper is to propose a new methodology for understanding the role of the system through the interplay of experiences of audience, designer and performer throughout the performance, and describe how our time based media annotation system can be used to support that methodology. We present the results of this grounded ontology methodology applied to the text-based commentaries between system engineer, performer and listener. We argue that by developing an understanding of these inter-related experiences we can understand the desired and potential role of computational systems in creative contexts which can help in the design of new systems and help us curate new kinds of performance scenarios.

Introduction

The field of computational creativity has exploded into life in the last five to ten years with a whole range of work that reaches across theories, computational architectures and systems. It is an important field for a number of reasons not least because it throws up a number of issues around understanding the human creative process, understanding how we can support that process with new systems, and how any such understanding can help us in novel approaches to the design of these systems. Moreover, it is an important field because it allows for non-traditional, perhaps more playful AI approaches to be considered. When the AI world is increasingly populated by big data, and deep learning seems to be conquering all, it provides an important counterfoil to the mainstream.

However, there are clearly issues with the word “creativity” and the multitude of definitions which currently exist (Still and d’Inverno 2016). These can refer to the output (such as the work of Boden who categorised different forms of creativity based on the resulting value and novelty (Boden 2004)), can refer to the nature of the specific person who is disposed to producing creative acts (Guilford 1957)), and can refer to the nature of the process undertaken to produce specific kinds of outputs (Csikszentmihalyi 2009). Just like the concepts of “agent” and “agency” that predominated the 1990s when it was almost impossible to write a paper without giving one’s own definition of agency (Luck and d’Inverno 1995; d’Inverno and Luck 2003; Wooldridge and Jennings 1995), the positive side is that it allows for a whole array of innovative work. The negative is that - just as with agents - there is room for everyone and everything. In response, there is recognised need within the research community for clear methodological approaches that can evaluate autonomous computational systems which interact or collaborate in creative contexts with humans (Bown 2015).

In this paper, we respond to this need for appropriate methodologies by demonstrating how a grounded theory approach can be used to reflect upon human experiences around a new autonomous music improviser (AMI) called *SpeakeSystem*. The AMI was commissioned by the BBC’s ‘Jazz Line Up’ programme in 2015 for a one off live performance at the Wellcome Trust in London with British saxophonist Martin Speake and which was also broadcast live on national UK radio.

Since we are interested in examining the human perspective and response to autonomous music systems, we align our work with d’Inverno and McCormack’s promotion of ‘collaborative AI’ over ‘heroic AI’ (d’Inverno and McCormack 2015) and the interest of researchers such as Bown and Banerji in investigating the experience of musicians who play with these types of creative systems, and how they might be used as ethnomusicological probes (Bown 2015; Banerji 2012). This view is perhaps most in line with John Dewey who in his seminal work “Art as Experience” (Dewey 1934) looked to move the focus of thinking about art away from the object and towards the experience that takes place when we are making and experiencing art. Making and listening to music is a celebration of life, and it is through the experiences of making and listening where music - and all

art - has its meaning.

The contributions of this paper are as follows:

1. Description of a methodology that can be used to inform (interactive music) system design based on analysis of precise discourse around time based media from the perspectives of performer, listener and algorithm designer.
2. A grounded ontology of time tagged comments made from the perspective of the human instrumentalist, a listener and the system designer that can inform the design of future systems and concert curation.
3. Description of a system for enabling shared annotation of time based media that supports the methodology and grounded ontology approach.
4. Documentation, source code and analysis data for an autonomous music improviser which was commissioned by the BBC 'Jazz Line Up' programme and which performed live in a high profile concert in 2015 (Yee-King 2016).

Research questions

The work is framed with the following research questions:

1. How can we design a methodology based around collaborative annotation of video or audio recordings of performances which can effectively inform the design of autonomous music improvisers?
2. How does this methodology validate and expand upon previous research around autonomous music improvisers?
3. Which aspects of live human/machine improvisation performances are of particular interest to listeners, performers and algorithm designers?
4. How can understanding the interplay between the experiences of designer, performer and audience help in the design of future systems (and curated concerts)?

Structure

In the following section, we discuss related work before discussing the implementation of the system itself. In the section entitled Evaluation Method, we describe our methodology for evaluating human experiences with our system. In the section called 'Results', we present our ontology and further information about our categories. In 'Analysis', we reflect upon our results and compare them to those of other researchers. In 'Concluding Thoughts' we re-state our research questions and how we have addressed them.

Related work

First, we shall consider the evaluation of systems designed to be used in creative contexts with humans. Bown reflects upon the state of affairs in creative systems evaluation, noting that the lack of empirical grounding for evaluations might be preventing the kind of iterated improvement seen in other areas of AI research (Bown 2014). As a solution, he promotes user based analysis in real creative contexts. Eigenfeldt noted that "some attempts have been made at evaluation" but that many systems are "idiosyncratic ... specific to the artist's musical intention" (and thus presumably

difficult to compare to each other) (Eigenfeldt 2015). We address these issues - we describe and demonstrate a specific, transferable methodology which explicitly aims to develop knowledge that can inform future iterations of AMIs. Whilst we agree with Bown's appraisal, we acknowledge that other researchers have made significant attempts to specify evaluation methodologies. Collins proposed three areas in which AMIs can be evaluated: technically, aesthetically (audience reaction) and in the sense of interaction for the musicians (Collins and D'Escriván 2007) and Stowell et al. described a range of techniques that are suitable for evaluating live human-computer improvisation systems, including Turing Tests, audience surveys and task analyses (Stowell et al. 2009). Both schemes include aspects of human experience, but it is not the main focus. Hsu and Sosnick describe an HCI framework that directly considers human experience, where usability for the musician and musical interest for the audience of AMIs are evaluated using survey instruments (Hsu and Sosnick 2009). Subsequently to the work above, Bown provided a qualitative, thematic analysis (Clarke and Braun 2006) of musicians' experiences with his Zamyatin system (Bown 2015). Finally, Banerji reported an ethnographic approach to analysing how musicians changed their playing in response to an AMI, placing the system in a kind of socio-cultural map (Banerji 2012). We will contextualise our work by relating it directly to some of this previous work on the evaluation of AMIs

Human experience is also considered in non-music specific evaluation methodologies. It appears in one leg of Colton's "Creative Tripod", (skill, *appreciation* and imagination), but only the audience is considered, since Colton's work is focused on machine only creation (Colton 2008). Jordanous' Standardised Procedure for Evaluating Creative Systems (SPECS) provides a set of components of creativity, several of which relate to human-in-the-loop type interaction and experience, e.g. component 10, Social Interaction and Communication (Jordanous 2012). We shall revisit SPECS in our analysis later.

Considering the specific methodology used in this paper, we conduct a qualitative discourse analysis with a grounded theory method (Glaser and Strauss 1967). Grounded theory is chosen as it is suitable for the extraction of an ontology that can describe a discourse (Stern 2007). Our grounded theory approach consists of iterated data collection and categorisation followed by theory construction, in the form of an ontology (Birks and Mills 2011, p10). Whilst it is widely used, particularly in the social sciences, for qualitative analysis we note that grounded theory is not a panacea and that since its development in 1967, it has split into dialects and has been criticised for being overly dogmatic in its insistence upon emergent analysis as opposed to mapping analysis to existing theory (Goldkuhl and Cronholm 2010). Goldkuhl et al's Multi-Grounded Theory provides a solution to this, wherein pre-existing theory is mapped back onto the emergent theory (Goldkuhl and Cronholm 2010). We take this into account in our analysis, connecting our grounded ontology to Bown's thematic analysis and Jordanous' SPECS components.

To conduct the data collection phase of our analysis, we

made use of a collaborative media annotation system called MusicCircle which allowed the participants to discuss very specific parts of the performance in a solitary, then a collaborative phase (Brenton et al. 2014). In a sense, we wanted the annotators to become ethnographers, where they were thinking about what the musician and the system were doing, as they were doing it and the use of the annotation system allowed them to focus on specific aspects of the ‘exhibited behaviour’. For a reference point, consider the ethnographic approach described in (Barthet and Dixon 2011).

Now, we shall consider the second area of previous work: AMIs that are technically similar to SpeakeSystem. SpeakeSystem uses a hierarchical Markov model which is trained in real time from an audio stream. Pachet’s Continuator built Markov models from MIDI input in real time and used them to generate stylistically related, MIDI output (Pachet 2002). Yee-King has reported a series of systems that carry out timbral and symbolic sequence analysis and mimicry, including a ‘Matt Yee-King simulator’ that was based on Markov modelling of MIDI input (Yee-king 2011; Yee-King 2007; Nort 2014). Hsu’s timbral improvisation systems built non-Markovian, hierarchical models of timbral features from a live audio input (Hsu 2008). Collins’ FinnSystem used a pre-trained model of saxophonist and flautist Finn Peters combined with realtime audio analysis to control the output of the model and improvise (Nort 2014). Bown’s Zamyatin system used an evolved decision tree to move between target behaviours during live improvisation (Bown 2015). All of these systems were designed to operate in a human-machine creative context.

Implementation of the SpeakeSystem

The system was developed in the SuperCollider environment, and consists of essentially 3 modules: input, modelling and output. The input module shown in Figure 1 is responsible for generating a stream of labelled events and a stream of pitches from an audio signal obtained from a microphone. Event labels consist of *event type*, either note or silence and the *quantised length*. For example, note_500 would be a 500 ms long note. It was designed to work with monophonic instruments, but could be adapted to polyphonic instruments, given a sufficiently reliable polyphonic pitch tracker (or MIDI input).

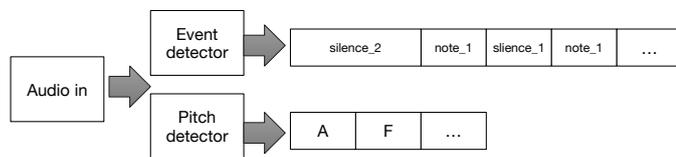


Figure 1: The input module analyses audio into a series of silences and note events and a series of detected pitches.

The modelling module consists of two multi-order Markov chains, one for pitches and one for events. As an example, the sequence of pitch labels a, b, b, d would result in several, different order entries to the pitch chain:

- $a \rightarrow b$

- $b \rightarrow b$ and $b \rightarrow d$ would be combined to make $b \rightarrow [d, b]$.
- $ab \rightarrow b$
- $bb \rightarrow d$
- $abb \rightarrow d$

The resulting chain is visualised in Figure 2.

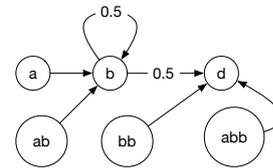


Figure 2: The pitch chain resulting from the input $abbd$.

Pitches and events could have been stored in a single chain but the output of the system was more varied when it was able to model pitch and event sequences separately, as it could generate similar rhythms with different notes to those played and vice versa. The output module ran the Markov chains in generative mode to make a sequence of events. The length and type of event was taken from the event chain and the pitch of note events was taken from the pitch chain. Considering the above input, and just the pitch chain, the initial note would be a, b, b or d , with 25% chance of a , 50% chance of b and 25% chance of d . If b was chosen, the generator state would be b , so there would then be 2 options: b or d , with equal chance. The system would always choose the highest order option that had at least two possible next steps; if only one option was available for bb , it would shorten its state description from bb to b and look up the options following state b , if only one option was available for b , it would pick from the distribution of all observed single notes. This combined accurate modelling with an interesting level of variation in the output.

The chosen pitch and duration would be used to generate MIDI note on and note off messages which were sent to an Access Virus C hardware synthesizer running in monophonic mode. The synthesizer was programmed with a sound which combined subtractive synthesis with some frequency modulation. There was no technical reason for choosing a hardware synthesizer over synthesis inside SuperCollider, but the Access Virus C is considered to have a very distinctive and powerful sound palette. The choice of a note based system as opposed to a more timbral system was made based on a discussion with the producer of the radio programme who commissioned the work, who pointed out that the performance was to be broadcast on the mainstream jazz show ‘Jazz Line Up’, as opposed to its more experimental counterpart ‘Jazz on 3’. The output module had some additional features which were designed to make it a more interesting improviser:

1. *Riffing with diminishing energy*. The system plays varying length sequences of notes wherein the modulation index of the FM synthesis was reduced in variably sized steps.

2. *Leaky models.* The system ‘forgets’ the training data leading to temporarily naive output. The aim was to provide a more structured feel to the piece.
3. *Separate timing and pitch models.* The system stored separate models of event and pitch sequences, so it could combine separate timing and pitch structures from the audio input. We aimed to provide more interesting and varied output over simply mimicing the performer.

The system was developed against a recording of a saxophone improvisation provided by Martin Speake prior to the performance. The various characteristics described above were hand tuned to maximise the musicality of the system when it was playing against the fixed recording.

Evaluation Method

In this section, the method by which the system was evaluated is described. In summary, a live performance was recorded and uploaded to a collaborative annotation system called MusicCircle. The performer, system designer and a listener annotated the recording, then a grounded theory approach was used to analyse the annotations they made. A DOI'd github repository providing a recording of the performance, the system source code and the annotation dataset can be found at (Yee-King 2016).

Performance

The piece was performed by Martin Speake, an experienced British jazz saxophonist, playing alto saxophone and the system, as specified in the previous section. A photograph of the ‘performers’ is shown in Figure 3, but we note that the computer operator was simply there to execute the autonomous system and to set the output level. Martin had not previously performed with the system or any other autonomous improviser, aside from a short technical test in the sound check on the night. He knew that he was performing with an autonomous system but he was given minimal insight into its design. The performance was recorded live at the Wellcome Trust on 26th September 2015 and simultaneously broadcast on BBC Radio 3.



Figure 3: Matthew Yee-King and Martin Speake at the live performance. The system was autonomous but Matthew had to adjust the volume level at the start of the performance.

The annotation system

The recording of the improvisation was then annotated using a system we have developed called MusicCircle. MusicCir-

cle was developed within a European research project and is available through www.museifi.com.

It has certain key features which were not available in other systems and which make it an appropriate tool for a range of applications including research and education. It can be classified as a scalable, web based, collaborative, time based media annotation tool and it has been used by several thousand students and researchers. For a more in depth discussion of MusicCircle and how it was developed, we refer to (Brenton et al. 2014) and (Yee-King et al. 2014). For the purposes of this work, MusicCircle allows its users to select regions of an audio or video file and to enter text comments which are then attached to the regions. Each user’s annotations are displayed along a ‘social timeline’, which shows each person’s commentary as a series of coloured blocks. Clicking on a coloured block reveals the comment and allows replies to be added. Each annotated region can then become a separate discussion thread. Figure 4 shows the user interface of the annotation system, where the recording of the improvisation has been annotated by 3 different people, as described in the next section.

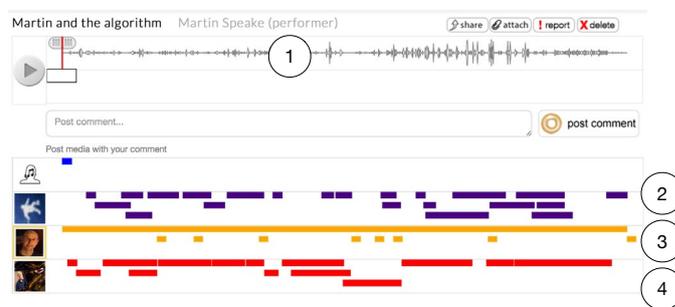


Figure 4: The annotation system, showing the time series of the recording (1) and 3 sets of annotations below, from the algorithm designer (2), performer (3) and the listener (4).

Annotation and tagging protocol

The concept of the annotation and tagging protocol was to obtain 3 independent perspectives on the improvisation in the form of time linked annotations, then to use an iterated grounded theory approach to create a set of tags categorising the annotations. The algorithm designer, the performer and a listener carried out the annotation. The listener did not attend the live concert and is a jazz music and autonomous agent expert, so is not a typical listener. In a future study, we would gather annotations from a wider range of listeners. The following protocol was followed:

1. A recording of the complete performance which lasted 3m 35s, was obtained and uploaded to the annotation system.
2. Each person was provided with a login for the system and their own copy of the recording for annotation.
3. They were asked to select regions of the recording that were interesting to them and to explain in the comment attached to the region why that region was interesting. They could not see each others’ annotations at this stage.

4. The annotations were combined onto a single timeline, as shown in Figure 4.
5. The annotators were asked to read each other's annotations and type replies if they wished. This marked the end of the annotation phase.
6. In the tagging phase, the algorithm designer read through the comments and replies, assigning tags to each.
7. The process of reading comments and adding tags was repeated until no new tags were needed and no comments needed to have any more of the existing tags added to them.

The production of the set of tags through the above protocol represented the initial and intermediate coding stage of grounded theory (Birks and Mills 2011, p9). Following this stage, tags were organised into a hierarchy of categories. This was achieved by considering each tag in turn and identifying whether that tag could be placed as a sub tag of any of the other tags. A constraint that each tag could only have one parent tag was imposed to simplify the process but it was found this did not induce excess 'stress' in the structure; each tag either stood alone or fit well beneath another. After this stage, we refer to the tags as categories, and the overall set of categories as a grounded ontology.

Results

Figure 4 shows all of the annotations as they appear in the user interface of the annotation system. There were 46 comments and 23 replies which were placed into 51 categories. An annotation could belong to several categories, and the number of categories assigned to an annotation varied between 1 and 8 with a rounded average of 4 categories per annotation. The number of annotations per category varied between 1 and 24, with a rounded average of 3 annotations per category. Tables 1, 2 and 3 show category frequencies for each of the three annotators. The frequency value is relative to the total number of categories assigned to annotations by that person, to make the numbers more comparable by compensating for the fact that different people left different numbers of annotations.

Frequency	Category
0.08	interaction
0.08	algorithm leading
0.07	autonomy
0.05	space
0.04	real
0.04	conversation
0.04	musician leading
0.04	structure
0.04	collaboration
0.04	roles

Table 1: Most popular categories for the listener

Figure 5 shows the grounded ontology that was derived from the process described in the previous section. Each category has a number next to it which is the number of annotations that were assigned to that category (this value does not include sub categories). The thickness of the border around the categories indicates this information visually,

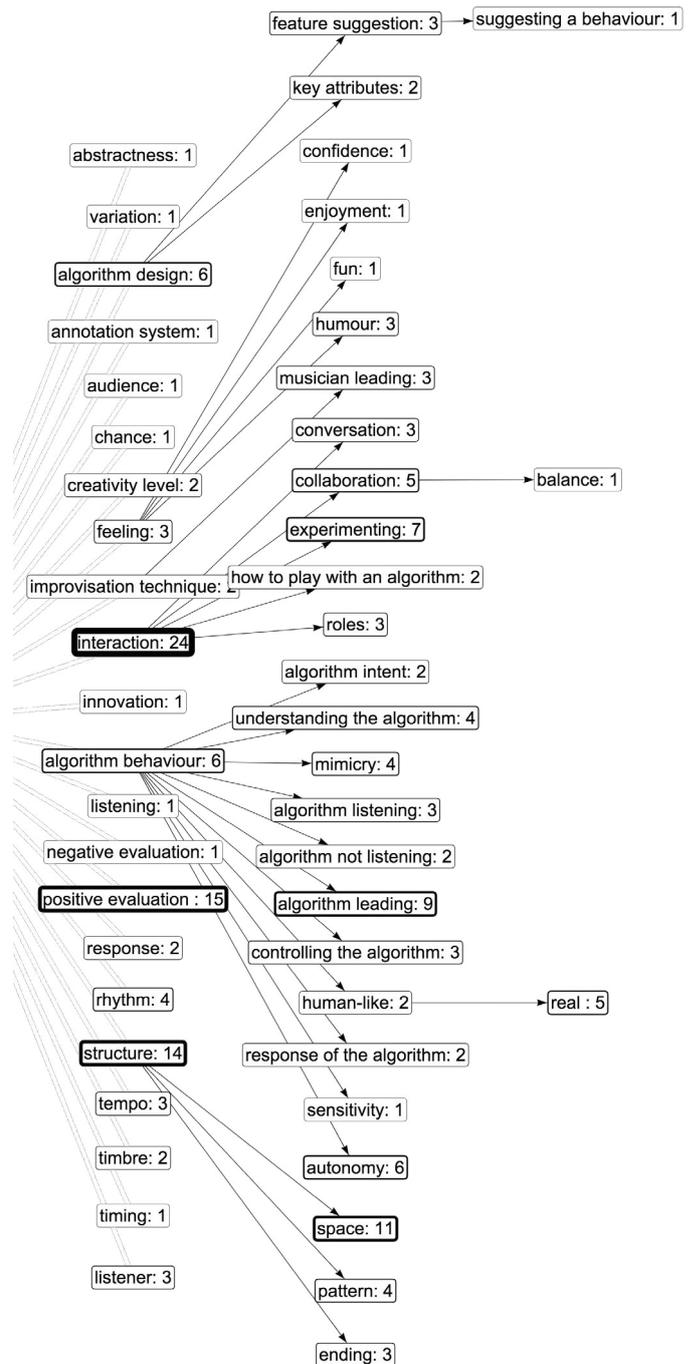


Figure 5: The ontology that was derived from analysis of the annotations left by the three participants. The number of annotations which were assigned to a category is indicated after the category name. The thickness of the border for a category visually indicates the number of annotations in that category. Counts are for that category only, not the category and its sub categories.

Frequency	Category
0.11	experimenting
0.09	structure
0.07	interaction
0.07	positive evaluation
0.07	space

Table 2: Most popular categories for the performer

Frequency	Category
0.15	interaction
0.10	positive evaluation
0.07	structure
0.05	algorithm design
0.04	pattern
0.04	space
0.04	algorithm behaviour

Table 3: Most popular categories for the algorithm designer

where thicker bordered categories had more annotations assigned to them.

Analysis

In this section, we will consider the results of the grounded analysis as they compare to other analyses. Jordanous empirically derived a set of 14 linguistic ‘components’ for the SPECS framework which were statistically more likely to be present in a corpus of research papers about creativity than in a corpus of research papers not about creativity (Jordanous 2012). Whilst the components were developed for the purposes of measuring the creativity of computational systems, we find them useful in framing our analysis.

Taking the category *Algorithm Leading* from the ontology which was assigned to 9 annotations, we can connect it to Active Involvement (SPECS component 1), Independence and Freedom (component 6), Social Interaction and Communication (component 10) and Value (component 13). So the system exhibited creative behaviour, but this is not the main focus of our work; perhaps we would prefer to consider if the human exhibited creative behaviour as a direct result of the actions of the system. The *Experimenting* category was assigned to 7 annotations, and looking at the commentary from the musician, they carried out three phases of deliberate experimentation to understand the behaviour of the algorithm. Experimenting links to Thinking and Evaluation (component 12), Variety, Divergence and Experimentation (component 14), Dealing with Uncertainty (component 2) and Social Interaction and Communication (component 10). The system seemed to encourage creative behaviour on the part of the human musician in a range of areas, and we can qualify this with reference to SPECS.

Next, we shall consider Bown’s thematic analysis which derived four key themes from a focus group discussion with musicians who had played with an AMI (Bown 2015). We contrast our approach with Bown’s approach in two key areas: 1) the method used to gather the data and 2) the method used to analyse the data. Our data gathering method was different in that our annotation system forced comments to be connected to a very specific region of a recording, so each comment came with explicit, musical evidence. Our data analysis method was different as it involved a categorical

rather than a thematic analysis.

Bown’s (paraphrased) themes were 1) interacting with the system gave a stronger sense of the nature of the interaction than watching someone else interact 2) there was an interest in the tangibility of the rules the system was using 3) participants did not refer to the system as a virtual musician, rather as an instrument or composition 4) they felt that long term structure was lacking.

Does our analysis support Bown’s themes? The contrast between interacting with the system and listening to someone else doing it did not appear in our ontology; perhaps a comparison of the categories connected to the listener’s annotations and those for the musician would shed some light here but we should note that Bown explicitly had the musicians listen to each other performing, but we did not. Understanding and discussing the rules used by the system was very evident in our data, as represented in particular by the *Algorithm behaviour* category and its sub-categories, which were used 41 times in total. The grounded analysis did not pick up on different ways of referring to the system but with hindsight, each annotator did have a different way of referring to it - the listener, who was an autonomous agent expert, decided early on in their annotations how they would refer to it:

[Listener]... amazed that the CMA (short for computational music agent or algorithm - someone else can decide)

The Musician referred to it as ‘the computer’:

[Musician]... to see how the computer would respond

The algorithm designer used ‘the algorithm’ or ‘it’:

[Designer] The algorithm picks up well on the rhythm here ...

Regarding long term structure, we tagged 14 annotations (over four times the average per category) with the ‘structure’ category, suggesting this was a strong theme in our dataset, given the assumption that structure refers to the compositional structure.

In summary, we found evidence for three of the themes identified by Bown, though we had to retrospectively look at the annotations for the ‘referring to the system’ theme, and this theme was a necessity in a sense as the commenters had to refer to the system somehow. Despite this, the fact that these themes emerged from two quite different data gathering and analysis approaches, with different people and different systems supports Bown’s findings and supports the validity of our findings.

Examining system design decisions

Our annotation methodology enables a very precise connection between the commentary, its derived ontology and specific sections in the recording of the performance. This allows us to consider the impact or otherwise of system design decisions upon the performance - when the system exhibits behaviour as a result of certain features, is this noticed by the annotators? As mentioned in the system description earlier there were three distinct features which aimed to produce more interesting output: *Riffing with diminishing energy*, *Leaky models* and *Separate timing and pitch models*.

We can map these features directly to comments such as the performer responding to the result of leaky models:

I slowed down my activity to the one long held note with the computer repeating it as separate notes until it seemed to give up realising I had finished playing! The audience felt like they were really with me/us in the moment too with their laughter at the end.

We can then look in the ontology for the categories that are associated with this comment: ‘humour’, ‘structure’ and ‘space’. Another example of the leaky models being noted (again, from the performer):

Yes i did wonder as sometimes it seemed to have logic in how it responded and then at other times it didn't make sense to me.

Here is an example of the listener responding to a section of the performance where the separate model feature was prominent:

Here we hear there very high notes which the CAM seems to “hear” and then responds to them in different ways each time.

In this way, we can consider key system design decisions and look for evidence that they had an impact upon the human experience, without the humans needing to understand how the system worked. This is similar to the unlocking of tacit knowledge made possible by user centred design.

A final potential application of this technique is that it might be used to inform the curation of concerts involving human/machine improvisation (McCormack and d’Inverno 2016). We can take the key items in the ontology and turn them into a set of challenges for algorithm designers - ‘create a humorous algorithm which experiments with algorithm leading and human leading’, ‘create an algorithm which uses space to encourage experimentation on the part of the human’, and so on.

Concluding thoughts ...

In this paper we have described and evaluated a new autonomous music improviser using a novel methodology. Here are the research questions stated at the start of the paper, with brief summaries of how we have addressed them. We start with the first two together.

1. *How can we design a methodology based around collaborative annotation of video or audio recordings of performances which can effectively inform the design of autonomous music improvisers?*
2. *How does this methodology validate and expand upon previous research around autonomous music improvisers?*

Response: Our methodology uses a social, time based media annotation system to enable focused annotation then discussion of human/machine performances. We have shown how this data can then be further analysed through grounded theory to yield an ontology that describes the resulting discourse. We have shown how the output of this method can be compared with that from other methodologies and that we are able to contrast and compare these results.

3. *Which aspects of live human/machine improvisation performances are of particular interest to listeners, performers and algorithm designers?*

Response: We derived and presented a grounded ontology describing the themes observed in a set of annotations left on a specific human/machine performance by a listener, a performer and an algorithm designer. We found that key themes included interaction, structure, space and algorithm behaviour. We were also able to verify our themes by mapping them to those described by previous, related research.

4. *How can understanding the interplay between the experiences of designer, performer and audience help in the design of future systems and concerts?*

Response: We have described how the kind of highly specific annotations and analysis enabled by our methodology can provide evidence for the impact of system design decisions upon the experience of listeners and performers. It is interesting to note that this can also inform the curation of concerts of such systems, where we can perhaps use our ontology to provide a list of challenges for system designers. In this way we can communicate interesting research themes in the field to the wide range of participants and investigate the themes through practice based activity.

We believe that key to designing systems that enable human/machine improvisation is starting from the perspective of the unfolding human experience, not just in music but in all forms of human creative activity.

.. and an Epilogue

A quote from the very beginning of Dewey’s seminal book *Art and Experience* feels appropriate here. Back in the 1930s Dewey argued that it is experience that is key to understanding the nature of art and creative endeavour:

In common conception, the work of art is often identified with the building, book, painting or statue in its existence apart from human experience. When an art product attains classic status, it somehow becomes isolated from the human conditions under which it was brought into being and from the human consequences it engenders in actual life-experience. When artistic objects are separated from both conditions of origin and operation in experience, a wall is built around them that renders almost opaque their general significance ... The task is to restore continuity between the defined and the everyday events, doings and sufferings that are universally recognised to constitute experience.

Our view is that it is of little practical interest to consider the amount, or system, of “creativity” contained within a computational system, but much more compelling to design systems that provide new kinds of creative experiences and opportunities for us all.

Acknowledgements.

We would like to thank the British saxophonist Martin Speake for generously taking part in this experiment and to Arthur Still for introducing us to the work of John Dewey.

References

- Banerji, R. 2012. Maxine's Turing Test A Player-Program as Co-Ethnographer of Socio-Aesthetic Interaction in Improvised Music. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment of the Artificial Intelligence and Interactive Systems Conference*, 2–7.
- Barthet, M., and Dixon, S. 2011. Ethnographic observations of musicologists at the British Library: Implications for music information retrieval. In *Proceedings of the 12th International Society for Music Information Retrieval conference (ISMIR 2011)*, number Ismir, 353–358.
- Birks, M., and Mills, J. 2011. Essentials of grounded theory. *Grounded theory: a practical guide* 1–14.
- Boden, M. 2004. *The Creative Mind: Myths and Mechanisms*. London: Routledge.
- Bown, O. 2014. Empirically Grounding the Evaluation of Creative Systems: Incorporating Interaction Design. In *Fifth International Conference on Computational Creativity*.
- Bown, O. 2015. Player Responses to a Live Algorithm: Conceptualising computational creativity without recourse to human comparisons? In *Proceedings of the Sixth International Conference on Computational Creativity*, 126–133.
- Brenton, H.; Yee-King, M.; Grimalt-Reynes, A.; Gillies, M.; Kirvenski, M.; and d'Inverno, M. 2014. A Social Timeline for Exchanging Feedback about Musical Performances. In *British HCI Conference*, 1–6.
- Clarke, V., and Braun, V. 2006. Using thematic analysis in psychology. , 3(2):77101, Jan. 2006. [5]. *Qualitative Research in Psychology*, 3:77–101.
- Collins, N., and D'Escriván, J. 2007. *The Cambridge companion to electronic music*. Cambridge University Press.
- Colton, S. 2008. Creativity Versus the Perception of Creativity in Computational Systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*, 14–20.
- Csikszentmihalyi, M. 2009. *Flow*. HarperCollins.
- Dewey, J. 1934. *Art as Experience*. New York: Perigree Books.
- d'Inverno, M., and Luck, M. 2003. *Understanding Agent Systems*. Springer.
- d'Inverno, M., and McCormack, J. 2015. Heroic versus collaborative AI for the arts. In Yang, Q., and Wooldridge, M., eds., *IJCAI International Joint Conference on Artificial Intelligence*, 2438–2444. AAAI Press.
- Eigenfeldt, A. 2015. Generative Music for Live Musicians: An Unnatural Selection Real-time Notation. In *Proceedings of the Sixth International Conference on Computational Creativity*, 142–149.
- Glaser, B. G., and Strauss, A. L. 1967. The discovery of grounded theory. *International Journal of Qualitative Methods* 5:1–10.
- Goldkuhl, G., and Cronholm, S. 2010. Adding Theoretical Grounding to Grounded Theory: Toward Multi-Grounded Theory. *IJOM: International Journal of Qualitative Methods* 9(2):187–206.
- Guilford, J. P. 1957. Creative abilities in the arts. *Psychological Review* 64(2):110–118.
- Hsu, W., and Sosnick, M. 2009. Evaluating Interactive Music Systems An HCI Approach. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 25–28.
- Hsu, W. 2008. Timbre-aware improvisation systems. In *Proceedings of ICMC2008, International Computer Music Conference, Belfast, UK*.
- Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation* 4(3):246–279.
- Luck, M., and d'Inverno, M. 1995. A formal framework for agency and autonomy. In *Proceedings of the First International Conference on Multi-Agent Systems*, 254–260. AAAI Press/MIT Press.
- McCormack, J., and d'Inverno, M. 2016. Designing improvisational interfaces. In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016)*. Universite Pierre et Marie Curie.
- Nort, D. V. 2014. Sound and Video Anthology: Program Notes. *Computer Music Journal* 38(4):119–127.
- Pachet, F. 2002. The continuator: Musical interaction with style. In *Proceedings of the International Computer Music Conference, ICMA, Gotheborg*, 333–341.
- Stern, P. 2007. Grounded theory methodology: Its uses and processes. *Journal of Nursing Scholarship* 12(1).
- Still, A., and d'Inverno, M. 2016. A history of creativity for future AI research. In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016)*. Universite Pierre et Marie Curie.
- Stowell, D.; Robertson, a.; Bryan-Kinns, N.; and Plumbley, M. D. 2009. Evaluation of live human-computer music-making: Quantitative and qualitative approaches. *International Journal of Human Computer Studies* 67(11):960–975.
- Wooldridge, M., and Jennings, N. 1995. Intelligent agents: Theory and practice. *Knowledge engineering review* 10(2):115–152.
- Yee-King, M.; Krivenski, M.; Brenton, H.; and d'Inverno, M. 2014. Designing educational social machines for effective feedback. In *8th International Conference on e-learning*. Lisbon: IADIS.
- Yee-King, M. J. 2007. An Automated Music Improviser Using a Genetic Algorithm Driven Synthesis Engine. volume 4448 of *Lecture Notes in Computer Science*, 567–576. Springer.
- Yee-king, M. J. 2011. An Autonomous Timbre Matching Improviser. *Proceedings of ICMC2011, International Computer Music Conference, Huddersfield*.
- Yee-King, M. 2016. speakesystem: ICCO ontology release <http://zenodo.org/record/46232>.

Applying Core Interaction Design Principles to Computational Creativity

Liam Bray

Art and Design

The University of New South Wales
Cnr Oxford St and Greens Rd
Paddington NSW 2021 Australia
liam.bray@sydney.edu.au

Oliver Bown

Art and Design

The University of New South Wales
Cnr Oxford St and Greens Rd
Paddington NSW 2021 Australia
o.bown@unsw.edu.au

Abstract

If we understand computational creativity (CC) as ultimately leading to useful interactive systems, then interaction design (ID) is a relevant body of theory with which to develop and test systems. Yet by engaging with complex and opaque systems, CC appears to break core ID wisdom, which preferences the comprehensibility of the system to users. We discuss core ID principles and ask how we can bring together ID and CC towards a better understanding of interaction in CC, whether in ‘merely’ generative art, human-computer co-creativity or full blown automated creativity. We look at ID issues surrounding creative processes of playful and non-objective search and consider how a more developed form of ID theory could work in these contexts.

Introduction

Recent work in computational creativity (CC) has begun to look at applying interaction design (ID) principles to CC systems, with the intention of advancing the usability and experience of these systems. But to date, no detailed discussion of the application of ID principles to CC has been had.

ID theory is a rich and diverse body of knowledge which extends the ability of designers to address the behavioural and experiential, whilst potentially being inclusive of computationally complex systems Gero (1990). If CC is to truly embrace the interaction between people and CC systems, then it follows that core interaction design issues should be explored.

CC designers intend the goals of their systems to be clear, but a means by which to determine the users’ perceived success in achieving these goals is not, as the evaluation of creative success is not empirically grounded in an objective methodology. In a previous paper we (Bown, 2014) speak to this:

“Terms such as ‘creativity’ and ‘imagination’ do not describe things that we can readily measure or objectively identify, they are concepts that frame other kinds of measurable and objectively identifiable things, as part of a loose theoretical framework.”

This is echoed by Carroll (2013) “It is critical to look beyond traditional time, error, and other productivity measure-

ments that are commonly used in HCI because these measures do not capture all the relevant dimensions of creativity support” and again by Shneiderman (2007): “The complex nature of human discovery and innovation cannot be studied like pendulums or solid-state materials”.

By contrast, if we consider CC from an ID point of view we are able to engage with the challenge of evaluation of creative systems in a meaningful way. Both enabling systems to be more effectively designed for use by creative practitioners, and genuinely resolving dilemmas of empirical grounding (Bown, 2014).

In this paper we take a more detailed look at key principles from ID, and how they might apply to CC systems, in order to develop a more holistic means of evaluating and designing CC systems from a user’s perspective. We also suggest a simple framework that describes potential visibility concerns in CC systems by defining the behaviour of a system in terms its structure and its trajectory.

Our comments apply most readily to more traditional creative tool use cases, and in this sense are focused on supporting creative users (Candy and Edmonds, 1997). These same comments might not carry so well into the relationship between audiences and creative art machines, but we nevertheless pursue the possible value of this ID approach in such cases. We take the view that there is always an interface of some description, that warrants a discussion about the design of that interface. At the same time, we realise that different interaction scenarios will have very different conditions, and although we aim for general principles, we do not expect to be able to find too narrow a set of principles that is applicable such a wide set of cases.

Application of ID to CC

Norman’s conceptual model approach (Norman, 1988) popularised several key principles for the design of ‘everyday things’. His principles were rapidly applied to interactive technologies.

One of Norman’s most influential usability principles is *perceived affordance*. This describes a person’s conception of the various things you can do with a given object. This encompasses the heuristic experience of working with a system and broadly outlines the ability of the user to perceive and recognise a system’s interface. A common example of this is a door handle; a door handle affords pulling, as its physical

properties constrain what can be done with it in relation to its environment (Rogers, Preece, and Sharp, 2007). This is the same for a mouse button, which has a physical relationship to the digital interface it controls. The digital interface itself also offers perceived affordances as it too can be described as having constraints, and intuitive heuristic methodologies can be applied to it. For example a user can use past experience or common sense about what a digital button might do when they click on it.

Norman's principle of *visibility* is the simple idea that the more visible the operations of the system are, the more likely users will be able to know what to do next (Rogers, Preece, and Sharp, 2007). The complexity of CC systems often requires that functions are simplified or hidden from the user. This can lead to a conceptual black box. A user sees an input and receives an output, but the extent and nature of what happened in-between can be hard to understand.

The lack of visibility of the process can also scale with complexity. A system which appears simple at first can, in a CC process, become complex and unmanageable for a user to effectively make decisions. For example, a user may be able to manage a simple 2D physical model such as balls bouncing around in a 2D environment, which are easy to recognise and mentally model. But if the environment contains any more than a few interacting agents the ability of the user to make meaningful and effective decisions decreases. This has a downward-spiral effect for users; as interactions become more complex their ability to maintain and develop a clear conceptual model decreases along with the systems visibility.

Mapping, the direct relationship between controls and their effect on a system, is closely related to visibility, contributing further to the intelligibility of the system. A user's ability to interpret the affordances of an interface element depends in part on the arrangement of interface elements as they are presented to the user. In CC systems that are designed to enable users to manage computationally complex scenarios, it becomes paramount to the intelligibility of the system that a coherent mapping remains visible and intuitive.

At this point, conventional wisdom might say that if the complexity and opacity of CC systems are completely at odds with these very foundational principles of ID, then perhaps ID principles are simply not relevant to CC.

We contend that instead ID and CC should evolve together to develop a rich model of ID that is specifically suited to CC (as well as a wealth of other situations involving rich interaction with AI systems that are likely in the near future). Part of the argument for this is that it is hard to think of CC systems in the absence of some form of interaction. Instead, despite the isolated lab-based nature of much CC research, the majority of CC researchers do take care to emphasise the essential embeddedness of art in a complex of human social behaviour, and ultimately aspire to create work that interconnects with this complex, whether in the form of simulated artist agents, creativity support tools, Twitter bots, multi-agent simulations or other types of interactive systems. Work in CC that displays ignorance of this inherent network complexity has not generally been widely accepted.

In short, all CC research requires the developers to 'design' (at least establish and observe) interactions at some point along the way.

Lubart states that computers can facilitate (a) the management of creative work, (b) communication between individuals collaborating on creative projects, (c) the use of creativity enhancement techniques, and (d) the creative act, through integrated human-computer cooperation during idea production (Lubart, 2005). If we consider different CC goals according to Lubart's classification of the different ways in which computers can act as creative partners, then ID clearly plays a role in each of these forms of interaction, almost by definition. The clear application of the ID principles discussed so far becomes harder as we work our way through this list. In extension of Lubart's list, we could add (e) the complete artistic autonomy of the system, interacting with others only as an artist interacts with her audience. One contention of this paper is that even the latter should be subjected to ID thinking, and that ID principles need to be modified to extend that far.

One practical systematic approach to this conundrum is to distinguish between areas where (or levels at which) transparency is needed and where opacity can be allowed, extracting the former into what both programmers and designers understand as an 'interface'. This is only to reiterate conventional ID thinking in a way that might be more palatable for the above concerns in CC.

Dennett's (1989) *intentional stance* offers one well-known strategy for interaction with a certain group of complex systems – other humans and animals. We 'model' (i.e., intuitively understand) these systems not in terms of their physics or mechanical design, but in terms of their thoughts, intentions and goals. This reduces the complexity involved in predicting the system's behaviour, and we do this innately because our brains have evolved to do so. It would be useless trying to use a 'physics stance' to model what an adversary was going to do next, even though it will help predict the swing of their arm in a fist fight. In particular we have specifically evolved to 'model' the minds of other humans, the cognitive product of a competitive coevolutionary race (Whiten and Byrne, 1997; Dunbar, 2004; Boyd and Richerson, 1985) that some think is key to our sense of consciousness. Making systems that behave exactly like humans might be a good strategy in CC, but it is interesting to note that this would not make them necessarily easy to model.

Gaver argues that as computing has become progressively more ubiquitous, it has brought with it the values of the workplace. Concerns for clarity, efficiency, productivity and a preoccupation with finding solutions to problems have been imposed on digital devices as if they are limited purely to mirroring the work required to achieve an ordinary life, such as the completion of everyday chores (Gaver, 2002). He suggests that the idea of *homo ludens*, a term taken from Huizinga, humans defined first as playful creatures, brings our curiosity, our love of diversion, our explorations, inventions and wonder to the fore of designing interactive technologies. Gaver is intentional in his definition of play, and diverges from Huizinga's definition, preferring

Kaprow's (Gaver, 2002) definition of play as distinct from games. Kaprow acknowledges that while games and play:

“both involve free fantasy and apparent spontaneity, both may have clear structures, both may (but needn't) require special skills that enhance the playing. Play, however, offers satisfaction, not in some stated practical outcome, some immediate accomplishment, but rather in continuous participation as its own end. Taking sides, victory, and defeat, all irrelevant in play, are the chief requisites of game. In play one is carefree; in a game one is anxious about winning(Gaver, 2002).”

Gaver's application of homo ludens comes to bear on CC in that, if we are to leave work behind and design systems that embrace human creativity, then we need to intentionally seek play as a form of engagement. “This is an engagement that has no fixed path or end, but instead involves a wide-ranging conversation with the circumstances and situations that give it rise.” (Gaver, 2002), it is important that open-ended and self motivated forms of interaction are employed. This enables users to find new perspectives and new ways to create, “through ambitions, relationships, and ideals” (Gaver, 2002).

Gordon Pask, an early proponent and practitioner of cybernetics sought to build machines that coexisted in a mutually constructive relationships with users (Negroponte, 1975). Pask defined this process as conversation theory. He was specifically interested in how human-machine interactions could be subject to context and interpretation as an additional way of locating meaning in the interaction with the machine. Recent practitioners of conversation theory include Haque (2007), who argues that creative use of computers needs to incorporate these mutually constructive relationships as a means of expanding creative potential.

Likewise, many CC researchers have attempted to engage with the open-ended nature of creative discovery (Saunders and Gero, 2002), building on creativity research (Csikszentmihalyi and Sternberg, 1988; Boden, 1990) to design systems that exhibit these properties. Biological evolution has been one source of inspiration here. Whilst building emergent complexity into closed computer systems has proven difficult (Bown and McCormack, 2010), several researchers have reported moderate success with interactive genetic algorithms (IGAs) as human-computer collaborative tools for open-ended search. Stanley and Lehman Stanley and Lehman (2015) have been notable advocates for the open-ended nature of creativity following the observation that a distributed IGA system, *Picbreeder*, built by their team, was used by participants in a way that clearly demonstrated an absence of preconceived goals. Users were observed selecting images to evolve and then allowing the image evolution process to lead them towards recognisable shapes. Highly recognisable images emerged, such as faces and cars, but not because users set out to draw faces and cars. In their book Stanley and Lehman develop such observations, as well as their research in novelty search as a form of optimisation, into a general theory that attempts to define objectives as more of an impediment than a help to true discovery. By definition, they argue, a hard creative problem does not in-

dicates the direction in which you should head to discover the solution, so setting out in the apparent direction of the objective is a flawed approach.

Stanley and Lehman are in a sense restating a well-known principle of creativity theorists, with added evidence from computer science. Perkins (1996), for example, examines successful creative individuals and identifies their most common strategy as being one of spreading their bets across a wide range of solutions. Others in design creativity have identified a form of reverse creative discovery where problems are found to suit existing solutions (the story of the Post-It Note is one of the best known examples).

In this discussion we have encountered a series of ideas around the intersection of ID and CC: that Norman's principles of visibility and a clear conceptual model have limits in the context of the complex and opaque nature of CC systems; that, according to Gaver (2002), opacity is acceptable in the context of playful interaction, and that according to Stanley, Lehman and many others, open-ended search, which we closely associate with playful interaction, is critical to true creative search; and that for any system or use-case we should attempt to identify where transparency is needed (i.e., in the context of goal-directed functional behaviour) and where opacity can be accepted (i.e., in the context of open-ended search, with conditions attached), in the creation of CC interfaces.

Visualising Structure and Trajectory

The above formulation still does not give much insight into specific methods for breaking down CC-ID problems to find suitable balances between opaque and transparent aspects of interaction. Our claim is only that it reframes a common problem from ID in a way that is palatable for CC. In our previous work studying popular end-user generative music composition tools (Bray and Bown, 2014), we have sometimes found it useful to think about how users attempt to understand the behaviour of the system in terms of a breakdown between its structure and its trajectory. Most systems can be easily decomposed into these two parts: a structure that is generally assumed to be fixed, but may be mutable to some minor extent, and a set of ongoing movements or state changes around that structure. For example, dropping a pinball into a pinball machine, we think of the fixed structure of the pinball machine layout dictating the trajectory of the pinball. We think of all traditional acoustic instruments as having specific fixed structures around which a musician defines a trajectory. The instruments can be (imperfectly) parameterised, as is often seen performed in the creation of virtual instruments.

Our suggestion is that Norman's visibility or clear system model occurs wherever the user can clearly perceive the system's structure, and get a handle on how this structure dictates the trajectory.

By contrast, more complex generative systems can get harder to model because, we suggest, it is harder to see the structure and pull it apart from the complex movement of the system. McCormack and McIlwain's *Nodal* (McCormack et al., 2007) is an example we explicitly looked at in this way

(Bray and Bown, 2014). Whilst there is always a clear structure in Nodal, it is hard to tell how it will influence the system's unfolding trajectory just by looking at it. In the case of Nodal, the user is expected to build the networks by hand, so this kind of opacity could be seen as an impediment. But it may not be: another view is that the user develops strategies for progressing their work, and heuristics for thinking about what is going on, even though they struggle to develop a clear model of the system behaviour. This type of user behaviour would seem to make a clear break into the domain discussed above as more playful open-ended search.

Another possibility, alluded to here, is that users, drawing on their general intelligence, are able over time to better model the system, becoming experts. This expertise might be equivalent in ways to the species-specific expertise we have discussed in the case of Dennett's intentional stance. Whether or not these derived models have any common abstract properties would be of great interest.

Specific CC scenarios

We now briefly work through how these ideas might be applied to specific CC scenarios, and look at the different ways in which we might apply ID concepts to these different areas.

The first case we consider is already introduced above: generative tools such as *Nodal* that employ different generative paradigms with diverse approaches to user interaction. In this case, the distinctions between opaque and transparent approaches are applied straightforwardly, as described above. If a system can be transparent, then it could be potentially used in a more goal-directed manner. Opaque systems can often only be used in a goal directed way if you are focusing on process-based creativity, otherwise they require a search-based approach. However, as we suggest, there may be strategies for making opaque systems less opaque, through their representation, learnability and so on.

Another case we have already discussed are IGAs, which, as we have seen, seem to lend themselves to open-ended search more readily than to goal directed search. This is perhaps due to their randomness – it would be frustrating to aim for goals because you'd be forever looking for the next link in the chain, much better to respond to the available options. Conventional GA theory does however require certain conditions of transparency, for example in that mutated objects should be similar to their parents; there must be smoothness and consistency which we can think of as something clearly modellable.

Corpus-based learning approaches are interesting from a transparency point of view because they rarely offer any intuitive way to understand what the system has learnt. Such systems also tend to be self-contained processes, transforming an 'inspiring set' into new candidate outputs. There are rarely coherent ways for users to get involved in this process except in the tweaking of parameters, although this is hugely important for successful results, and ID research has been conducted in this area. Martin, Jin, and Bown (2011) observe that this imperviousness to user input has been a key problem to making usable systems.

More recently, work such as that of Pachet and Roy (2014) involves the use of corpus-based systems that 'mash-up' musical styles, where there is plausibly more involvement of the user. Here we may approach something akin to an intentional stance approach, where we might ask for, say, a performance of a Beatles song in the style of Wagner. Here the user can clearly engage in tasks in goal-directed or open-ended ways.

User interfaces that allow users to specify target goals are now common across a range of application areas, and is becoming an increasingly active area in architecture, where we need to reach multi-objective targets of, for example, structural stability, temperature regulation and visual criteria all at once. Some interfaces consist of a simple bank of sliders, whilst others must be programmed. In other producer-critic models, we might interactively evolve the fitness function that is used to do targeted evolution of an outcome. In other cases we might train a neural network to learn a preference. Veale (2015) argues that these types of CC systems go beyond 'mere generation' and take on artistic responsibility for selection or evaluation. Users become meta-creators, creating with and through CC processes.

Looking at the bigger picture, Plotkin (2009) discusses how automated discovery methods transform computers from machines that we instruct to perform specific tasks, to *genies* that respond to specific requests for outcomes. Such systems may therefore have wide reaching implications for how we interact with computers on a daily basis.

Lastly, we have recently seen work in art-making systems that explain themselves to their audience in natural language (Colton and Ventura, 2014), as a form of interactive experience that, in Colton's terms, 'frames' the artwork with additional relevant information. This is an approach that very much places ID at the centre of the design of CC systems, both by holistically considering the user experience associated with evaluating art, and more specifically by breaking from the unidimensional approach to aesthetic evaluation just mentioned, considering instead a rich multimodal set of possible interactions and judgements. What has yet to be elaborated on in theoretical terms is how we might frame these interactions between a machine artist and its audience in terms of a set of goals. The makers of the system invariably have goals when they place the system in front of people, just as other software developers do, and indeed, individual artists do when they interact. If the goal is open-ended co-creative search then the ID issues will be framed by this, and if the goal is to produce entertaining artworks for the home, then the ID issues will be different.

Conclusion

In this paper we have presented a series of ideas that can be summarised as follows:

- Opacity is inherent to CC but generally problematic in ID, except in the context of playful interaction.
- Open-ended search, associated with much creativity, can be stimulated through playful interaction.
- An ID approach to CC would be to attempt to work out what can be made visible, and what cannot, and work

out how the opaque elements can still be usable given the above assumptions.

- It may be possible to make some opaque aspects of systems more visible by considering how we mentally model these systems. We pose a distinction between structure and trajectory as one way this might be handled.

As each of these areas within CC research matures and starts to be applied in real software, the ID issues become more relevant, apparent, and better understood. As this happens we have the opportunity to build ID techniques specific to advanced CC.

References

- Boden, M. 1990. *The Creative Mind*. George Weidenfeld and Nicholson Ltd.
- Bown, O., and McCormack, J. 2010. Taming nature: tapping the creative potential of ecosystem models in the arts. *Digital Creativity* 21(4):215–231.
- Bown, O. 2014. Empirically grounding the evaluation of creative systems: incorporating interaction design. In *Proceedings of the Fifth International Conference on Computational Creativity*.
- Boyd, R., and Richerson, P. J. 1985. *Culture and the Evolutionary Process*. Chicago, IL, US: University of Chicago Press.
- Bray, L., and Bown, O. 2014. Linear and non-linear composition systems: User experience in nodal and pro tools. In *Proceedings of the Australian Computer Music Association Conference*.
- Candy, L., and Edmonds, E. A. 1997. Supporting the creative user: a criteria-based approach to interaction design. *Design Studies* 18(2):185–194.
- Carroll, E. A. 2013. *Quantifying the personal creative experience: evaluation of digital creativity support tools using self-report and physiological responses*. Ph.D. Dissertation.
- Colton, S., and Ventura, D. 2014. You can't know my mind: A festival of computational creativity. In *Proceedings of ICCO 2014 (International Conference on Computational Creativity)*.
- Csikszentmihalyi, M., and Sternberg, R. 1988. The nature of creativity: Contemporary psychological perspectives. *Society, culture, and person: A systems view of creativity* 325–339.
- Dennett, D. C. 1989. *The intentional stance*. MIT press.
- Dunbar, R. 2004. *Grooming, Gossip and the Evolution of Language*. London: Faber and Faber.
- Gaver, B. 2002. Designing for Homo Ludens, Still. *Interaction Research Studio, Goldsmiths, University of London, 13 Magazine No. 12* 163–178.
- Gero, J. S. 1990. Design prototypes: a knowledge representation schema for design. *AI magazine* 11(4):26.
- Haque, U. 2007. The Architectural Relevance of Gordon Pask. *Architectural Design* 77(4):54–61.
- Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4):365–369.
- Martin, A.; Jin, C.; and Bown, O. 2011. A toolkit for designing interactive musical agents. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*. ACM.
- McCormack, J.; McIlwain, P.; Lane, A.; and Dorin, A. 2007. Generative composition with nodal. In Miranda, E., ed., *Workshop on Music and Artificial Life*.
- Negroponte, N. 1975. *Soft architecture machines*. MIT press Cambridge, MA.
- Norman, D. 1988. *The Design of Everyday Things*. New York: Basic Books.
- Pachet, F., and Roy, P. 2014. Non-conformant harmonization: The real book in the style of take 6. In *Proceedings of ICCO 2014 (International Conference on Computational Creativity)*.
- Perkins, D. N. 1996. Creativity: Beyond the darwinian paradigm. In Boden, M., ed., *Dimensions of Creativity*. MIT Press. chapter 5, 119–142.
- Plotkin, R. 2009. *The genie in the machine: how computer-automated inventing is revolutionizing law and business*. Stanford University Press.
- Rogers, Y.; Preece, J.; and Sharp, H. 2007. Interaction design.
- Saunders, R., and Gero, J. S. 2002. How to study artificial creativity. In *Proceedings of the 4th conference on Creativity & cognition*, 80–87. ACM.
- Shneiderman, B. 2007. Creativity Support Tools: Accelerating Discovery and Innovation. *Communications of the ACM* 50(12).
- Stanley, K. O., and Lehman, J. 2015. Why greatness cannot be planned. *Springer Science Business Media. doi* 10:978–3.
- Veale, T. 2015. Creativity ex machina. *Routledge Handbook of Language and Creativity* Rodney Jones:353–366.
- Whiten, A., and Byrne, R. W. 1997. *Machiavellian Intelligence II: Extensions and Evaluations*. Cambridge, UK: CUP.

Designing Improvisational Interfaces

Jon McCormack
Monash University
Caulfield East, Australia
jon.mccormack@monash.edu

Mark d’Inverno
Goldsmiths, University of London
London, UK
dinverno@gold.ac.uk

Abstract

This paper examines the possibilities for creative interaction with computers, in particular modes of interaction based on improvisation and spontaneous creative discovery. We consider research findings from studies in psychology that investigate how humans improvise together to see what could be useful in helping us to design systems that provide new kinds of interactive opportunities. We draw on our personal experiences both as computer scientists working across art and music, and as practicing artists and musicians, to examine what artists and musicians would want in any system designed to support creative interaction with a computer. We bring together these different findings to propose a series of working principles which form a basis for designing systems that facilitate collaboration and improvisation with computers in creative domains.

Introduction

We are interested in designing systems that provide individuals and groups with opportunities for new kinds of collaborative creative experiences with machines. There is the potential to design new experiences and interaction scenarios which can increase the scope and depth of an individual’s artistic practice and enhance creative development. In order to do this, we want to fully realise the potential concept of the machine as a *creative collaborator*. We differentiate this approach from the view of computers as “smart tools” or “productivity enhancers”, or approaches that seek to create machines as lone systems capable of autonomous, independent creative activity.

In order to design such systems, and to understand the range of potential scenarios, our principle design approach is one which centre stages the contemporary needs of artists in their own practice. In doing so, we believe it is possible to design new kinds of interactions and outcomes by imbuing the machine with the agency of a creative collaborator. Through this approach we aim to bring new experiences to a wide range of people; encouraging greater levels of creative activity and engagement in general. If we can build systems with the right sense of agency and autonomy, then we can provide new opportunities for learning through collaboration, new opportunities for performances involving human-machine interaction, and new opportunities for individual artistic discovery and expanded creative cultures.

To appreciate the machine as a collaborator, it has to be perceived as having a degree of *creative agency* (Bown and McCormack 2009), being an active contributor to the unfolding creative process rather than simply responding automatically as a tool. The degree of proactive, autonomous creative agency facilitates experiences closer to “collaboration” than with software tools having a fixed reactive functionality. A high-level of creative agency should enable us to interact with the computer in natural and intuitive ways, just as we might if collaborating with other human artists: jamming and improvising, listening and responding in artistically meaningful ways. In this sense our approach is a humanist one that sees technology’s role as nurturing, supporting and expanding human creative activity, rather than mimicking or replacing it (d’Inverno and McCormack 2015). We emphasise the *human experience* (Dewey 1934), rather than the system per se.

Undoubtedly, designing these systems is a challenging task. A key reason that makes the design challenging is that whilst artists are always exploring new ways of working, they like to explore things in their own way, and rarely like the idea of giving over agency in the creative process to a machine or being forced into constrained interactions that are dictated by the hardware or software design. It’s our experience that when a machine has its own agency, the artist using the machine for the first time (rather than the software engineer who built it) is typically frustrated rather than excited. Nonetheless, a number of existing projects illustrate the enormous potential embodied in this approach, provided we are guided by the needs of artists rather than the needs of the engineer in demonstrating a new system design.

As a starting point for this endeavour, we will focus on *improvisational interaction*. We want to build interactive computer systems that intelligently interact and perform with artists in real time; systems that adapt to – and learn about – a person’s style, dexterity and proficiency in general. Good improvisational interaction requires people and computers working together seamlessly in an on-going dialogue where, as this dialogue progresses, it grows in nuance and virtuosity, even as the human artist’s capacity and insight expand. In this style of interaction the emphasis is on play and experimentation rather than formal composition, but this doesn’t preclude the development or progression of substantive creative ideas and works. Successful improvisational exchange

between person and machine requires a free and natural interaction, unmediated by unnecessary layers of technology. Hence the problem is one of both successful physical interface design for a given context (how one interacts when improvising) and the substance of that interaction (the creative agency of the system you are interacting with). Any solution will necessarily involve a high degree of co-dependency between interaction, intention and agency.

As part of this paper we will examine a number of findings from the psychological literature on human creativity, where there has been a strong tradition of treating “creativity” as something distinct and tangible in the human mind that can be measured. Whilst we remain skeptical of this view (see (Still and d’Inverno 2016) for example), we do believe that there are studies in Psychology, when looking at “human creative activity”, that can help better understand the processes involved when humans improvise: with each other, with other actors, and with tools, all of which are important considerations in designing software systems that support creativity. The literature on improvisation is quite expansive and detailed, with many long-term studies revealing interesting features and theories about what happens when humans take part in an improvisational activity. We believe it is worth exploring these findings to better understand how we approach the design of human-computer interaction in a collaborative, improvisational context.

In addition to the psychological literature, we also draw on our own experience as researchers who are also practicing artists and musicians. To date, many of the most successful systems within the field of *computational creativity* have been designed by people who are practicing artists, bringing the full depth of their artistic knowledge to the system’s fundamental design, typically to further develop their artistic practice (e.g. (McCorduck 1990; Cope 1991) are two classic examples). In this paper we attempt to articulate some of this artistic knowledge and discuss it in a way that may be helpful for others designing systems as collaborators in a creative activity. In the final part of the paper, we draw together these ideas and propose a series of guidelines for designing systems that can be used in creative contexts where improvisation is key. Through this work we hope to inspire new insights into the design of systems that become collaborators with humans engaged in any creative activity.

Background

Mental states that best support a person’s creative activity (known as “flow states”) are most effectively attained when there is a good balance between creative challenges and the person’s skills (Csikszentmihalyi 1997). Encouraging these states is increasingly a consideration in designing new kinds of creative computer systems, particularly now that touch, gestural, and body-based interaction with technology are increasingly popular. Coupled with software that can learn and adapt to individual users, these new technologies present an enormous opportunity to reimagine how people and computers might interact to achieve flow states.

As an individual’s skills improve, the creative challenges must similarly grow in sophistication. Learning mastery in

many creative professions – such as music, dance, performance, and fine art – is a difficult and time-consuming enterprise, requiring extensive personal dedication and perseverance. Typically it takes around 10 years or 10,000 hours of practice to become an expert in any domain (Gladwell 2008). Apart from a few gifted autodidacts, the reason the majority of creative professionals become competent in their field is that they had one-on-one tuition as a child. Without individual tuition it is hard to receive the support and feedback that motivates regular practice (Fig. 1). Consequently, many young people fascinated by the creative arts do not develop skills that would give them the satisfaction and the power to be truly resourceful and imaginative artists – rising to be originators, rather than ordinary consumers of the commercially infiltrated arts of our time.



Figure 1: The system *Music Circle* from Goldsmiths allows human and automatic feedback on music performance. Understanding limitations of automatic feedback helps scope the potential of artificial systems to support improvisation.

Improvisational Interaction Design

As discussed, successful improvisation requires a free and natural interaction. Traditional hardware, such as the 2D screen and mouse, imposes constraints on the range of possible interactions, particularly for improvisation. Coupling new sensor technologies with a computer system that improvises with the artist as it learns and adapts to their individual style creates a powerful new creative learning and performing environment. Orthodox “creative software” (software designed to support humans in creative contexts) is largely construed as a tool derived from the medium’s pre-computational history. Software mimics paint brushes, photographic darkrooms, note pads, architect’s drafting boards, recording studios and traditional instruments. Mass-production requires a standard interface and compliance to the constraints of the underlying machine architecture and operating system. Computers have an increasingly complex and significant influence on creative cultures, so mimicry of pre-computational tools in software seems limiting, particularly when computers offer many new possibilities that previous technologies are incapable of.

Yet the complexity of modern digital tools often prohibits an exhaustive exploration of all their functional possibilities. Users are typically biased and unwilling to explore, or question the software’s fundamental assumptions. As a result they tend to stick with paradigms whose success is modest but at least proven. *They adapt to the software rather than it adapting to them.* Software design and development is

largely an engineering discipline, not an artistic one. Mathematical and engineering conventions frequently dominate the conceptual basis of software design, forcing users to conceptualise their process according to the embedded conventions, limiting the development of creative ideas that can be naturally supported through the software's use.

Computer Improvisation

Computer improvisation involves the simultaneous creation and performance of sonic, visual, physical, or linguistic elements. It may be considered a creative means in itself, or part of a wider process in developing a creative work or idea. Early research, pioneered by Fry (1980), and by Lewis (1999), whose *Voyager* system could accompany its human designer at a professional musical level – was generally constrained to low-level creative tasks or specific artistic genres. A breakthrough came with *The Continuator*, an interactive music system based on variable-length Markov models developed by Pachet (Pachet 2002a). This system could learn and creatively respond interactively to any human musical input, from children with no musical training to jazz virtuosos (2002b). The Continuator builds and refines Markov representations in real-time as the person plays musical phrases into the system. If the player pauses momentarily the system responds with its own phrase, based on accumulated knowledge of the player's previous phrases, but biased toward the most recent. This simple interaction creates, over time, an increasingly complex musical dialogue.

Developing this work further, Pachet and colleagues introduced the concept of “reflexive interactions”: human/machine interactions with a system that attempts to imitate a player's style. The *Reflexive Looper* is a progression of the concept of musical looping, where a learning system allows you to play with past virtual copies of yourself (Pachet et al. 2013) (Fig. 2). The looper can take on different instantiations of a guitarist (for example) playing a bass line, a chord line, and a solo, with each of these instantiations responding to live playing from the performer. The system shares the performer's goal of trying to create great music, and it achieves this by aiming at the best “ensemble” sound possible. The creative activity of musicians is challenged and stimulated by playing with responsive copies of themselves, leading to impressive musical creations that would not have been possible for a musician playing alone.

This system was an important advance in designing systems for improvisational interaction. It enabled “virtual copies” of a musician to be made, allowing them to improvise with themselves.¹ The system's success stems from its ability to evoke genuine musical agency that was often interpreted as autonomous when the machine would “lead”. Because the system knows the tempo, feel and chord sequence it provides the human player with a strong degree of confidence and certainty. The system could also take part in collaborative improvisation by first understanding what the “live” version was doing, such as playing a bass line, the harmonic chords, or a lead solo line. The looper would then “fill in the gaps”, responding to what was happening live by try-

¹<https://www.youtube.com/watch?v=VVgXX1XkzNQ>

ing to use the most musically appropriate segments of what had been played previously. As the performance develops, the representation of you as a performer develops, allowing the system to make increasingly varied and nuanced decisions about what to play while performing. This dialogue challenges and stimulates the live performer to push themselves further, creating yet more ideas that are then added to the growing data base for instantiating musical copies.



Figure 2: The reflexive looper from Sony CSL. This system provides a clear sense of musical agency for the performer.

Another popular general model for improvisational interaction between artists and machines has been the “live algorithms” framework (Blackwell, Bown, and Young 2012). A live algorithm is an autonomous machine that interacts with artists in an improvised setting. It consists of an input module which “listens” to incoming features, an output module that “plays”, and internal modules for analysis, synthesis and patterning that are updated in real-time in response to features extracted by the analysis module and by internal evaluation. The nature and implementation of these modules were deliberately left open, leading to numerous implementations, bespoke to a particular style or researcher. This framework and its successors have predominantly been applied to live music performance.

It remains an open problem how to extend this and other frameworks beyond musical improvisation. As part of the research ambitions described in this paper, we therefore confront three important challenges: (i) how to extend existing frameworks to other creative tasks beyond musical improvisation, including activities such as sketching, arrangement, and composition; (ii) investigate how any generalised framework can play a significant role in improving learning and development of creative proficiency (such as playing a musical instrument, writing lyrics, sketching), particularly in children and younger adolescents; and (iii) how we can develop the most productive improvisational interactions between artists and machines in these expanded settings.

Learning from Improvising Musicians in Jazz

Arguably (at least in the eyes of the 2nd author) the greatest human-made improvisational interface is the piano, and the opportunity afforded for improvisation in jazz. In most situations jazz musicians will play from a set of standard musical repertoires which feature in various *real books* which are

compendiums of jazz standards which consists of the tune and a chord sequence called leadsheets. Typically, the tune is played first, then members of the band *improvise* over the chord sequence - once through a “*chorus*” - with soloists typically take a few choruses each², before the tune is then played again to a finish. The typical constants and variables are as follows:

1. Constants.

- (a) Feel. Do we play latin, swing or bossa?
- (b) Tempo. Up, down or rhuato?
- (c) Key. (Typically musicians stick to the same key. But going up a semitone or tone – *as long as everyone does it together* – is an old trick that often works well.)
- (d) The leadsheet of chords. (Everyone follows this. And if you get lost in the leadsheet never let anyone know!)
- (e) Structure of performance (tune, choruses, tune).

2. Variables.

- (a) Original choice of tune, feel, tempo and key.
- (b) Order of soloists. (Agreed in advance or just emerges.)
- (c) Introductions. (Do you go *straight in* to the song or do you have a rhuato introduction, signalling the speed through the playing?)
- (d) Chord alterations. (The chordal instruments such as piano and guitar are free to move away harmonically from what is written on the lead sheet.)
- (e) Choice of scale. (Good jazz musicians are able to move between different scales that fit over a particular chord in the chord sequence.)
- (f) Notes played. (Where improvisation can take place.)

So in a performance context you might say “Autumn Leaves. Gm. Swing. In 3. Bass solo first. Straight in. One ... , two... one, two, three, four”. In jazz improvisation there is a huge amount that is fixed which enables jazz music to happen without a huge amount of obvious interaction apparent to the audience. Visual signals include “I’m coming to the end of my solo can you go next?” and “follow me on this rall would you?”, but there aren’t many and so the unfolding interactions are very subtle and nuanced and almost completely contained within the music itself.

In free jazz on the other hand there are no constraints. The only constraint is that someone starts and that you have to finish; finishing being much harder than starting much of the time. But in this context – to improvise well – you have to be incredibly responsive. Visual and aural cues are coming in all the time and to work out how to respond, how to support, how to texture, how to subvert, how to challenge and so on requires an ongoing heightened awareness and sensitivity. This is often developed following many years of practice and experience. You make yourself open to any and all possibilities and you hope and expect your collaborators to be

²The story goes that Miles Davis once asked John Coltrane why his solos were so long. John Coltrane replied “I don’t know how to stop.” to which Miles responded with “Try taking the f***** horn out of your mouth.”

doing the same, and to sustain the intensity of music improvisation with others requires that interactions are entirely natural and intuitive.

There are many theories of what happens in improvisation but to do it well requires a deep and virtuosic understanding of how to play an instrument, an instinctive ability to navigate chord sequences, and a deeply-honed musical awareness and sensitivity to what is going on around you.



Figure 3: The Mark d’Inverno Quintet launching the album *Count on It* at London’s leading venue: Pizza Express Soho. The guest sax Gilad Altzman (centre stage) joined for several songs having never played with the quintet before.

Improvisation in tonal jazz consists essentially of creating an (often singable) melodic line which is consistent with the underlying chord sequence (when there is one) utilising notes from these chords and their associated scale or scales to create motivic elements, often starting from elements based on the tune and developing the melodic line into a memorable melodic structure usually with a beginning, middle and end. Some commonly explored improvisational routes would typically be: 1. chordal improvisation – where we typically use notes on the current chord in a chord sequence in any order and run up and down the notes (arpeggiation); 2. scalar improvisation – where we typically run up and down parts of the scale or scales associated with a given chord, starting and stopping anywhere in the scale; 3. motivic improvisation – where we use notes from the associated scale to create an (often singable) musical phrase and then develop this motivic element using various techniques such displacement, rhythmic displacement, inversion, variation and recapitulation; 4. special devices – where we use particular devices with discretion to enhance a solo such as crushed notes, octaves, double octaves, multiple notes in the right hand, different variants of “locked hands” and “bluesy-fication” to add interest to the solo.

Of these, motivic improvisation is often considered the most important – an approach strongly embedded in the courses *Learn to Play Jazz Piano Online*³ by Ray d’Inverno, who has over 60 years as a jazz pianist and close to 50 years as a jazz educator (Fig. 4). The course covers a huge range of material to do with playing jazz piano, indicating that effective improvisation can only happen with a wide and deep range of concrete musical knowledge. Another way to appreciate improvisation is to provide some quotes from the greats that articulate what playing jazz and improvisation means to them, and perhaps sheds light on how we approach

³<https://vimeo.com/99517780>



Figure 4: Learn Jazz Piano Online by Ray d’Inverno. Jazz courses like this give a sense of the enormous scope of technical knowledge needed before improvisation can happen.

the issue of designing systems that are truly responsive to human improvisation.

I realised anytime I came home, the thing I was missing was the sights and sounds from this property. It’s very lush, real winters, real summers. Everything changes all the time, you see struggle and that struggle to me is a parallel to the artistic struggle.”

Keith Jarrett, pianist.

When we’re playing something in straight time, boy! When this thing locks, something else takes over and it’s like you’re not playing ... it’s kind of floating! This level is reached on every track of Standards Live, effortless, as if it is the norm.

Gary Peacock, bassist. Talking about playing with the “standards trio”. (Fig. 5).

I love him because, as a pianist and drummer myself, I can identify with him, the concept of what to ignore, what to leave in, what to leave out – we intuitively understand that – that’s why when we play together we never know what’s going to happen, but we always get something happening that turns us on.

Jack DeJohnette, drummer.



Figure 5: The Standards Trio: Keith Jarrett, Gary Peacock and Jack DeJohnette. Arguably the most accomplished jazz improvisation outfit ever because of a deep connection with the emotionality of the unfolding music and an almost telepathic awareness of each other.

As these quotes illustrate, improvisation embodies many of the complexities of being human. It encompasses learning, life experiences, expectation, virtuosity and skills that are typically developed over many years. When improvisation between players works they respond by articulating states of heightened awareness, well beyond the mechanical act of playing an instrument. Indeed, many aspects of the improvisational experience appear beyond conscious knowledge, or at least its verbal articulation. To give a concrete example, the Mark d’Inverno quintet had played many times together before including an intensive period recording a new album. However, with use of the lead sheet, they were able to invite a special guest – the virtuoso sax player Gilad Altmann – to join them on a couple of the album’s tracks without ever having rehearsed or playing together before (Fig. 3). This may seem “magical” to some audiences but relies on having a clear structure as defined by the lead sheet, a clear set of norms in terms of who is soloing and how the soloist leads the musical journey. However an empathy and awareness of what is unfolding in the improvisation from all musicians so that the band can interact successfully is also vital. In order for improvisation to be facilitated effectively, any computational system will need both the domain knowledge and an on-going sense of the activity of other participants.

In summary, improvisation in modern jazz is making up your own tune which fits with the chord sequence, where a tune consists of musical phrases that are often “singable”. The limitation of this definition is the use of the word “musical”. We can often recognise non-musical improvising (sometimes called “noodling”), where all the notes in use are correct in that they fit the chords and scales, but they do not add up to anything “musically meaningful”. Again, “musically meaningful” is hard to define, although those with a suitably trained ear can mostly agree when it happens. It has something to do with a musical phrase or line having a “shape” or a “structure”, with components identifiable as a beginning, middle and end. Since music takes place in time it is also about how it occurs in relationship to what has come before and what happens afterwards. Perhaps to help with these abstract definitions it is best to listen to the greats of the modern jazz world such as Charlie Parker, Miles Davis, John Coltrane and Michael Brecker and pianists Bill Evans and Keith Jarrett. The musical tradition of jazz can be thought of as a quest, a journey or race where the torch is handed on from one generation to the next, thereby retaining the best of the old but frequently searching for the new.

Learning from Artists

The concept of art and what activities it encompasses has undergone regular revisions in Western culture, particularly in the last 200 or so years. One important shift in emphasis in the process of art-making has been from *problem-solving* to *problem-finding*. In problem-solving the creative emphasis is on how to achieve outcomes – “how do I represent this?”, for example. Problem-solving relies on developing mastery and skills over a working lifetime, hence when, as a society, we value problem-solving in artistic creative activity, the importance of a person’s creative work tends to

increase with their age and experience. An individual's creative peak comes late in their career, in contrast to popular notions that people reach a creative peak at a young age or in the early or middle of their professional careers.

Artists like Cezanne explored a single "problem" for their entire career, and they gradually got better at it; that's why Cezanne's later paintings are worth more. (Sawyer 2011, p. 302)

In contrast, problem-finding shifts the emphasis to the *process of making art* as an exploration, rather than a finished product. Changes in our conception of what constitutes the creative value of an artwork in twentieth century art favoured the problem-finding approach. For example, a major study by Galenson showed the art world increasingly favoured problem-finding artists as art developed in the nineteenth and twentieth centuries (Galenson 2009). Similarly, Csikszentmihalyi found that contemporary problem-finding artists had more successful careers. This shift from problem-solving to problem-finding also brings changes in what we consider "good creativity" in an artwork and when we typically view an artist to be at their creative peak.

Such shifts seem culturally determined. As we discuss below, Western, individualistic cultures emphasise value in originality – problem finding – as a point of differentiation, whereas other, less-individualist cultures place value in the faithfulness of a representation or idiom. Current graduating art shows often appear to display acute diversity, typically with some "standout" works being perceived as far more creative than others. But historically this isn't the case. Looking back at graduating art shows over decades reveals a homogeneity and sameness that is bound with the particular point in time the works were developed, again suggesting that our idea of creative value shifts with time and culture. These observations point to the hypothesis that many factors in our judgement of creative activity and creative value are culturally determined. As time and culture changes, so does the *creative value* assigned to any artefacts produced. As complex computer technology is increasingly pervasive in our culture, we would naturally expect this change to influence how our creative judgment is formed and the value we ascribe to creative activities.

From a more personal perspective, improvisational creative activity in a visual arts practice often involves a nuanced feedback between action and result. The artist is constantly evaluating a work as it emerges, often trying many different ideas or approaches before finally arriving at a fruitful idea. The form and application of this evaluation is quite different than one would get from an audience or reviewer, for example. Sketchbooks – either literal or metaphorical – that allow easy and rapid expression of ideas, anytime, anywhere, support this developmental process. The metaphor of a sketchbook has been successfully applied in the area of creative coding, for example, however as previously discussed physical metaphors translated to software may limit creative expression.

This is precisely the same as music composition or play writing for that matter: there is one part of you creates something, then another part of you that assesses and edits. Wear-

ing two very different hats in the creative process is often difficult for the lone creative, illustrating the potential value of an artificial collaborator that can take on specific and changing roles in human/machine collaboration.

Learning from Psychological Theories

In this section we look at psychological theories of human creativity and discuss how they might inform the design of improvisational systems that support a person in a creative task or act. In broad terms, there have been two major streams of thought about creativity and its locus of influence in human Psychology. These are often referred to as *individualist* and *sociocultural* theories of creativity. Individualist creativity has its origins in associationism theories of Psychology. As the name suggests, its focus is on the individual creative mind and views creativity as a distinct, but general human capability.

A major early goal of individualist research was to quantify and measure individual creativity or creative potential. The idea of being able to predict a person's creative potential was especially popular as an objective methodology to select "gifted" children for accelerated or enhanced learning programs. Over 100 measures of creative ability can be found in the psychological literature, but the most widely known are the *Divergent Thinking Tests* (Runco 1991). These tests typically ask participants to come up with as many unusual uses for common objects (e.g. bricks) as they can in a fixed time period. In this test, scoring is based on the total number of responses and the number of statistically unlikely responses.

While convenient to numerically score and rank a person's creative potential or ability, such tests have many problems (including methodological, design, and correspondence issues) and have been widely contested in the literature. Moreover, they are counterintuitive: individuals considered especially "creative", typically excel only in a specific domain and the psychological literature suggests that many aspects of an individual's creative ability are highly domain specific (Hirschfeld and Gelman 1994; Kaufman and Baer 2004). Doing a short test to think of many different uses for bricks doesn't intuitively seem similar to the actuality of expressing oneself as a creative artist or musician.

Modern European cultures primarily associate individual creativity with novelty and originality, but other cultures focus more on how well an artist's work interprets an existing style or idiom. In such cultures, being new or different is not seen as creative or a positive trait. In highly individualist cultures, such as the US, emphasis is placed on individual creativity and ownership of originality. More than 2,000 patents worldwide are attributed to the American inventor, Thomas Edison, yet most of Edison's inventions were developed by large teams. These cultural differences suggest that the common concept of individual creativity, how it is assessed and evaluated, has significant determination and validation by the culture in which it arises.

It could be argued that the computational creativity community has sometimes favoured this individualist understanding of creativity, with one of its main goals being "to construct a program or computer capable of human-level

creativity”.⁴ In contrast to individualist approaches, socio-cultural creativity considers creativity as a product of social and physical interactions over time. It seeks to explain creative processes through interactions between groups, societies and cultures.

It is often seductive to think of creativity as residing exclusively in an individual, but we believe that any modern creative activity relies heavily on multiple social innovations and incremental discoveries. Contemporary creative artefacts – including skyscrapers, automobiles and computers – are created by multiple groups and organisations, who are distributed globally, connected using complex mechanisms and rely on numerous innovations developed previously. Music, cinema, dance and performance also rely on social creative activity; jazz ensembles and theatre performers innovate collectively and rely on group dynamics to drive the creative process (Sawyer 2003).

Creativity also occurs at a societal level. Systems of trade, complex organisational and distribution structures are not created by any single individual or group. They emerge through a complex interaction between many different individuals, groups, organisations, etc., which take place over decades or even centuries. Creative precincts and cities that have occurred throughout human history, from renaissance Florence to Silicon Valley, are complex creative ecosystems (McCormack 2012) that earn their creative currency from the interactions of many individuals, systems and events, not from any one super-creative individual working in isolation.

In the sociocultural view, creativity *emerges* through these interactions between people, objects and environments. It is incremental and builds on many small discoveries and often chance events. Increasingly, computers are a significant influence in the modern world’s creative ecosystem. So any design of a system to support creativity must include the possibility of creativity being changed by that system itself.

Group Improvisation

Sawyer (2011) lists ten key characteristics of group improvisation, generalising from a number of studies in free jazz improvisation, but also from many years of research in collaborative teams in business, industry, theatre, and so on – anywhere where group improvisation plays an important role. These characteristics can be summarised as follows:

- Provide a strong **match** between the group and the goal;
- Facilitate **close listening**, which can lead to unplanned responses to what has been said;
- Each person must **concentrate** and have complete focus on the task;
- Being in **control** – having the autonomy and authority to execute;
- **Blending egos** – each person’s ideas build on the groups;
- **Equal participation** – everyone participates equally;
- **Familiarity** – tacit knowledge enables better communication;

⁴<http://computationalcreativity.net/home/about/computational-creativity/>

- **Communication.** The group members are always in communication, always talking;
- **Keep it moving forward.** Each person builds on and elaborates the ideas generated by the others;
- **The potential for failure** - the potential for failure motivates peak performance.

These characteristics provide a pathway to developing computationally creative improvisational partners, designed to collaborate with human artists. It is also interesting to compare these characteristics with our earlier explanation of specific examples in free jazz improvisation.

Towards a Theory of Human-computer Collaboration

The literature from psychology has explored how creativity occurs in individuals, groups and societies. To summarise our discussion thus far. The modern, sociocultural view sees creativity as an emergent process that arises through interactions, rather than the romantic idea of the lone, emancipated creative individual. Developing individual creativity takes many years of focused practice and dedication. Creative works develop mainly with small, incremental improvements, typically with many relatively small innovations rather than singular “eureka” moments of deep insight. Regular review, tweaking and feedback is generally what makes creative works great. Being especially creative in one area doesn’t necessarily make you very creative in others.

Moreover, popular understanding of what constitutes creativity and how we assign creative value varies according to time and culture. Most people are able to place a piece of music in the decade it was written, even if they have never heard the song before and it originated before they were born. When viewed historically, artworks from any specific period appear similar in style and influences, specific to the period of their creation, even if at the time there were large differences in their reception and perceived creativity.

These findings are important considerations for designing human-computer collaborative systems. As a starting point, below we present a set of guiding principles that we believe are necessary for building computationally creative improvisational systems. Our guiding principles are:

- creative activity is supported by social interaction, therefore we need a social infrastructure that supports both human and machine agency on an equal footing; (note that this not suggesting that the machine is necessarily creative in its own right. We leave that question for others, only that it brings creative agency or even creative autonomy to a specific creative context);
- proficiency takes many years of dedicated practice to develop mastery of a specific creative activity or discipline. The idea of “general creativity” doesn’t correlate with the specificity observed in most creative domains; (as we touched upon playing jazz piano requires a huge amount of knowledge and practice, without this strong base effective group improvisation is simply not possible);

- the challenges and responses must grow in proportion to each individual's development. Tasks that achieve a good balance between challenge and skills work best, so the computational system must change with each individual to support their creative development and virtuosity;
- people learn and flourish *into* their creative practice, they need support and encouragement but also critical feedback on how to improve; giving and receiving feedback on our developing practice is arguably a critical role for future systems (Fig. 1);
- in early development, free-play readily encourages creative exploration;
- interactive communication between active participants needs to be facilitated, often non-verbally.

Conclusions

Taken together with the characteristics of group improvisation, these guiding principles point to a way forward for designing collaborative machines with their own creative autonomy, that support improvisational development of artists. Clearly, we have only articulated basic principles, not described detailed designs for specific systems. Our research is not yet at this stage, but these principles – together with our understanding of the successful improvisational systems described earlier in the paper – form the basis for further investigations as we work towards developing computational systems that can significantly enhance and broaden both individual and group human creative activity.

We have looked at a number of findings from Psychology with regards to the concept of “improvisational creativity” in humans to see what we might learn as designers of systems interested in supporting and provoking the human creative. Additionally, we have described how improvisation flows in non-computational settings and drawn a set of broad guiding principles from this work. We hope that these insights will be helpful for the design of systems supporting human-computer improvisation (Yee-King and d’Inverno 2016).

Acknowledgments

This research was supported by Australian Research Council Discovery Projects grant (DP160100166) and the EU FP7 Project Praise (FP7- 318770). We will always be grateful to Arthur Still, Matthew Yee-King and François Pachet for many ongoing discussions on these ideas.

References

Blackwell, T.; Bown, O.; and Young, M. 2012. Live algorithms: Towards autonomous computer improvisers. In McCormack and d’Inverno (2012). Chapter 6, 147–174.

Bown, O., and McCormack, J. 2009. Creative agency: A clearer goal for artificial life in the arts. In Kampis, G.; Karsai, I.; and Szathmáry, E., eds., *ECAL (2)*, volume 5778 of *Lecture Notes in Computer Science*, 254–261. Springer.

Cope, D. 1991. *Computers and Musical Style*, volume 6 of *The Computer Music and Digital Audio Series*. Oxford: Oxford University Press.

Csikszentmihalyi, M. 1997. *Flow and the Psychology of Discovery and Invention*. New York, N.Y.: HarperPerennial.

Dewey, J. 1934. *Art as Experience*. New York: Perigree Books.

d’Inverno, M., and McCormack, J. 2015. Heroic vs collaborative AI for the arts. In *Proceedings of IJCAI 2015*, 2438–2444.

Fry, C. 1980. Computer improvisation. *Computer Music Journal* 4(3):48–58.

Galenson, D. W. 2009. *Painting outside the lines: Patterns of creativity in modern art*. Harvard University Press.

Gladwell, M. 2008. *Outliers, the Story of Success*. Allen Lane.

Hirschfeld, L. A., and Gelman, S. A., eds. 1994. *Mapping the mind: Domain specificity in cognition and culture*. Cambridge [England]; New York: Cambridge University Press.

Kaufman, J. C., and Baer, J. 2004. Sure, I’m creative—but not in mathematics!: Self-reported creativity in diverse domains. *Empirical Studies of the Arts* 22(2):143–155.

Lewis, G. 1999. Interacting with latter-day musical automata. *Contemporary Music Review* 18(3):99–112.

McCorduck, P. 1990. *Aaron’s Code: Meta-art, Artificial Intelligence and the Work of Harold Cohen*. W.H. Freeman.

McCormack, J., and d’Inverno, M., eds. 2012. *Computers and Creativity*. Berlin; Heidelberg: Springer.

McCormack, J. 2012. Creative ecosystems. In McCormack and d’Inverno (2012). Chapter 2, 39–60.

Pachet, F.; Roy, P.; Moreira, J.; and d’Inverno, M. 2013. Reflexive loopers for solo musical improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2205–2208. ACM.

Pachet, F. 2002a. The continuator: Musical interaction with style. In *Proceedings of ICMC*, 211–218. Göteborg, Sweden: ICMA.

Pachet, F. 2002b. Playing with virtual musicians: the continuator in practice. *IEEE Multimedia* 9(3):77–82.

Runco, M. A. 1991. *Divergent thinking*. Westport, Connecticut: Ablex Publishing.

Sawyer, R. K. 2003. *Group creativity: Music, theatre, collaboration*. Mahwah, NJ: Erlbaum.

Sawyer, R. K. 2011. *Explaining creativity: The science of human innovation*. 2nd Edition. Oxford [England]; New York: Oxford University Press.

Still, A., and d’Inverno, M. 2016. A history of creativity for future AI research. In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016)*. Université Pierre et Marie Curie.

Yee-King, M., and d’Inverno, M. 2016. Experience driven design of creative systems. In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016)*. Université Pierre et Marie Curie.

MODELS OF CREATIVITY



Visual Hallucination For Computational Creation

Leonid Berov and Kai-Uwe Kühnberger

Institute for Cognitive Science
University of Osnabrück
49076 Osnabrück, Germany
{lberov, kkuehnbe}@uos.de

Abstract

Research on computational painters usually focuses on simulating rational parts of the generative process. From an art-historic perspective it is plausible to assume that also an arational process, namely visual hallucination, played an important role in modern fine art movements like Surrealism. The present work investigates this connection between creativity and hallucination.

Using psychological findings, a three-step process of perception-based creativity is derived to connect the two phenomena. Insights on the neurological correlates of hallucination are used to define properties necessary for modelling them. Based on these properties a recent technique for feature visualisation in Convolutional Neural Networks is identified as a computational model of hallucination. Contrasting the thus enabled perception-based approach with the Painting Fool allows to introduce a distinction between two distinct creative acts, sketch composition and rendering.

The contribution of this work is threefold: First, a computational model of hallucination is presented and discussed in the context of a computational painter. Second, a theoretic distinction is introduced that aligns research on different strands of computational creativity and captures the differences to current computational painters. Third, the case is made that computational methods can be used to simulate abnormal mental patterns, thus investigating the role that “madness” might play in creativity – instead of simply renouncing the myth of the mad artist.

Introduction

Computational creativity research often stresses that the creative act is a rational process instead of a divine gift or the byproduct of madness. But while it is certainly true that “one does not need to be [...] an ear-lobbing manic-depressive to be creative” (Veale 2012, p. 16), it is also the case that some creative artefacts owe their uniqueness precisely to the workings of a deranged mind. Self-reports indicate that visual hallucinations were an important source of inspiration for many artists. Some, like van Gogh, were involuntarily influenced by the changes of perception inherent in their psychological disorders (van Gogh 1889). Others, like Joan Miró, willingly induced hallucinations to draw creativity from the arational (Phillips 1948). And while self-reports are not necessarily reliable evidence, recent findings will be introduced

that also establish quantitative evidence for this connection. Such findings can be seen as contradictory to a view that rejects the arational as corroborating the myth of the “mad genius”. By deriving a computational model of hallucination, and explaining how it can be used in a computational painter, the present paper will illustrate how arational processes can be employed as generative computational models of creativity.

For this we will first present the mentioned art-historic findings on the role of hallucination in creativity. Thus motivated, we will investigate from a psychological perspective how a creative process based on perception (be it normal, or aberrant) can be formulated. After having identified the role of hallucination in such a process, we will descend one level of abstraction to outline how hallucination is implemented in the human brain. This will allow us to derive functional properties that a computational model of hallucination must possess. These properties will be used to argue that a recently introduced technique for visualising features of Deep Convolutional Neural Networks (ConvNets) can be used to simulate hallucination. This argument will be partially validated by demonstrating how three phenomena associated with hallucination can be modelled with this technique. Coming back to the introduced psychological process of perception-based creativity we will present how such a model of hallucination can be used to implement a computational painter, and discuss it in the context of the current state of the art.

Taking all together, our goal is not just to present a computational model of hallucination as a potential source of inspiration, but also to broaden the scope of computational creativity. By providing a case study on how to model a part of creativity that is arational, we argue that accepting abnormal mental patterns as potential sources of creativity does not imply yielding to the myth of the “mad genius”.

Hallucinations in the Fine Arts

The role hallucination plays in the fine arts is most apparent in modern art movements due to their departure from the primacy of naturalistic depiction. Post-impressionist artwork, for instance, is characterised by depictions of the artists’ subjective impression of a scene – something that can be influenced by perceptual disorders. Most notable in this context is painter Vincent van Gogh, who increasingly

suffered from psychotic episodes including visual hallucinations (Blumer 2002) and had to move to mental asylum in 1889. This development was accompanied by a noticeable, qualitative change in his style, tending to wavy lines and thick, intensive colouring. Van Gogh himself noted this connection in a letter where he states that “some of my pictures certainly show traces of having been painted by a sick man” (van Gogh 1889). This can be backed by quantitative evidence, as artwork from van Gogh’s psychotic phases appears to capture mathematical properties of light-turbulences in a way that artwork from healthy phases does not (Aragón et al. 2008).

Even more relevant is surrealist artwork, which is characterised by the drive to capture the subconscious. This can be taken quite literally, since one group of surrealists intended their art to be an “exact transcription of personal hallucination” (Frey 1936). In fact, Frey emphasizes that in this approach hallucination is to be considered “antecedent” to the painting, which, in turn, is just a means for “immediate fixation of the violent [...] images that haunt the brain”. Consequently, healthy surrealists resorted to artificial means for inducing hallucinations, like Joan Miró, who painted from hunger hallucinations: “I began gradually to work away from the realism I had practiced [...] until, in 1925, I was drawing almost entirely from hallucinations” (Miró, qtd. in Phillips 1948). A qualitative analysis suggests that also Max Ernst was influenced by visual hallucination, as all spatial properties of hallucinatory phenomenology were identified in his artwork (Keeler 1970).

A thorough analysis of art-historic material is outside of the scope of this paper. However, what we have shown is that (1) hallucination can be systematically related to modern art agendas, that (2) artistic self-reports support such theoretic conceptions and that (3) qualitative and quantitative evidence corroborate artists’ claims.

Perception-based Creativity

Psychological inquiry on the artistic use of hallucination can be found in a discussion of the role of perception in creativity, which identified two relevant types of mental processes (Flowers and Garbin 1989): The first type are executively controlled perceptual processes like mental imagery or selective attention. These processes can be used to generate novel mental representations by effortful construction. Individuals with superior control of such faculties derive their creative abilities from the scope and complexity of available mental operations. Processes of the second type, on the other hand, are involuntary because they are based on the perceptual organisation of input data, which is performed automatically by the visual system. Individuals whose perceptual organisation operates less deterministically, or fundamentally divergent from what is typical, can derive novel mental representations straight from their percept. Their creativity stems literally from seeing things in an unusual way. Creative behaviour usually results from a combination of both types, with the emphasis shifting from one individual to the other. The role of hallucinations can be identified as one possible source of “loose” perceptual organisation.

Because “common mental resources are used in executive control of mental representations and processing of corresponding forms of sensory data” (Flowers and Garbin 1989) the authors state that interference effects can occur when processes of different types happen to coincide temporally. This will become relevant later, when we show that such behaviour can actually be observed in the proposed computational model. Flowers and Garbin furthermore point out that conceiving a creative artefact includes selection processes in order to identify if a mental representation is novel and valuable. This can be especially hard for individuals with a loose perception, since it involves hypothesizing about the judgement of non-aberrant perceivers.

A widely accepted psychological model of creativity (Csikszentmihalyi 1997) postulates a five steps process:

1. preparation: gathering knowledge and values of the relevant domain,
2. incubation: subconscious combination, consolidation and re-organisation of knowledge,
3. insight: unexpected event, the surfacing of an idea,
4. evaluation: deciding weather the idea is novel and valuable,
5. elaboration: detailed concretisation and implementation of the idea.

The process is not linear but rather recursive in nature, and especially the last three steps can reoccur iteratively thereby informing each other.

The first two steps of this process can not be linked to the account of Flowers and Garbin directly, however, preparation and incubation could be mapped to the maturation and knowledge acquisition of the visual system, most prominently during infancy, which must be an implicit precondition for any perception-based account. An actual creative process would thus start with the insight phase. In the context of Flowers and Garbin this can be taken to be the generation of an unconventional percept by a loose perception-process; the characteristic phenomenology of an insight-event is attributable to the involuntariness of sensory organisation. The evaluation step can be connected to Flower’s selection processes, while elaboration, taken to be the most effortful step, maps well to Flower’s description of executively controlled construction. We thus arrive at an iterative, perception-based process of creativity: (1) loose perceptual organisation, (2) selection and (3) executively controlled construction.

This synthesis could in principle serve as the basis of a computational model of hallucination based creativity. However, while there is work on computational accounts that might be dubbed effortful construction (Cohen-Or et al. 2006; Bhattacharya, Sukthankar, and Shah 2010) and aesthetic selection (Li and Chen 2009; Luo, Wang, and Tang 2011; Yao et al. 2012), to the best of our knowledge, no work has been done on computational models of the sensation of visual hallucination-based loose perception. In order to devise such a model we first need to understand how visual hallucinations are implemented in the brain.

Neurological Correlates of Hallucination

Visual sensory information is ambiguous. Thus in order to generate a stable, unambiguous percept, and perform higher-order tasks like object recognition, a processing of the input data has to take place in the visual cortex (Teufel et al. 2015). The primate visual cortex is comprised by a hierarchical system of specialised brain areas. Lower areas are responsive to primitive visual features like oriented gratings, while higher areas use information from lower layers, and are responsive to complex features like e.g. faces, houses or landscapes (Zeki et al. 1991; Felleman and Van Essen 1991). The visual system thus combines bottom-up sensory input processing with top-down predictions based on prior-knowledge of the environment.

Hallucinations occur when the balance in information processing shifts to prefer this knowledge over sensory evidence (Teufel et al. 2015; Mocellin, Walterfang, and Velakoulis 2006). Mocellin and colleagues take hallucination to be a “sensory perception that has the compelling sense of reality of a true perception but that occurs without stimulation of the relevant sensory organ”. Functional imaging has shown that visual hallucinations (at least within the Charles Bonnet Syndrome¹) correlate with increased cerebral activity in specialised visual cortex areas: “colour hallucinations [are] accompanied [by] increased activity in cortex specialized for colour; face hallucinations, increased activity in cortex specialized for faces [...] and so forth.” (Santhouse, Howard, and Ffytche 2000). A recent study by Mégevand et al. (2014) was actually able to induce complex visual hallucinations (CVH) of outdoor scenes in non-psychotic subjects, by applying direct electrical stimulation to the parahippocampal place area. This implies a causative connection between increased activity of specialised areas and hallucinations.

For our purpose we can sum up hallucinations to be the product of the visual cortex where the processing balance between input and prior knowledge shifted towards the latter, for instance due to an artificial increase of activity in a specialised brain area, resulting in a percept that is not rooted in sensory information. Abstracting away from the neurological implementation in humans, this would mean a system that (1) performs visual processing, is (2) comprised by specialised subsystems, and where (3) increasing a subsystem’s activity leads to the generation of a visual representation that has no correlate in the input image but in the knowledge encoded in the respective subsystem. We thus have derived three properties that a system needs to demonstrate in order to be taken to model hallucinations.

A Computational Model of Hallucination

The state-of-the-art approach to many computer vision problems are Deep Convolutional Neural Networks, a specific type of the Multilayer Perceptron (MLP) that is informed by

¹The Charles Bonnet Syndrome describes visual hallucinations correlating with a partial loss of vision (Burke 2002). Mocellin et al. argue that the distinction between CBS and lesion-based hallucinations is not clear. Thus these findings might generalise.

the workings of the mammalian visual cortex (LeCun et al. 1998).

Simple-cells in the primary visual cortex are sensitive to a small part of the retinal image, the so called *receptive field*. Neighbouring cells are processing neighbouring parts of the retinal image and have overlapping receptive fields which results in a topographical map of the input. This is beneficial due to the specific statistical properties of natural images, specifically the strong spatially-local correlations. Analogously, units in the convolutional layer of a ConvNet are only connected to a small subset of neighbouring units from the previous layer, instead of being dependent on all the units of the input, like it is the case in the fully connected layers of conventional MLPs. Because of this, the filters computed by each unit are not responsive to variations outside of their respective receptive field – they are just responsive to spatially local patterns. However, stacking several convolutional layers allows the receptive fields of units from deeper layers to become bigger with respect to the input image, and facilitates the detection of more complicated, global patterns.

Pattern-detectors that are useful in one part of the input-image are likely to be of use in other parts as well. This is exploited by ConvNets by employing parameter sharing. Each convolutional layer is organised in planes, consisting of units whose combined receptive fields cover the complete input-layer. The same parameters are used to compute the activation of each unit of a plane, which results in the same filter being applied on each patch of the input. Since the activation of units from a plane indicates the presence of the encoded pattern in the respective patch of the input, the output of each plane is referred to as a *feature-map*. Mathematically this operation can be described as a convolution of the filter-function with the input image. Usually, each convolutional layer is comprised by several different feature-maps.

The precise positions of a detected pattern is not as important as its position relative to other features. This allows ConvNets to perform a sub-sampling of the feature-maps computed by the convolutional layers by using pooling layers. These layers split the input into non-overlapping regions and compute a function (usually *max*) over the activation of all the units of each region. Thus the feature-map of a pooling-layer encodes the information whether a pattern was detected in a certain region of the image, without storing its precise position.

These three architectural idiosyncrasies significantly reduce the number of parameters involved in training ConvNets, as compared to conventional MLPs. This allows to create much deeper networks, which is quite beneficial if we consider that with each additional layer more complex, position-invariant features can be detected. Nevertheless, a significant amount of data is required to train a ConvNet, with state of the art approaches leveraging over one million of labeled images (Szegedy et al. 2015).

Investigations have been performed on the nature of the feature-representations learned by ConvNets. As it turns out, the features are not random and even interpretable (Zeiler and Fergus 2014). The first convolutional layer, seems to consistently learn to be responsive to (among others) oriented gratings (Zeiler and Fergus 2014; Krizhevsky,

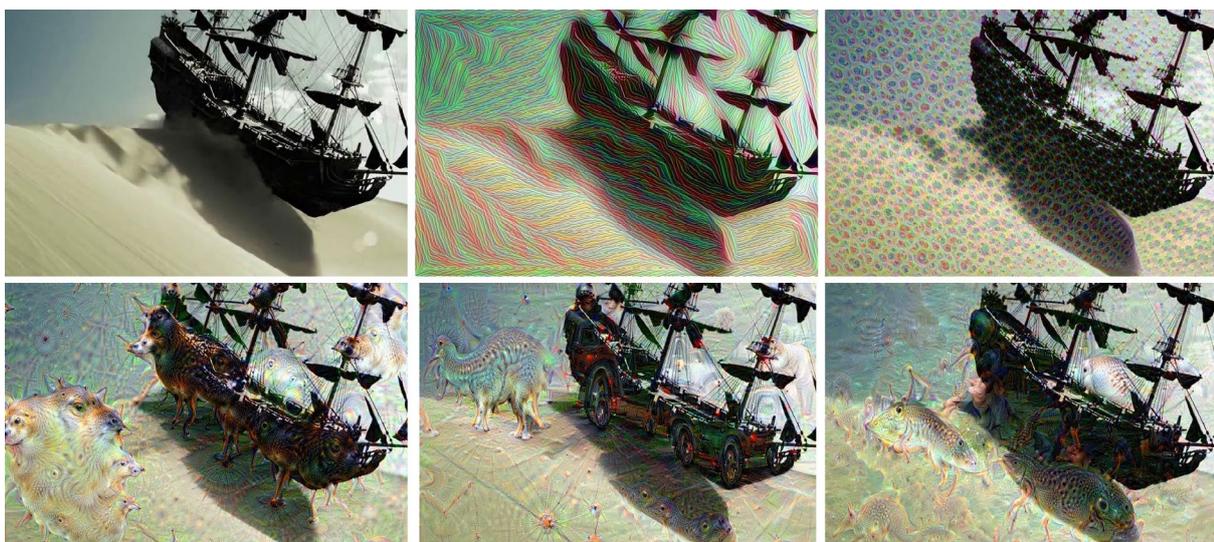


Figure 1: Pictures produced by deep dream. Left-top corner: Original image. From left to right and from up to down the target-layers were changed from lower-level to higher-level layers while keeping all other parameters constant. Enhanced features rise in complexity accordingly. It should be noted how especially in the last two exemplars the enhanced features are integrated into the scene. Also of interest is the fact that the enhanced animals appear to be hybrids. This is presumably the result of maximizing several units in one layer, which are specialised in recognizing different animals. Given this interpretation, the hybrids can be considered concept-blends. Best viewed on screen using zooming.

Sutskever, and Hinton 2012), which, incidentally, is also the case in the first mammalian visual cortex. Zeiler and Fergus have also shown, that features from higher levels exhibit many interpretable properties like compositionality and invariance to spatial operations like mirroring. Effectively it seems that features from lower layers represent properties of *image appearance*, while higher layers represent more and more abstract notions of the *image content* (Mahendran and Vedaldi 2015). If higher-level features are used to reconstruct an image they “invert back to a composition of parts similar but not identical to the ones found in the original image” (Mahendran and Vedaldi 2015). This all indicates that layers in ConvNets are indeed hierarchical, and that higher-level layers are specialised: they identify complex, interpretable objects like e.g. houses or faces. Thus, they exhibit the first two properties of a model of hallucination. What is lacking is a way to generate interpretable, visual representations by increasing the activity of the specialised layers.

Exactly that is accomplished by *deep dream*, a third approach to understand ConvNet feature representations, that focuses on visualising what was learned by individual layers (Mordvintsev, Olah, and Tyka 2015): An input image is forward-propagated through a fully-trained network. Starting from the layer to be analysed, back-propagation is performed in a way as to maximize the Euclidean Norm of activations in the target layer. However, unlike in usual training, the parameters of the network remain unchanged and a gradient ascent step is instead applied to the input image. Basically, the input-image is trained to maximize target-layer activation. To achieve better visibility of the changes in the

image, Mordvintsev, Olah, and Tyka iteratively repeat this step several times while regularly increasing the scale of the input image. This results in a visual enhancement, as well as adaptation, of features that were already present in the input, and the produced pictures have been described as “trippy” and “visually pleasing” (Koch 2015). What type of features are affected depends on the choice of target layer (see fig. 1). When lower-level layers are targeted, primitive features like oriented gratings are enhanced. Mid-level layers enhance simple objects like eyes and geometric forms, while high-level layers enhance complex objects like buildings or animal-blends in a pareidolia-like fashion.

Applying deep dream to a ConvNet results in a hierarchical system for visual processing, where increasing the activity of a layer results in input-data augmentations that are related to the knowledge encoded in the respective layer. It thus displays all three properties of a functional model of hallucination, and we argue that the images generated by deep dream can be considered computational hallucinations. Indeed Koch reports that a remarkable resemblance between the produced images and hallucinations induced by LSD has been widely noted on the internet. Details on how to affect the visuals created by the model, and how to simulate several phenomena that are related to hallucination, will be discussed in the next section.

Discussion

The following results were all achieved using Berkley Vision and Learning Center’s open-source reimplementa²

²https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet

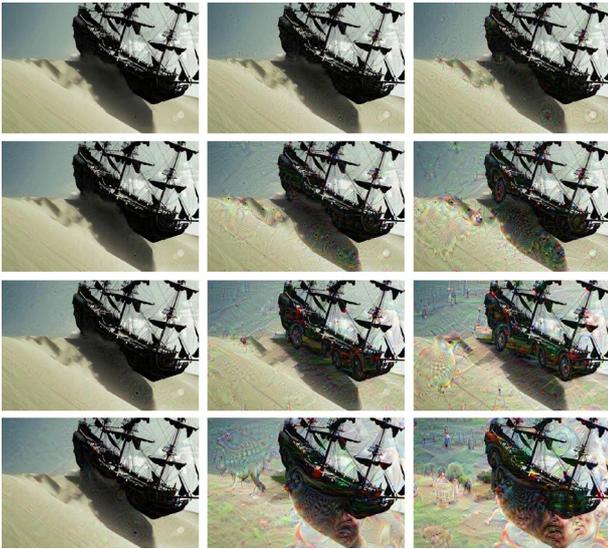


Figure 2: Effect of different scale and iteration settings. Rows (from up to down): 1, 3, 5 and 7 scales. Columns (from left to right): 1, 10 and 19 iterations. Each image was produced by an individual run with the given parameters and the target-layer *inception (4c)*. Best viewed on screen using zooming.

of GoogLeNet (Szegedy et al. 2015), one of the winning models of the ILSVRC 2014 classification challenge (Russakovsky et al. 2014). The architecture is a ConvNet with 22 layers. It employs convolutional layers of varying filter-sizes alternating with pooling-layers for spatial down-sampling. The last layer is a fully-connected soft-max classifier, that follows a 40% dropout layer (Hinton et al. 2012) to prevent overfitting. Furthermore, rectified linear activation (Krizhevsky, Sutskever, and Hinton 2012) is used. Like the original network, the reimplementation was trained on the 1.2 million labeled training-images of the ILSVRC 2014 dataset. For details, especially on the combined convolutional layers called inception and introduced by GoogLeNet, please refer to the original publication.

Parametrization

From the several knobs and levers afforded us by the deep dream algorithm the most relevant for visual appearance were identified as the *target layer*, the *number of iterations per scale* and the *number of scales*. Most importantly, the target layer influences the complexity of the generated hallucinations. For instance the second image of fig. 1 was generated using the second convolution layer and the enhanced features are colourful, oriented gratings. The last image of fig. 1, on the other hand, was generated using the tenth convolutional layer (*inception 5a*) and the enhanced features are hybrid fish-like creatures. As noted earlier, intermediate layers enhance features of varying, but rising complexity. No regularities could be identified between images generated from layers that share a type (like for instance the max-pool layers) but differ in their respective position in the network.



Figure 3: Picture produced by applying deep dream on a white-noise image. The employed parameters were: 8 scales, 50 iterations per scale and 100 repetitions of the algorithm; the target-layer was *inception (5a)*. Several distinct creatures emerged despite a complete lack of statistical structure in the input. This effect simulates the shift in processing balance from input-data to prior-knowledge that happens during hallucinations. Best viewed in colour.

As for the two other parameters, a larger number of iterations increases the intensity of the enhanced features, while a larger number of scales increases the size, number and detail-grade of the enhanced features (see fig. 2).

Simulating Hallucination-Related Phenomena

Apart from following our definition for a functional model of hallucination, and producing qualitatively plausible outputs, the deep dream system is capable of modeling several hallucination-related phenomena:

One is that hallucinations can be induced by sensory deprivation, be it artificial (Merabet et al. 2004) or due to a medical condition (Burke 2002). This can be simulated by repeatedly applying the deep dream procedure to a white-noise image, while significantly increasing the number of scales and propagation steps (see fig. 3). Although a noise-input provides no structure for the ConvNet to detect, network-internal noise nevertheless results in layer activation. This random activation is propagated to the input image and results in discernible but random effects. The stabilisation to a distinct structure is due to a complete shift in processing balance from input data to prior-knowledge encoded in the network.

Another phenomenon is that CVH usually follow a limited number of typologies (Santhouse, Howard, and Ffytche 2000; Mocellin, Walterfang, and Velakoulis 2006), e.g. disembodied, distorted faces or small figures in costumes. The same holds for images produced by the deep-dream system which display some patterns like eyes or dog-shaped creatures more often than others, which is suggestive of basins of attraction.³ The type of these basins is dependent on the dataset used for training the ConvNet and favours features

³As discussed in <https://github.com/google/deepdream>



Figure 4: Picture produced by guided dreaming at a high-level layer, an effect that can be taken to simulate the interference of temporally coincident perception- and manipulation-processes operating on the same type of data. Bottom-right corner: guiding image. The employed parameters were: 6 scales, 20 iterations per scale and the target-layer was *inception (5a)*. Best viewed in colour.

that were over-represented.

A last phenomenon is connected to perception-based systems in general, and was reported in the section on loose perception: Flowers and Garbin state that interference effects should occur when processing of sensory input coincides with the active manipulation of mental representations of the same type. This can be simulated using a technique called *guided dreaming*. For that, in a preparatory step, a guiding image is forward-propagated through the ConvNet and the layer-activity is noted. In the deep dreaming procedure the optimization objective is then changed to maximizing the dot product of input-image activation and guiding-image activation at the target layer. In that way only features that were detected in both images are enhanced, which especially at higher-level layers result in a spilling-over effect from the guide to the output image (see fig. 4).

While not directly relevant for the simulation of human phenomena, it shall be noted that our model allows the combination of guided dreaming with input-deprivation. This produces an interesting, artistic effect where shapes and features from the guide are transferred and randomly reassembled in the output image, resulting in a “colorful, free improvisation on the theme of the guide” (see fig. 5).

Role in Computational Painting

Of course a computational model of hallucination alone is not sufficient for a computational painter. There are two possibilities of incorporating the introduced model in the broader context of the previously outlined model of perception-based creativity. One is to rigidly define all parameters that influence the visual properties of generated images, which seems an obvious approach considering that it corresponds to something we would call perceptual disorder in a human. The other option is to leave these parameters (especially the target-layer for each iteration) variable, and



Figure 5: Picture produced by combining guided dreaming with input-deprivation using the target-layer *inception (4a)*. Bottom-right corner: guiding image. Best viewed in colour.

allow them to change depending on the results of the selection step. This creates a feedback-loop and potentially allows for the emergence of a particular style. This approach is less plausible from a psychological perspective since it seems to imply the volitional adaptability of perceptual disorders. From a computational perspective, however, it is a more promising option since it hands over more creative freedom to the system. In order to compare both approaches an implementation of the whole process needs to be realised.

Several options for implementing selection-processes can be explored. This includes training classifiers based on art-theoretic high-level features (Li and Chen 2009) or a recently introduced creativity-score that is a measure for a painting’s “abstraction in shape and form” as well as its “texture and pattern” (Elgammal and Saleh 2015). Approaches for implementing executively controlled construction include colour harmonisation (Cohen-Or et al. 2006), composition-enhancement (Bhattacharya, Sukthankar, and Shah 2010) and style-imitation (Gatys, Ecker, and Bethge 2015). It seems promising to use video-input instead of individual scenes, because this provides the selection system with a variety of perspectives to choose from. This not only appears to be a more natural context for a perception-based system but also transfers more artistic responsibility to the software. Such an implementation is currently in progress.

Related Work

Other attempts have been made to create computational models of hallucination (Jardri and Denève 2013). However these models focus on simulating cortical activity at different levels of abstraction. Our approach, on the other hand, does not claim to make predictions about structural properties of hallucination. Instead it operates on a functional level, by modeling the effect of visual hallucination. To the best of our knowledge this work is the first computer model that can simulate the sensation of visual hallucination.

Our overall goal was to discuss the role of visual hallucination in creativity. Several systems exist that are concerned with the computational accounts of the fine arts. One of

the earliest is called AARON and has been maintained for over 40 years by artist Harold Cohen (McCorduck 1991). AARON differs from our proposed system in two major ways: First, while it is definitely more than a mere artistic tool, it was not designed to function independently from Cohen, who actively takes part in its creative process. Second, AARON is mainly concerned with figurative and abstract art and does not draw inspiration from real-world scenes.

Our work shares the spirit of the currently most prominent computational painter, Colton's Painting Fool (Colton 2012). Colton's goal is to create "an automated painter which is one day taken seriously as a creative artist in its own right". His work was mostly concentrated on enabling the system to create painterly renditions of photographs simulating different natural media, and on choosing the most appropriate style to do so (Colton, Valstar, and Pantic 2008). This was criticised by artist Faure-Walker as a lack of imagination and creative intent (Colton 2012). The approach proposed here is, on the contrary, concerned with modifying the input in a meaningful way by changing its content and composition based on a systematic misperception. It thus addresses Faure-Walker's criticism by drawing imagination from an unconventional way of perception and making its intention one of sharing this unique type of impression, much in the fashion of modern artistic movements like Post-Impressionism.

Based on this analysis we propose to differentiate between two creative acts: *sketch-composition* (what to draw) and *rendering* (how to draw it). These two differ significantly in the involved problems (e.g. composition, colouring, symbol-language or intention in the first case and material, stroke-type, colour-palette and level of detail in the latter) as well as in the intended outputs (a mental sketch or idea in the first case and an artistic artefact in the latter). A similar distinction is already successfully employed in computational storytelling (Gervás 2009), where creative systems are concerned with creating *fabula* (what is told) or *discourse* (how it is told). Thus our proposal helps to align research in different strands of computational creativity by drawing out the differences between the conception of a work of art, and its implementation. It also helps to disentangle research on computational painting, since such a division is for instance applicable to recent advances on the Painting Fool, because the system judges its rendered artefact by comparing them with previously generated sketches (Colton et al. 2015).

With the distinction between sketch-composition and rendering in mind, a combination of the system proposed here and the Painting Fool becomes plausible. The former can select appropriate scenes and generate a sketch based on loose perception and effortful construction, potentially grounded in art theory. The latter can select an appropriate rendering style and render the sketch accordingly, thus resulting in a more complete model of a human painting process.

Conclusion

Starting from the observation that some painters, especially from modern art movements, drew inspiration from natural or artificially induced perceptual disorders we performed

an investigation of the role of visual hallucination in creativity. For that the neurological correlates of hallucinations were outlined and criteria were derived that a computational model must meet in order to be considered a functional model of hallucination. Subsequently we argued that deep dream, a technique for ConvNet feature visualisation, meets all the necessary criteria and can be functionally compared to inducing hallucinations by electrically stimulating specialised brain areas. This conclusion was further corroborated by showing how several phenomena connected with hallucination can be simulated using deep dream. On a technical level this might be a straightforward exploration of the deep dream tool. What is relevant here, however, is not *how* deep dream changes images, but rather *what* these changes constitute. The significance of this exploration is on a conceptual, rather than a technical level, by partially validating the proposed model.

Just having hallucinations does not necessarily make an artist. Based on psychological research on the role of perception in creativity we derived a three-step process that illustrates how hallucinations can be used for creative insight. Taking this as a framework we then outlined possible avenues for implementing a misperception-based computational painter. Contrasting this implementations with current work on computational painters allowed us to introduce the distinction between sketch-composition and rendering, two distinct creative acts that are both necessary for a successful painter but involve very different processes.

Thus the contribution of the present work is threefold. First, it demonstrates an algorithm that allows computational painters to draw inspiration from systematically misperceiving input scenes. By that it, second, makes the case for a broader approach to creativity that, instead of renouncing the myth of the mad artist, uses computational methods to simulate abnormal mental patterns to further understand the role that madness might play in creativity. Third, it introduces a theoretic distinction, which helps disentangle different processes involved in implementing computational painters, and aligns research on computational painters and computational storytellers.

References

- Aragón, J. L.; Naumis, G. G.; Bai, M.; Torres, M.; and Maini, P. K. 2008. Turbulent Luminance in Impassioned van Gogh Paintings. *Journal of Mathematical Imaging and Vision* 30(3):275–283.
- Bhattacharya, S.; Sukthankar, R.; and Shah, M. 2010. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the international conference on Multimedia*, 271–280. ACM.
- Blumer, D. 2002. The illness of Vincent van Gogh. *American Journal of Psychiatry*.
- Burke, W. 2002. The neural basis of Charles Bonnet hallucinations: a hypothesis. *Journal of Neurology, Neurosurgery & Psychiatry* 73(5):535–541.
- Cohen-Or, D.; Sorkine, O.; Gal, R.; Leyvand, T.; and Xu, Y.-Q. 2006. Color harmonization. In *ACM Transactions on Graphics (TOG)*, volume 25, 624–630. ACM.

- Colton, S.; Halskov, J.; Ventura, D.; Gouldstone, I.; Cook, M.; Blanca, P.; and others. 2015. The Painting Fool Sees! New Projects with the Automated Painter.
- Colton, S.; Valstar, M. F.; and Pantic, M. 2008. Emotionally aware automated portrait painting. In *Proceedings of the 3rd conference on DIMEA*, 304–311. ACM.
- Colton, S. 2012. The Painting Fool: Stories from Building an Automated Painter. In McCormack, J., and d’Inverno, M., eds., *Computers and Creativity*. Berlin, Heidelberg: Springer Berlin Heidelberg. 3–38.
- Csikszentmihalyi, M. 1997. Flow and the Psychology of Discovery and Invention. *HarperPerennial, New York* 39.
- Elgammal, A., and Saleh, B. 2015. Quantifying Creativity in Art Networks.
- Felleman, D. J., and Van Essen, D. C. 1991. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex* 1(1):1–47.
- Flowers, J. H., and Garbin, C. P. 1989. Creativity and perception. In *Handbook of creativity*. Springer. 147–162.
- Frey, J. G. 1936. Miro and the Surrealists. *Parnassus* 8(5):13–15.
- Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style.
- Gervás, P. 2009. Computational Approaches to Storytelling and Creativity.
- Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs]*.
- Jardri, R., and Denève, S. 2013. Computational models of hallucinations. In *The neuroscience of hallucinations*. Springer. 289–313.
- Keeler, M. H. 1970. Klüver’s Mechanisms of Hallucinations as Illustrated by the Paintings of Max Ernst. In M.D, W. K., ed., *Origin and Mechanisms of Hallucinations*. Springer US. 205–208.
- Koch, C. 2015. Do Androids Dream? *Scientific American Mind* 26(6):24–27.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.
- LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.
- Li, C., and Chen, T. 2009. Aesthetic visual quality assessment of paintings. *Selected Topics in Signal Processing, IEEE Journal of* 3(2):236–252.
- Luo, W.; Wang, X.; and Tang, X. 2011. Content-based photo quality assessment. In *International Conference on Computer Vision*, 2206–2213. IEEE.
- Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE CVPR*, 5188–5196. IEEE.
- McCorduck, P. 1991. *Aaron’s code: meta-art, artificial intelligence, and the work of Harold Cohen*. Macmillan.
- Mégevand, P.; Groppe, D. M.; Goldfinger, M. S.; Hwang, S. T.; Kingsley, P. B.; Davidesco, I.; and Mehta, A. D. 2014. Seeing Scenes: Topographic Visual Hallucinations Evoked by Direct Electrical Stimulation of the Parahippocampal Place Area. *The Journal of Neuroscience* 34(16):5399–5405.
- Merabet, L. B.; Maguire, D.; Warde, A.; Alterescu, K.; Stickgold, R.; and Pascual-Leone, A. 2004. Visual hallucinations during prolonged blindfolding in sighted subjects. *Journal of Neuro-Ophthalmology* 24(2):109–113.
- Mocellin, R.; Walterfang, M.; and Velakoulis, D. 2006. Neuropsychiatry of complex visual hallucinations. *Australian and New Zealand journal of psychiatry* 40(9):742–751.
- Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going Deeper into Neural Networks. Google Research Blog.
- Phillips, W. 1948. *Partisan Review, February, 1948, Volume XV, Number 2*, volume XV of *Partisan Review*. New York, NY: Added Enterprises.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2014. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]*.
- Santhouse, A. M.; Howard, R. J.; and Ffytche, D. H. 2000. Visual hallucinatory syndromes and the anatomy of the visual brain. *Brain* 123(10):2055–2064.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE CVPR*, 1–9.
- Teufel, C.; Subramaniam, N.; Dobler, V.; Perez, J.; Finne-mann, J.; Mehta, P. R.; Goodyer, I. M.; and Fletcher, P. C. 2015. Shift toward prior knowledge confers a perceptual advantage in early psychosis and psychosis-prone healthy individuals. *Proceedings of the National Academy of Sciences* 112(43):13401–13406.
- van Gogh, V. 1889. Letter 827: To Willemien van Gogh. Vincent van Gogh: The Letters. Van Gogh Museum.
- Veale, T. 2012. *Exploding the creativity myth: the computational foundations of linguistic creativity*. London ; New York: Continuum International Pub. Group.
- Yao, L.; Suryanarayan, P.; Qiao, M.; Wang, J. Z.; and Li, J. 2012. Oscar: On-site composition and aesthetics feedback through exemplars for photographers. *International Journal of Computer Vision* 96(3):353–383.
- Zeiler, M. D., and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision—ECCV 2014*, 818–833. Springer.
- Zeki, S.; Watson, J. D.; Lueck, C. J.; Friston, K. J.; Kennard, C.; and Frackowiak, R. S. 1991. A direct demonstration of functional specialization in human visual cortex. *The Journal of Neuroscience* 11(3):641–649.

Crossing the horizon: exploring the adjacent possible in a cultural system

P. Gravino¹, B. Monechi¹, V. D. P. Servedio^{2,3}, F. Tria^{3,1}, V. Loreto^{3,1,2}

¹Institute for Scientific Interchange Foundation, Via Alassio 11/c, 10126, Turin, Italy

²Institute for Complex Systems (ISC-CNR), Via dei Taurini 19, 00185 Roma, Italy

³Sapienza University of Rome, Physics Dept., P.le Aldo Moro 2, 00185 Roma, Italy
pietro.gravino@gmail.com

Abstract

It is common opinion that many creative exploits are triggered by serendipity, fortuitous events leading to unintended consequences but this interpretation might simply be due to a poor understanding of the dynamics of creativity. Very little is known, in fact, about how innovations emerge and sample the space of potential novelties. This space is usually referred to as the *adjacent possible*, a concept originally introduced in the study of biological systems to indicate the set of possibilities that are one step away from what actually exists. In this paper we focus on the problem of portraying the adjacent possible space, and of analysing its dynamics, for a particular cultural system. We synthesised the graph emerging from the Internet Movies Database and looked at the static and dynamical properties of this network. We dealt with the subtle mechanism of the adjacent possible by measuring the expansion and the coverage of this elusive space during the global evolution of the system. We introduce the concept of adjacent possibilities at the level of single node to elucidate its nature by looking at the correlations with topological and user annotation metrics. We find that the exploration of the space of possibilities (potentially infinite by definition) shows a saturation size. Furthermore, single node analysis unveiled the importance of the adjacent possible as a useful probe for cultural impact.

The Invisible Horizon from the Shoulders of Giants

In a 1676 letter of Sir Isaac Newton can be found one of his most famous quotes: "if I have seen further, it is by standing on the shoulders of giants". With these words he meant to acknowledge and thank all the scholars that, with their efforts, made his work possible. The quote itself, actually, stems from at least four centuries before and was originally attributed to Bernard of Chartres. All cultural evolution processes strongly depend on the ability to stand on the shoulders of giants. Each new outcome of a cultural system is influenced by prior outcomes, just like in a biological system each offspring is the result of replications, recombinations and/or mutations of its ancestors DNA. The dynamics of evolution and innovation in cultural systems represents a very hot cross-disciplinary topic, which attracted several efforts from the scientific commu-

nity in recent years (Mayer 1998; Elgammal and Saleh 2015; Tria et al. 2014; Jordanous, Allington, and Dueck 2015). In particular, the topic has been tackled from several angles: for example, by trying to understand and quantify the unexpectedness of commercial products (Grace and Maher 2014), by analysing the balance between originality and generativity in the creative cooperative production of online communities (Hill and Monroy-Hernández 2012) or by studying user linguistics behaviours and innovations on the web (Danescu-Niculescu-Mizil et al. 2013). These efforts have been made possible by the unprecedented availability of data tracking influences in the cultural activity typical of the Information Age we live in. Innovation phenomena do not just depend on the shoulders one is standing on. Innovators stand on the edge separating the previous knowledge from what still remains to be discovered. There is a wide horizon of innovations reachable from the verge of what is already known and, after Kauffman (1996), we name it as "adjacent possible". By definition the adjacent possible gets continuously reshaped at every step forward in the unknown. We can describe cultural innovation processes like explorations in the hypothetical network of cultural entities linked by their influences (Wang, Song, and Barabási 2013; Spitz and Horvát 2014; Mauch et al. 2015). Though the way in which these influences are combined to produce novel outcomes is currently under the attention of scientists, very few attempts have been done, to the best of our knowledge, to analyse the way in which cultural networks are explored so that the very notion of *adjacent possible* in cultural systems remains largely unexplored. Several questions arise around this fascinating concept. How creative solutions do explore the adjacent possible frontiers? Do exploration patterns have long time lasting influence in the cultural network? Can this mechanism be improved to foster the insurgence of creative exploits? And, if so, how? In which way the creative exploration path covered in the past does influence future steps? Shedding some light on these questions could strongly improve our understanding of creativity and innovations both at an individual and at a societal level. This paper takes these lines of investigations by focusing on the cultural system behind the cinematographic production. We adopted in particular a Web dataset of cinematographic production to reconstruct the network of influences among motion picture films. This network has been

recently investigated (Wasserman, Zeng, and Amaral 2015; Spitz and Horvát 2014) with the aim to identify the most influential movies. Instead, here we focus on the notion of adjacent possible, both at the individual and collective level, with the aim of investigating its very definition and its structure as well as it gets explored by its community and reshaped over time. Though the adjacent possible remains a very elusive concept, a first portrait of its dynamics will emerge along with an interpretation of its meaning.

The Weaving of Influences in the History of Cinema

The Internet Movie Database (abbreviated IMDb, available at <http://www.imdb.com>) is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews. The information comes from various sources. The IMDb team actively gathers information from studios and filmmakers though the bulk of information is submitted by people in the industry and visitors. Sources of information include, though not limited to, on-screen credits, press kits, official bios, autobiographies, and interviews. Each movie web-page features metadata about awards, box office, releases date, plot keywords, ratings and connection between movies (spoofs, references, quotations, etc). In particular, the connections between pairs of movies are the crucial data we are interested in. To use them as a proxy for the movies influence, we downloaded the dataset of movies and connections, enriched with metadata about awards, ratings, etc., and then applied the following filtering procedure.

From the Raw Dataset to a Movie Influence Network

The IMDb dataset contains several millions of entities, many of which are not movies at all. The filtering procedure explained in the following, partly reproduces the work of (Wasserman, Zeng, and Amaral 2015). The platform contains information about TV shows, game show, news, video-games, music video and short movies and other formats. We reduced our analysis to “normal” movies, labeled in this way by the platform itself. Also, we considered movies with publication date in the period from 1909 to 2005. In this way we avoided the recentism of latest years productions, i.e., the tendency to over-annotate recent movies with respect to their historical importance. This over-annotation in connections would lead to a boosted high degree of some nodes that could bias the structure of the network.

Regarding the connections themselves, we adopted three of the eight types present in the dataset:

spoofs a fun reference to a title is made in a subsequent production;

features extracts from a title appear in another movie; e.g., a movie shows characters attending a cinema screening another movie, or the audio from a program is heard on a TV or a flashback sequence;

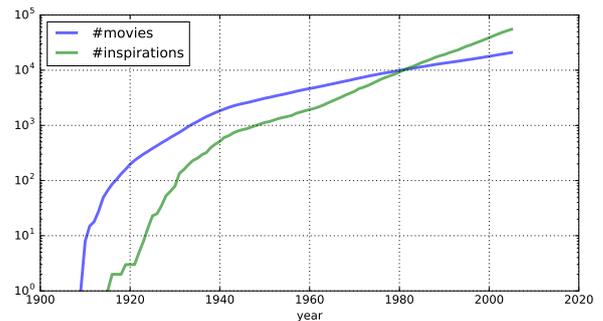


Figure 1: Growth in time of the number of movies and the number of inspiration links between them for the filtered graph.

references a title is referenced or a homage is paid to it in a subsequent movie; this includes recreations of movies scenes and off-screen references (e.g., the background music score)

The other five kinds of connections have been neglected because they are used much less frequently (ca. 10^3 times vs 10^5) and because they are mainly technical connections (e.g., re-edit or alternate language version). From the resulting set of movies and connections we constructed a direct graph (where the direction of links is chronological: influence moves from older movies to newer). Since time resolution is, in the worst cases, 1 year, we adopted this value as time resolution for every movie. We neglected all the interactions between movies of the same year. These interactions are usually unlikely in the dataset, and by doing so we get a tree structure, needed for our analysis. The graph resulting from this filtering has then been reduced to the largest weakly connected component. The final outcome is a graph, that we name the *inspiration graph*, with 20860 movies and 55219 links. The growth in time (by year) of the number of movies and of the number of connections is reported in Fig. 1. The links we are considering represent only the most explicit type of relation that can exist between two movies, without wanted to be exhaustive. Surely, influences are absolutely not limited to the ones reported in our dataset. The assumption we make is that our sample only captures the strongest relations among movies, somehow crucial for the development of a specific movie. In other words, we are assuming that a certain movie could not exist as it is without all the previously created ones with which it shares an inspiration link.

Properties of the inspiration graph

Before proceeding with the operative definition and the analysis of the adjacent possible, we report some basic analysis about the inspiration graph. Since we shall focus on the whole cinematographic system and its productions, it seems natural to consider the production itself, intended as the number of movies produced, as the intrinsic time of the dynamics. In this sense, the temporal unit of our system will be the creation of a movie, instead of the physical time. Fig. 2

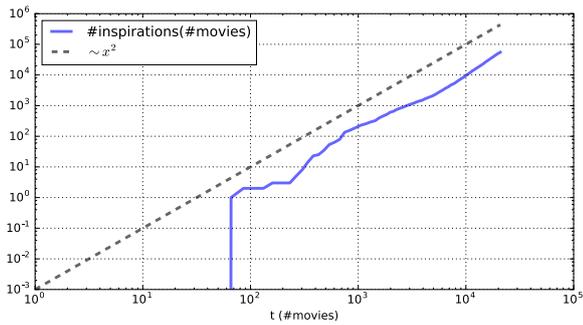


Figure 2: Growth of the number of inspiration links in the intrinsic time of the cultural system (the number of movies). A growing power law $\sim x^2$ is reported as a guide to the eye.

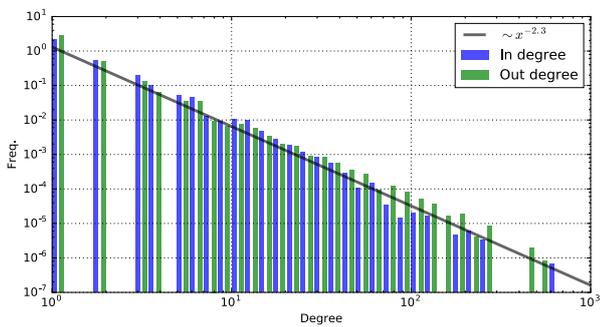


Figure 3: The histograms of the in- and out-degrees for the inspiration graph. The continuous line is the best fit with a power-law function.

reports again the growth of the network in this intrinsic time. The growth of the connections shows a steady power-law like growth (with exponent around 2) except for a few fluctuations, likely to represent the influence of historical, social and economical events (like World War II). An insight about the structure of the network is provided by the distribution of the in-degree (the number of influences received by a title) and of the out-degree (the number of influences coming from a title). Fig. 3 shows that unsurprisingly the distributions of these metrics can be described by power-law distributions. This kind of degree distribution is the signature of scale-free networks, which appear often in the analysis of human behaviour, in annotation process and in other well studied influences network (Spitz and Horvát 2014; Newman 2005; Wang, Song, and Barabási 2013). The distribution proved to be stable also against time resampling (e.g., by taking only a fraction of the story of the system), which means it is a stable feature consequent of the dynamic process we are analysing. The exponent of the power law has been estimated in ~ -2.3 , with no significant difference between out- and in-degree distributions. We complete this preliminary analysis by looking at the distribution of time separations between related movies. The histogram of these distances is presented in Fig. 4, together with two

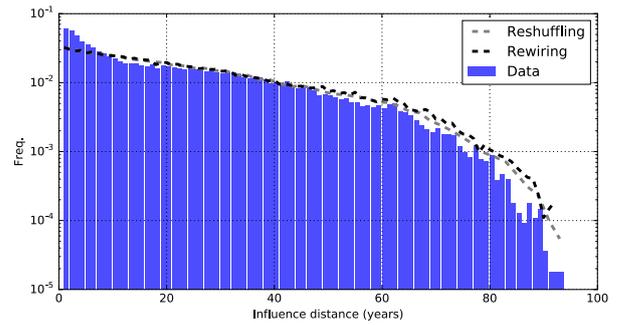


Figure 4: Histogram of the distances in years between related movies. As a reference, we reported the histograms of the distances of two null models: a random reshuffling of all edges and a rewiring preserving the degree distribution.

null models (a rewiring preserving degree distribution and a complete reshuffling of edges) (Albert and Barabási 2002; Wasserman, Zeng, and Amaral 2015). The comparison features a strong bias towards short temporal distances in real connections, which proved to be stable over time. This behaviour of the system highlights the natural tendency of movies to be influenced by those sharing the same cultural moment, like semantically correlated elements clustering in time (Tria et al. 2014).

The Adjacent Possible: Just One Step Away, in the Future

In this section we start by giving an operational definition of adjacent possible. Let us consider a generic graph of cultural productions linked by their influences with a dynamical process on it. At each time step, the graph can be divided in two parts: the known (or the actual) $K(t)$, i.e., the subset of nodes already explored, and the unknown (or the possible) $U(t)$, i.e., the subset of nodes still unseen. The exploration of this graph can only take place through influence links. We can thus define, at each time step, a subset of the unknown set containing all those nodes with all their influencers nodes belonging to the known set. This subset is defined as the “adjacent possible” at time t , $AP(t)$. Alternatively it can be defined as the set of unknown nodes that can be reached with the next step of exploration. An exemplification of the process is reported in Fig. 5. Since, by definition, the adjacent possible lies in the unknown part of the graph, we have no immediate access to it. Also, there is no guarantee that the future evolution of the system will reveal all the nodes belonging to the adjacent possible at any given moment. For sake of clarity let’s consider an example. Suppose we are in 1950 and we look at the network of the whole production so far. In 1950 the adjacent possible of the nodes from 1930 ($AP(1930)$) will be represented by a given number of nodes (for instance the orange nodes in Fig. 5). If we now fast forward in time and land in the year 1980, we observe that the size of $AP(1930)$ will be larger, i.e., the number of orange nodes will have increased. This is a key point. The size of the observed adjacent possible depends on the point in time

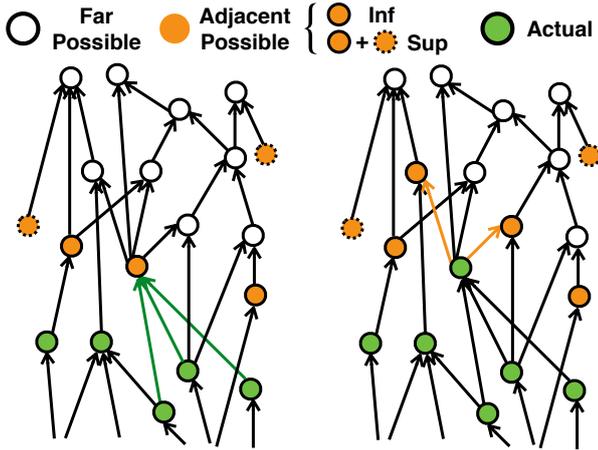


Figure 5: An exemplification of the exploration of the adjacent possible. The known nodes are in green (*actual*). Unknown and unaccessible nodes, i.e. with undiscovered inspirations, are in white (*far possible*). All the productions still unknown but with all the dependencies already discovered are in orange and represent the *adjacent possible*. Nodes with continuous contour have a non-zero in-degree, thus we know their main inspirations and their belonging to some specific adjacent possible, in its strict definition (*Inf*). Nodes with dashed contour do not have a in-degree and their inspiration are not known (they could be completely original or those inspirations could be simply not reported in the dataset or come from external media like books, news, etc). Thus, these nodes can be considered always in the loose definition of the adjacent possible (*Sup*), until they happen to be discovered. On the left, the graph before a production step. The new production is chosen among those in the adjacent possible. After the step, on the right, a new node is now known, and it has unlocked new nodes that are now part of the new adjacent possible.

from which we retrace the whole history. Presumably in 20 years time there will be new movies produced that will be adjacent adjacent to those of 1930. This means that, based on what we know and what we can measure, the adjacent possible could be an infinite set and it is only the finiteness of our sample that makes it finite. The best we can do is to measure the subset of adjacent possible observed at any given time. In practice what we can observe depends on two times: the time t at which we define the adjacent possible and the time t' ($t' > t$) from which we retrace the history. We can thus define the *observed adjacent possible* as:

$$\Gamma(t', t) = AP(t) \cap K(t') \quad (1)$$

where $K(t')$ is the set of known nodes at time t' . Though this set does not allow for a direct measure of $AP(t)$ it is very useful to provide us with valuable insights on how the exploration of $AP(t)$ takes place. Let us now apply this definition to our system. In our dataset we do not have the information about each intrinsic time step (i.e., each time a new movie comes out) since our time resolution is one year.

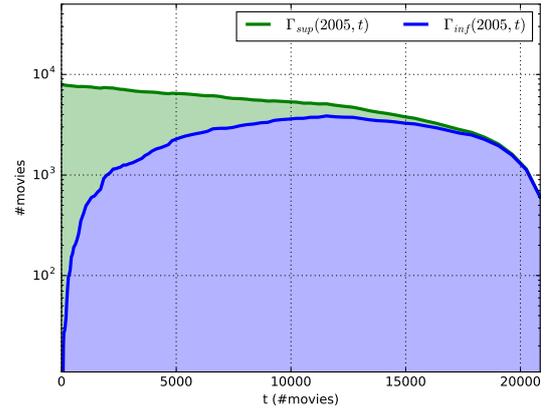


Figure 6: Measure of the superior ($\Gamma_{\text{sup}}(2005, t)$, in green) and inferior ($\Gamma_{\text{inf}}(2005, t)$, in blue) estimates for the observed adjacent possible of the inspiration network vs. the intrinsic time t of the system, i.e., the number of movies produced.

Still, we can define the state of knowledge of the network once a year, and consequently, we can estimate $\Gamma(t', t)$.

Before proceeding we should consider another element. In order for a node to be in the adjacent possible of other nodes, it must receive at least one influence, which means that the in-degree must be larger than 0. However, since the in-degree is distributed according to a power law, $k_{\text{in}} = 0$ is not only possible but is the most likely value. Actually, we cannot consider all these nodes as not having any influences at all. It is more likely that those influences have not been tracked yet or they come from sources external to our network (e.g., a book, a song, etc.). In order to overcome this problem, we define two metrics for the adjacent possible, depending on how we choose to treat nodes with $k_{\text{in}} = 0$. We can consider them as potentially uninfluenced, and then always in the adjacent possible until they happen to become part of the K set or we can simply neglect them. In one case, we are overestimating the size of the observed adjacent possible we can access, in the other case we are underestimating it. These ideas are explained in Fig. 5. We named these two metrics Γ_{sup} and Γ_{inf} and we measured both for each yearly state of knowledge of the network. Results are shown in Fig. 6. The measure gives us a general information about the typical size of Γ , which lies between 10^3 and 10^4 for the whole evolution. The measure in the final part loses reliability due to size effects, but still we can suppose that the size of Γ does not diverge with the size of the system. Let us now study directly the evolution of the coverage of the observed adjacent possible at a given time t . With our data, the best estimation that can be given of how the adjacent possible is going to be known during the exploration is to measure the evolution in time t of $\Gamma(2005, t') \cap K(t)$, i.e., the number of movies of the observed adjacent possible that are actually realized at each time t . In other words, with our data the best estimation for the adjacent possible of a given year t'

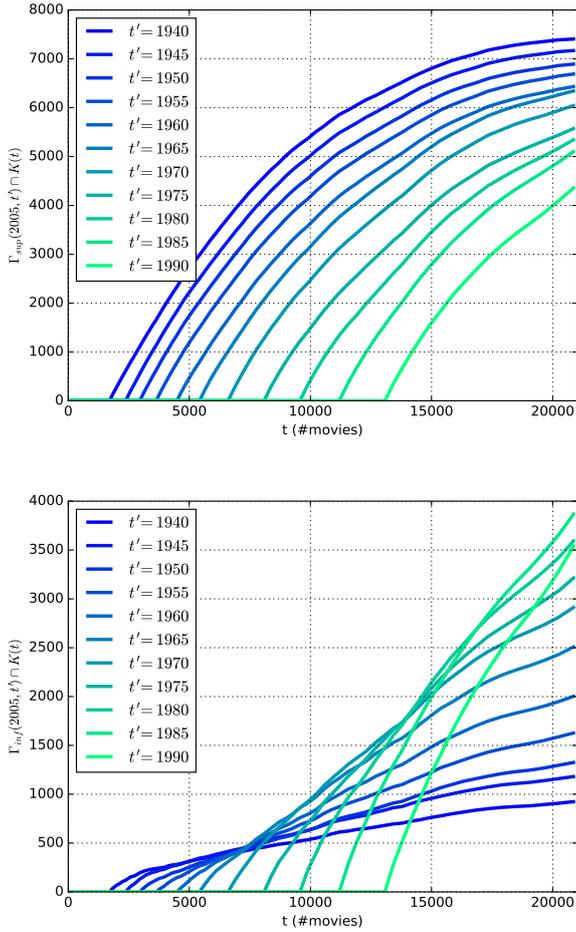


Figure 7: The evolution of the coverage of the observed adjacent possible Γ_{sup} (upper figure) and Γ_{inf} (lower figure). Different curves correspond to different values of t' .

is the *observed adjacent possible* calculated using the whole timespan, i.e. updated to 2005. This set, according to Eq. 1, can be indicated with $\Gamma(2005, t')$. What we want to measure is how many movies of this set have been actualized in time t (where obviously $t > t'$). The results, for both metrics (Γ_{inf} and Γ_{sup}) of the observed adjacent possible, are reported in Fig. 7. Both measures, in particular Γ_{inf} , seem to show a tendency to saturation if a sufficient elapse of time has passed. To quantitatively account for this effect we fitted both types of curves with a function of the kind $y = a(1 - e^{-x/b})$, describing and exponential asymptotic relaxation towards a constant value defined by a . Fit results are not reported for the sake of brevity. Instead, we show in Fig. 8 how all the curves in Fig. 7 collapse when shifted and rescaled according to the transformations $x \rightarrow x/b$ and $y \rightarrow y/a$. We observe a convincing collapse for Γ_{sup} curves while Γ_{inf} curves feature some fluctuations around the master curve. To investigate such fluctuations, we fitted the curves of Γ_{sup} and Γ_{inf} for every time t' . Figure 9 shows the obtained fit-

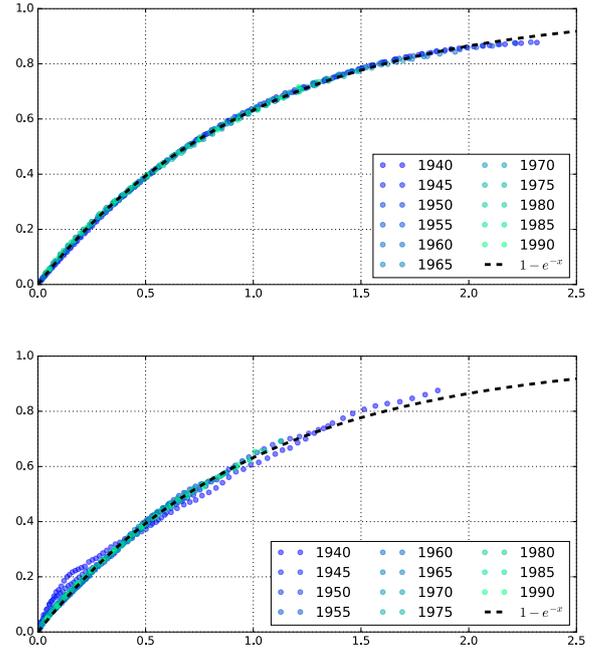


Figure 8: Collapse of the observed adjacent possible coverage curves of Fig. 7 when shifted and rescaled according the exponential fit parameters. Γ_{sup} curves are reported in the upper panel, while Γ_{inf} in the lower.

ting parameters a and b as a function of the intrinsic time (i.e., number of movies). It can be seen that for Γ_{inf} the largest fluctuations correspond to curves related to the first years of the dynamics. These measures are important because they tell us that even if the adjacent possible could be infinite, the *observed adjacent possible* of a given state of the inspirations network is covered in time in a way that suggests its boundedness. Indeed, and it was not obvious a priori, its discovered size seems to converge. Moreover, we have a quantitative account of time scales to reliably observe the convergence of the observed adjacent possible, or at least to estimate its size. The upper part of Fig. 9 shows the evolution with the growth of the system of the b parameter which is the time-scale of the exponential function fitting the coverage curves of Fig. 7. The parameter b can also be interpreted as the order of magnitude of the intrinsic time one should wait to have a reliable observation. Considering Γ_{sup} we see that the behaviour of both a and b as functions of the intrinsic time changes around a time $\approx 10^4$. Hence, considering this change as an effect of the finite size of the system, we are able to estimate correctly the size of Γ_{sup} for at least half of our dataset (i.e., every movie produced before $t \approx 10^4$). The evolution of b relative to Γ_{inf} tells us a different story. There is a clear peak (apparently limited only by the size of the dataset) representing a divergence of the timescale. Accordingly, since to accurately measure size we need data spanning more than a timescale, we can then conclude that size measures in the time frame of the di-

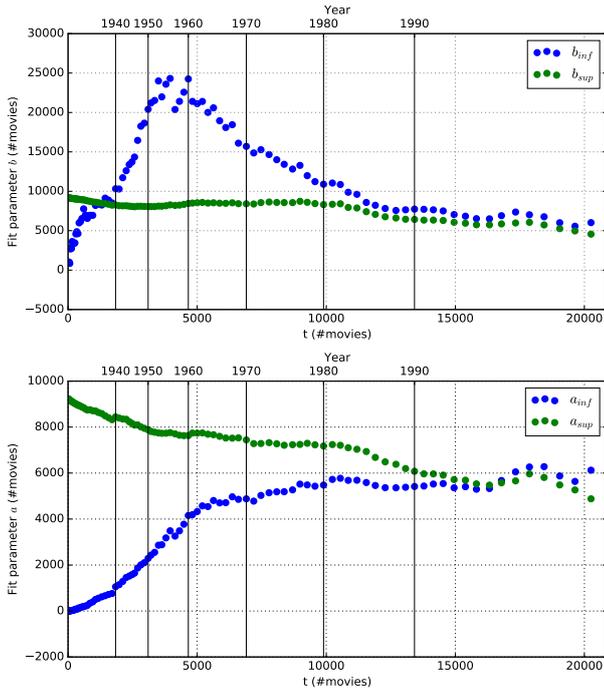


Figure 9: The evolution of exponential fitting parameters a and b . Each dot is a year for which we calculated the observed adjacent possible, measured the coverage curve (like in Fig. 7) and estimated the exponential fit parameters a and b . The curves represent the evolution of the coverage functions of the observed adjacent possible. In the upper figure, the sequence of values for the parameter b , representing the scale of time, both for Γ_{sup} (green) and for Γ_{inf} (blue); in the lower panel, the analogous for the parameter a , measured in number of movies.

vergence are less reliable. Comparing this with Fig. 1 and Fig. 2 we notice that such period is characterised by a strong change in the dynamics, which could have led the system to this instability. Looking at Fig. 9, we can see that this effect disappears for $t \sim 10^3$ (approximately half of the story covered by the dataset), where the curve lies smoothly in the same range of the b_{sup} curve. Looking instead at the size parameter a we observe, for Γ_{sup} , a decreasing curve with some discontinuity in the slope around 10^4 . The asymptotic size of the observed adjacent possible seem not to be divergent with the size of the system and, thus, measurable (roughly estimate around $\sim 7 \cdot 10^3$). For the Γ_{inf} scale parameter we observe a different behaviour. A rapid growth until the peak of the unstable zone ($\sim 5 \cdot 10^3$) followed by a more or less stable plateau slightly under $\sim 6 \cdot 10^3$. In the last part of the evolution, the two parameters estimating the asymptotic size of the observed adjacent possible basically collapse, suggesting an high reliability of the measure.

Possible Meanings for the Adjacent Possible

The procedures implemented so far have, amongst the other purposes, the aim to prove the measurability of the observed

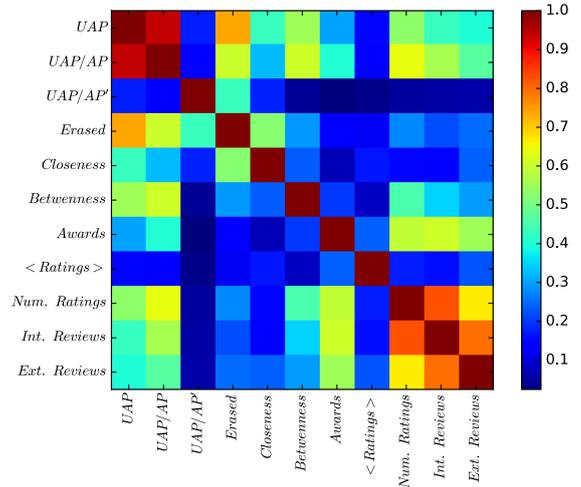


Figure 10: The matrix of the Pearson's coefficient between the adopted metrics.

adjacent possible and to give a quantitative insight about it. A qualitative understanding is also needed. To look more in detail at the possible meanings of the entity we defined at the global systemic scale, we can give also a microscopic definition of the adjacent possible of single nodes. In particular, if we consider Fig. 5, we can notice how the passage of a node from the adjacent possible to the known “unlocks” other nodes that, after the step, have become accessible and thus now belong to the adjacent possible. So, we can define for each node a metric depending on the unlocked adjacent possible (UAP in the following, referring to the observed adjacent possible). This metric, normalized in different ways, has to be compared with others, as described in the following. The comparison has been performed by calculating the Pearson's coefficient. The matrix of the results is reported in Fig. 10. The metrics evaluated are reported in the following, and are always relative to a generic node n .

UAP the number of nodes made available by the production of n ; the number of nodes which were missing only n as reference to be in the adjacent possible.

UAP/AP_{sup} the UAP metric normalised by the superior limit to the size of the adjacent possible observed for the year of n . AP_{sup} stands for $\Gamma_{sup}(2005, t(n))$.

UAP/AP_{inf} the UAP metric normalised by the superior limit to the size of the adjacent possible observed for the year of n . AP_{inf} stands for $\Gamma_{inf}(2005, t(n))$.

Erased the number of nodes that would be inaccessible if the node n would never be discovered. This number has been estimated with the following algorithm.

1. we remove the node n ;
2. we considered each node m amongst those influenced by n . We will assume that if n would not exist each node m would risk not to be discovered, depending on the importance of the influence between n and the specific m ;

3. to estimate this importance, we consider all influences received by the specific m . We weight each influence received by m as the inverse of the in-degree of the node m . If the weight of the influence between n and m crosses a given threshold (30%), the influenced node m is removed and all its descendants will be considered for removal; i.e. if a node m has only two influences including n then the importance of the influence of n can be roughly estimated to be around $\sim 50\%$; thus, since this value exceeds the threshold, m will be removed and the movies influenced by m will have to be checked;
4. all nodes in the list of those to be considered for removal are analysed chronologically with the same rules, removing them if the sum of the weights of the deleted influences passes the threshold and adding their influenced nodes to the list, in case of removal.

This metric is an adapted version of the vitality metrics from (Brandes and Erlebach 2005).

Closeness Closeness centrality (Freeman 1978) is the reciprocal of the sum of the shortest path distances from n to all $N - 1$ other nodes. Since the sum of distances depends on the number of nodes in the graph, closeness is normalised by the sum of minimum possible distances $N - 1$.

$$C(n) = \frac{N - 1}{\sum_m d(m, n)} \quad (2)$$

Betweenness The betweenness centrality (Brandes 2008; Brandes and Erlebach 2005) of a node n is the sum of the fraction of all-pairs shortest paths that pass through n :

$$c_B(n) = \sum_{m, m' \in V} \frac{\sigma(m, m' | n)}{\sigma(m, m')} \quad (3)$$

where V is the set of nodes, $\sigma(m, m')$ is the number of shortest (m, m') -paths, and $\sigma(m, m' | v)$ is the number of those paths passing through $n \neq m, m'$.

Awards The number of awards and nominations obtained by n as reported on the IMDb platform.

⟨Ratings⟩ The average vote (from 1 to 10) for the movie n given by IMDb platform registered users.

Num. Ratings The number of votes for the movie n given by IMDb platform registered users.

Int. Ratings The number of reviews for the movie n submitted by IMDb platform registered users.

Ext. Ratings The number of links to external website reviews (usually from major print or online media organisations). Links are submitted on the IMDb platform by film reviewer or editor or a movie site team.

Let us now discuss the correlation matrix reported in Fig. 10. Our main matrix, UAP strongly correlates with UAP/AP_{sup} but very poorly with UAP/AP_{inf} . The latter shows a weak degree of correlation with the *Erased* metric. This metric has been introduced to prove that UAP , despite its very local definition, can have long temporal range consequences. In fact, all the first three metrics show a strong

correlation with it, meaning that the influence of a node with high UAP metric can reach high temporal distances, more or less directly. Closeness and betweenness centrality measures correlate fairly with the UAP and UAP/AP_{sup} . This is an insight of their value in the identification of nodes important from a topological point of view, connecting different communities or standing in the core of the network. All the other metric correlations proves the cultural value of nodes with high UAP and of UAP/AP_{sup} metric. It is worth to note the weakness of the correlation with $\langle Ratings \rangle$. This seems to suggest that the cultural value we are observing deals more with the interest gathered than with the appreciation (according to interest and appreciation information given by IMDb users).

Conclusions and perspective: the adjacent possible of this paper

In this paper we analysed the adjacent possible, the space in which creative efforts can move a step over the frontiers of what is known. We synthesised a network of influences between the entities of a cultural system. In particular we dealt with the cinematographic production system by leveraging the data extracted from the IMDb platform. With a suitable filtering procedure we sketched a graph of the most important influences and studied its structure and dynamical properties. In particular, we observed that despite the fact that the system showed an unstable growth rate, it resulted in a scale-free network of influences among movies. Moreover, these influences were found to be preferentially attached over short time distances (as inferred by comparing with null models). We then defined the observable projection of the adjacent possible according to the temporal resolution of one year. For each year, the observed adjacent possible was considered as the set of movies not yet produced whose inspirations lay all in the past. We had to define two kinds of observed adjacent possible in order to take into account nodes without annotated influences, the upper and lower bounds of the adjacent possible. We measured the adjacent possible for every year, within the dataset limits. Then we tracked how the adjacent possible of each year was covered by what was already known at previous times. This evolution led us to fit the coverage curves, and to estimate the typical time scale and the asymptotic limit for the size of the observed adjacent possible. Both numbers are, in the majority of cases, substantially smaller than the size and characteristic times of the whole network. This seems to suggest the existence of a saturation in the size of the observed adjacent possible at any given time that will be eventually explored. In other words, this result indicates that, even though the adjacent possible of a given state of the network is potentially unbounded, only a finite part of it is likely to be visited, and the size of this part can be estimated in a finite amount of time (e.g., with datasets of other cultural systems with a longer timespan or through computer simulations). This result is somehow surprising given our absolute, though natural, ignorance about the structure of the adjacent possible. Again, it is worth remarking that our conclusions apply to those parts of the network space one can

observe, i.e., to the way in which that space was explored in history. In the last part of the paper we re-elaborated the definition of a suitable metrics for nodes, to be compared with other metrics, already known in literature or used by the IMDb dataset itself, related to the influence or the popularity of a movie. The metric we propose consists in the size of the unlocked adjacent possible (*UAP*). After a new node n is produced, the *UAP* is the number of nodes that were unreachable before and now are made available for production as a result of all known influences, including that of n itself. This metric, despite its local definition, was shown to be strongly correlated with a metric calculated on large temporal distances. Also, comparisons with standard topological metrics showed that high *UAP* values correspond to crucial nodes in the structure of the network. Finally, we also confirmed the cultural importance of the *UAP* as it correlates with the IMDb metrics, which are interesting for users. All these correlations confirmed the strategical importance of the adjacent possible concept even at the single node level. Thus, the study and understanding of its dynamic could be strategically fundamental to get a deeper comprehension of cultural system dynamics and evolution. The obvious problem for this is the time limit of the available statistics. This could be easily overcome by creating a model faithful enough to reproduce not only the statistical markers of the influence network but also the pattern of exploration of the adjacent possible. Given the peculiar characteristics of the network of influence this seems not to be an easy task, because the right balance between short time biases and preferential attachment (leading to a scale-free distribution) could be conflicting. Even when correctly balanced, there is no guarantee that the model would reproduce the correct adjacent possible exploration pattern. However, in case of success, such a model could confirm (or discard) our findings and could provide several answers about how creativity works and, maybe, can be improved at an individual and at a societal level. In fact, our metrics give us a new instrument to evaluate the value and the impact of creative productions. Also, this work can be considered as a first step toward a possible optimisation strategy for the exploration of the unknown. In fact, a deeper understanding of the *adjacent possible* exploration patterns could help to recreate opportune condition for a faster insurgence and spreading of creative solutions. We could understand if it is possible to efficiently drive innovation toward a given direction, and how, and this could completely transform, for example, our scientific research funding policies and our artistic or technological evolution cycles. It is likely that a good *theory of the adjacent possible*, capable of such wonders, lies still far from *our actual adjacent possible*, but we hope our work could move the boundaries a bit toward that direction.

Acknowledgments

We acknowledge support from the KREYON project funded by the Templeton Foundation under contract n. 51663. VDPS acknowledges the EU FP7 Grant 611272 (project GROWTHCOM) and the CNR PNR Project “CRISIS Lab” for financial support.

References

- Albert, R., and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74(1):47.
- Brandes, U., and Erlebach, T. 2005. *Network analysis: methodological foundations*, volume 3418. Springer Science & Business Media.
- Brandes, U. 2008. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks* 30(2):136–145.
- Danescu-Niculescu-Mizil, C.; West, R.; Jurafsky, D.; Leskovec, J.; and Potts, C. 2013. No country for old members: User lifecycle and linguistic change in online communities. *WWW'13*.
- Elgammal, A., and Saleh, B. 2015. Quantifying creativity in art networks. *arXiv preprint arXiv:1506.00711*.
- Freeman, L. C. 1978. Centrality in social networks conceptual clarification. *Social networks* 1(3):215–239.
- Grace, K., and Maher, M. L. 2014. What to expect when you're expecting: the role of unexpectedness in computationally evaluating creativity. In *Proceedings of the 4th International Conference on Computational Creativity, to appear*.
- Hill, B. M., and Monroy-Hernández, A. 2012. The remixing dilemma: The trade-off between generativity and originality. *American Behavioral Scientist* 57(5).
- Jordanous, A.; Allington, D.; and Dueck, B. 2015. Measuring cultural value using social network analysis: a case study on valuing electronic musicians. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 110.
- Kauffman, S. A. 1996. Investigations on the character of autonomous agents and the worlds they mutually create. In *Investigations*. Santa Fe Institute.
- Mauch, M.; MacCallum, R. M.; Levy, M.; and Leroi, A. M. 2015. The evolution of popular music: Usa 1960–2010. *Royal Society open science* 2(5):150081.
- Mayer, R. E. 1998. Fifty years of creativity research. In Sternberg, R. J., ed., *Handbook of Creativity*. Cambridge: Cambridge University Press. 449–460.
- Newman, M. E. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics* 46(5):323–351.
- Spitz, A., and Horvát, E.-Á. 2014. Measuring long-term impact based on network centrality: Unraveling cinematic citations. *PloS one* 9(10):e108857.
- Tria, F.; Loreto, V.; Servedio, V. D. P.; and Strogatz, S. H. 2014. The dynamics of correlated novelties. *Scientific Reports* 4:5890.
- Wang, D.; Song, C.; and Barabási, A.-L. 2013. Quantifying long-term scientific impact. *Science* 342(6154):127–132.
- Wasserman, M.; Zeng, X. H. T.; and Amaral, L. A. N. 2015. Cross-evaluation of metrics to estimate the significance of creative works. *Proceedings of the National Academy of Sciences* 112(5):1281–1286.

Computational Creativity Conceptualisation Grounded on ICCC Papers

Senja Pollak¹, Biljana Mileva Boshkoska^{1,3}, Dragana Miljkovic¹, Geraint A. Wiggins², Nada Lavrač^{1,4}

¹Dept. of Knowledge Technology, Jožef Stefan Institute, Ljubljana, Slovenia

²Computational Creativity Lab, Queen Mary University of London, London E1 4NS, UK

³ Faculty of Information Studies, Novo Mesto, Slovenia

⁴ University of Nova Gorica, Nova Gorica, Slovenia

{senja.pollak,biljana.mileva,dragana.miljkovic,nada.lavrac}@ijs.si; geraint.wiggins@qmul.ac.uk

Abstract

In information science, it is considered that domain conceptualization can be realized by (one or several) ontologies. This paper presents a method of semi-automated domain conceptualization, where the domain of interest is Computational Creativity (CC). Grounded on papers, which were published in six consecutive years since 2010 in the Proceedings of International Conferences on Computational Creativity (ICCC), this paper proposes a tentative conceptualization of the CC domain. Some additional properties of the CC domain are studied, analysed by means of fully mechanical or semi-automated information extraction and dependency analysis techniques. This approach affords an interesting opportunity for automated historiography of a research field.

Introduction

As a sub-field of Artificial Intelligence research, *Computational Creativity* (CC) is concerned with engineering software that exhibits behaviours which would reasonably be deemed creative (Wiggins, 2006; Colton and Wiggins, 2012). A part of CC research addresses *Concept Creation Technology*, concerned with engineering software that exhibits creative conceptualization behaviour.

For a given domain, whose conceptual space (Boden, 2004) is closed, pre-defined and yet unexplored, it is interesting to study computational means for automated (or semi-automated) domain conceptualization. In the current research, we use the term *conceptualization* in alignment with its standard use in information science: a *conceptualization* is defined as “an abstract (simplified) view of some selected part of the world, containing the objects, concepts, and other entities that are presumed of interest for some particular purpose and the relationships between them.” Domain conceptualization is, in information science, frequently realized by manually defining (one or several) ontologies formally describing the domain of interest (Gruber, 1993; Smith, 2003).

Manual construction of ontologies represents a significant investment of human resources when used for modelling a new domain. Therefore, methods for (semi-)automated extraction of domain knowledge from unstructured texts were developed, including automated taxonomy construction described by Velardi, Faralli, and Navigli (2013).

While an ontology is a “formal, explicit specification of a shared conceptualization” (Gruber, 1993), represented as a set of domain concepts and the relationships between

them, a so-called *topic ontology* is a set of domain topics or concepts—formed of related documents—represented by the most characteristic topic keywords and related by the *subconcept-of* relationship (Fortuna, Grobelnik, and Mladenić, 2007). The task addressed in this paper is semi-automated construction of a topic ontology from documents in the area of computational creativity.

CC domain conceptualization has not been substantially addressed in the CC literature. Jordanous and Keller (2012) used automated natural language processing methods and a statistical measure of association to identify words related to creativity (in general, not specifically CC). They clustered the words into semantically-related groups by using a lexical similarity measure, resulting in an ontology of creativity¹). Others presented extraction of creativity concepts related to, e.g., sub-fields of creativity (Agres et al., 2015) and creativity evaluation (van der Velde et al., 2015).

This approach to the study of collections of documents opens the prospect of an automated historiography of the field of computational creativity, an idea which constitutes a satisfyingly recursive application of the research outputs of that area of interest; a similar exercise has been undertaken within the Association for Computational Linguistics (Anderson, McFarland, and Jurafsky, 2012). Here, we illustrate, with real computational examples, the kinds of analysis (e.g., diachronic comparisons of conceptualisation) that would be used for such studies.

The current paper presents a method of semi-automated domain conceptualization, where the domain of interest is Computational Creativity (CC). The paper proposes a conceptualization of CC grounded in papers published in the Proceedings of International Conferences on Computational Creativity (ICCC). Some additional properties of the CC domain are studied, obtained by means of information extraction and dependency analysis techniques. The experimental data is presented, followed by the results of CC domain conceptualization and time dependency analysis.

The Computational Creativity Domain

This section describes the data used in the experiments, together with initial domain understanding achieved through automated terminology extraction.

¹<http://purl.org/creativity/ontology>

ICCC proceedings data The documents were taken from the ICCC proceedings published between 2010 and 2015, inclusive. In total, we considered 247 articles from all six proceedings, containing the following numbers of articles: 2010: 43, '11: 30, '12: 44, '13: 40, '14: 49, '15: 41.

The papers, in PDF, were first converted to plain text. We omitted the references, but added the information about the conference year (however, for time dependency analysis, presented in the last section of this paper, the version of the corpus including references was used).

Automated CC terminology extraction The goal of terminology extraction is to automatically extract relevant terms for a given domain, represented by a given corpus. We used terminology extraction method LUIZ-CF (Pollak et al., 2012), a modified version of the LUIZ term recognition tool (Vintar, 2010). LUIZ-CF is implemented as a workflow in the ClowdFlows environment.

Term extraction consists of two steps: extracting the noun phrase candidates based on morphosyntactic patterns; followed by weighting and ranking of the candidates based on their termhood value, for single word and multi-word terms. The termhood value is computed based on comparison of relative frequencies of lemmas of a term in the domain corpus (here, the ICCC proceedings) compared to a reference corpus: for English, the frequencies of the British National Corpus are used. The extracted terms are ranked by termhood value on a scale between 1 and 0. In addition to default stop words, we eliminated also the names of ICCC PC members, leading to the exclusion of some of the paper authors from the term list. The top ranked candidates are listed in Table 1, followed by a list of top ranked multi-word terms from the same term list given in Table 2. The term extraction method is explained in details in (Pollak et al., 2012). The term extraction workflow is available in ClowdFlows². The extracted terms may be considered as an initial computational creativity vocabulary for building a dictionary of computational creativity, which is planned in future work.

CC domain conceptualization with OntoGen

A tool named OntoGen³ (Fortuna, Grobelnik, and Mladenić, 2007) was used to build a topic ontology for CC domain conceptualization. OntoGen is a semi-automatic and data-driven ontology editor. Semi-automatic means that the system is an interactive tool that aids the user during the topic ontology construction process. Data-driven means that most of the aid provided by the system is based on the underlying text data (document corpus) provided by the user. The system combines text mining techniques with an efficient user interface and was already validated in several applications, including its application to inductive logic programming conceptualization Lavrač et al. (2010).

OntoGen accepts texts in various formats. We chose the named line document format, where each line represents one document, starting with the document ID and the conference edition (2010, 2011, 2012, 2013, 2014 or 2015) as a category. OntoGen performs basic lemmatization and stop word removal (and accepts additional user-defined stop word lists)

²<http://clowdflows.org/workflow/7219/>

³<http://ontogen.ijs.si/>

Table 1: Top 15 terms from the ICCC corpus.

Score	Term
1.000000	[creativity]
1.000000	[computational creativity]
0.862623	[system]
0.247012	[creative system]
0.182190	[process]
0.174810	[model]
0.141525	[image]
0.126607	[concept]
0.102306	[creative process]
0.101973	[word]
0.099952	[evaluation]
0.099844	[conceptual space]
0.081564	[domain]
0.080851	[generation]
0.073521	[story]

Table 2: Top 15 multi-word terms from the ICCC corpus.

Score	Term
1.000000	[computational creativity]
0.247012	[creative system]
0.102306	[creative process]
0.099844	[conceptual space]
0.030894	[computational model]
0.021593	[computational system]
0.018712	[fitness function]
0.018064	[jaguar knight]
0.012638	[genetic algorithm]
0.012078	[human creativity]
0.011300	[poetry generation]
0.011171	[story generation]
0.010011	[neural network]
0.009657	[creative domain]
0.009299	[transformational creativity]

and constructs Bag-of-Words (BoW) vector representations of documents, weighted by the TF-IDF weights (Salton and Buckley, 1988), where TF-IDF stands for Term Frequency-Inverse Document Frequency.

OntoGen is illustrated in Fig. 1. The “unsupervised concept suggestion” functionality is a central part of the system: for a given concept (e.g., the central concept “computational creativity” represented by all the documents of the ICCC domain), a list of sub-concepts is suggested by k -means clustering (Jain, Murty, and Flynn, 1999) and Latent Semantic Indexing (Deerwester et al., 1990) techniques. If the user does not want to affect the conceptualization outcome, only parameter k needs to be chosen to determine the number of concepts, i.e., the number of categories in which the documents will be clustered. “Keywords” (automatically assigned names of clusters) are the words that are the most descriptive for the content of the concepts instances (articles), i.e., words with the highest weights in the document centroid vectors (Fortuna, Mladenić, and Grobelnik, 2006). The main OntoGen window represents the ontology visuali-

sation in which each concept is represented by top three keywords unless manually edited, while the Concept hierarchy window (on the upper left corner) offers a quick overview of all the concepts with their position in the concept hierarchy that can be also directly manipulated.

An alternative view is over the Concepts' documents, where documents of each concept (document cluster) are visualized. Fig. 2 shows documents of the selected concept, i.e. the one represented by keywords "music, chord, improvisation", which could be reasonably be called "Musical creativity". In the similarity graph (at the bottom of the figure), the red dots represent documents belonging to the selected concept, while blue dots the documents not belonging to the concept. The graph inspection functionality can be used for selecting documents to be manually inspected and eventually removed or added to the concept. SVM keywords (see left bottom corner of the figure) are composed from words most distinctive for the selected concept concerning its sibling concepts in the hierarchy (obviously not available for the root concept). SVM keywords are explained more detailed in Fortuna, Grobelnik, and Mladeni (2006).

An important additional functionality of OntoGen is a supervised method for adding concepts. It is based on SVM active learning method Fortuna, Grobelnik, and Mladeni (2006). The user supervision is provided first by a query describing the concept that the user has in mind and followed by a sequence of questions whether a particular instance (document) belongs to the concept and the user can select Yes or No. The questions are chosen from the instances on the border between being relevant to the query or not and are therefore most informative to the system. The system refines the suggested concept after each reply from the user and the user can decide when to stop the process based on how satisfied he is with the suggestions. After the concept is constructed it is added to the ontology as a sub-concept of the selected concept.

Automated CC conceptualization First we performed k -means clustering for $k = 5$. In the topic ontology (Fig. 1), OntoGen uses the first three automatically extracted keywords as concept/topic descriptors.

By inspecting the keywords, we manually named the concepts of the automatically generated topic ontology as follows: "Musical creativity", "Visual creativity", "Linguistic creativity", "Creativity in Games" and "Conceptual creativity" (see Table 3). While some of the categories are quite uniform regarding the keywords (e.g., "music, chord, improvisation, melodies, harmonize, composition, accompaniment, pitch, emotions, beat" for the concept category that we named "Musical creativity"), other categories are more noisy, e.g., "image, story, actions, painting, character, agents, narrative, artists, robot, darci" do not denote a uniform category. We decided to name this category "Visual creativity", but it obviously contains documents from other topics as well, such as narratives generation.

From a set of ten automatically generated keywords⁴ characterizing each of the five document clusters (Table 3), we manually selected three keywords believed best to describe the cluster of papers belonging to each concept (in italics).

⁴Words with the highest weights in the document centroid vectors (Fortuna, Mladenič, and Grobelnik, 2006)

These keywords were added to the concept labels of Fig. 3.

In the next section, aiming at a more elaborated version of the CC ontology (see Fig. 4), we use the concept moving facility of OntoGen, by which we moved e.g., the concept "Narrative" from "Visual" to "Lexical", together with other techniques for manipulating the initial ontology.

Semi-automated CC domain conceptualization This section describes improved CC conceptualization, created by manipulating the initial ontology using different OntoGen functionalities. The main concepts were further divided and when forming meaningful concepts, the categories were added as sub-concepts (see e.g. the sub-concepts of Linguistic creativity in Fig. 4). As already mentioned, some (sub-)concepts were moved, e.g., the "narrative" category was moved from Visual to Linguistic creativity. Some concepts were added by query and active learning. On the first level, this is the case for the category Evaluation, which was a recurrent topic in other categories and we used query and active learning to form an independent category. We also used it, e.g., for creating the category "Recipes" as a sub-concept of lexical creativity. We also used the OntoGen function to (de)select the documents being categorized to one concept category.

Fig. 2 shows the documents belonging to a category. It is very interesting to inspect some outliers (documents similar to documents in the category not being classified to this category). In the concept document graph, we identified some of the outliers, represented by blue dots in the similarity line of red dots. An example is article 2014_44, entitled "Arts, News, and Poetry The Art of Framing", by Gross, Toivanen, Laane and Toivonen. This paper was not classified in the Linguistic creativity category, but was identified as similar to the documents of that category. The article indeed refers to linguistic creativity (poetry) but also to generated pictures as well as pictures painted by an artist. We manually added this document also to the category of linguistic creativity.

The result of this experiment is shown in Fig. 4. On the first level we distinguish between Musical, Visual, Linguistic creativity, Games and creativity, Conceptual creativity as well as newly created category of Evaluation. On lower levels, we added e.g. Narratives, Poetry, Recipes and Lexical creativity for Linguistic creativity, where the latter comprises e.g., humour, neologisms, etc.

Each concept is represented by descriptive Keywords (see keywords for six first level concepts in Table 4) from which we selected three keywords (in italics) to represent the concept in the visual ontology (Fig. 4). The ontology can be considered as a draft to be collaboratively improved by the CC community. Further, since the concepts are grounded in the documents, the top ranked documents might be considered as interesting reading for newcomers to CC. The bibliography can be created for concepts of different levels (as an example see three articles per selected topic):

Narratives:

- A System for Evaluating Novelty in Computer Generated Narratives (Pérez y Pérez et al., 2011)
- Kill the Dragon and Rescue the Princess: Designing a Plan-based Multiagent Story Generator (Laclaustra et al., 2014)

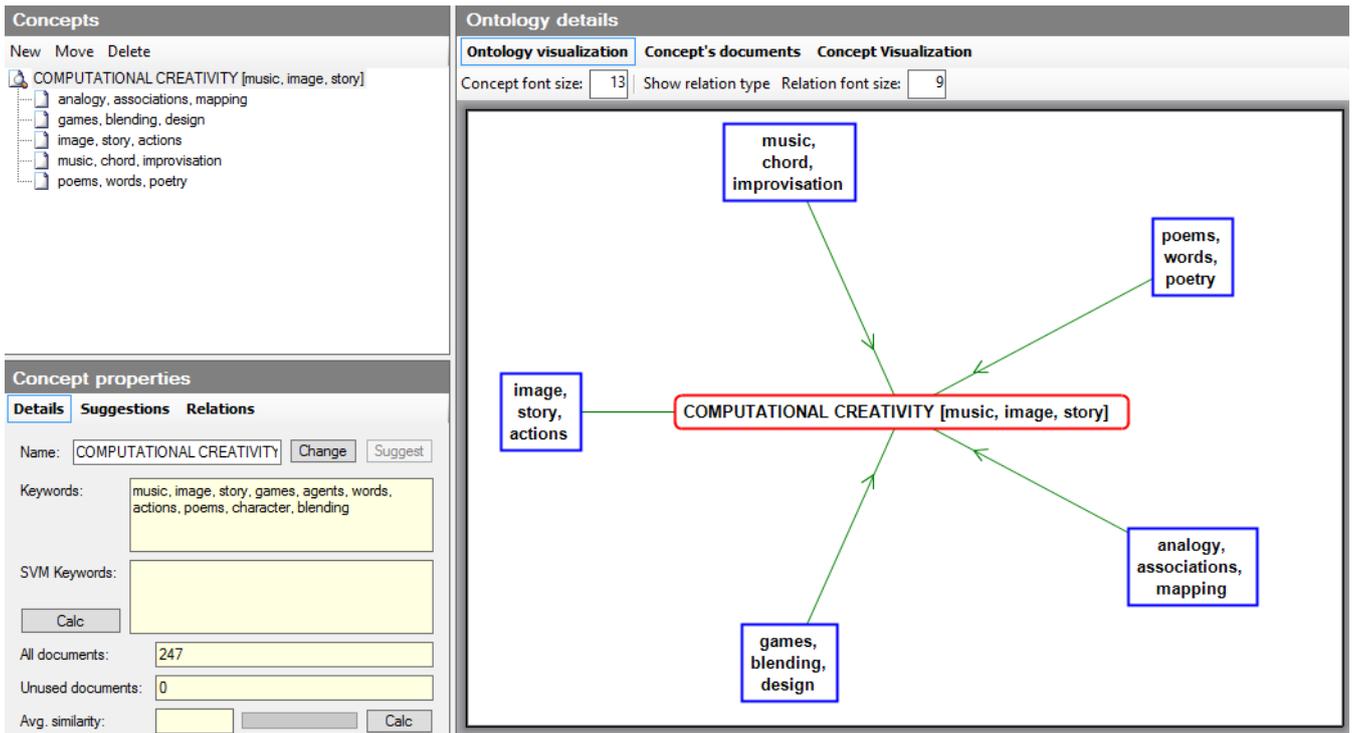


Figure 1: Automatically generated conceptualization of the CC domain.

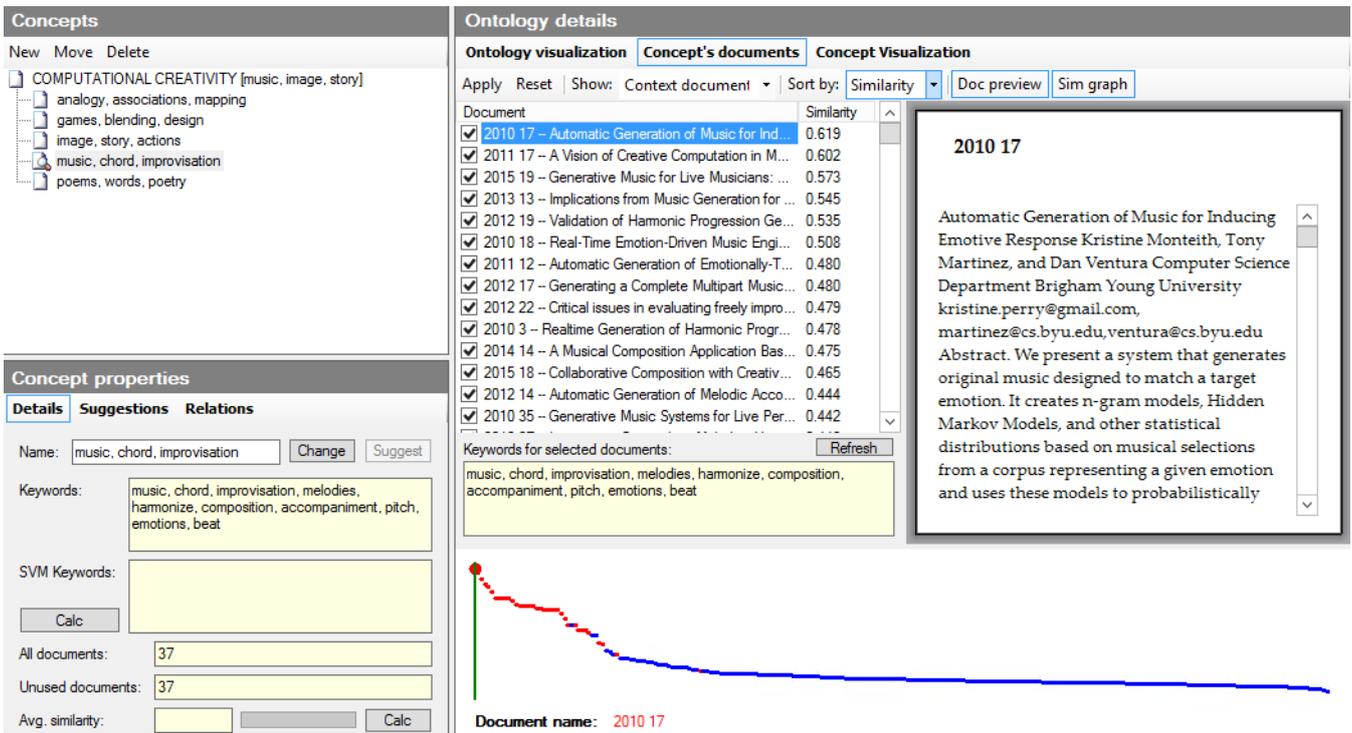


Figure 2: Document view of the automatically generated conceptualization of the CC domain.

Table 3: Automatically generated concepts (concept names were manually determined) and keywords.

Concept	Automatically extracted keywords
Music	<i>music</i> , chord, <i>improvisation</i> , melodies, harmonize, <i>composition</i> , accompaniment, pitch, emotions, beat
Visual	<i>image</i> , story, actions, <i>painting</i> , character, agents, narrative, <i>artists</i> , robot, darci
Linguistic	<i>poems</i> , words, poetry, artefacts, <i>story</i> , evaluating, creativity_system, predict, <i>text</i> , creativity
Games	<i>games</i> , blending, <i>design</i> , analogy, <i>player</i> , conceptual, games_design, angelina, ontology, agents
Conceptual	<i>analogy</i> , associations, <i>mapping</i> , graphs, objective, problem, fractal, domain, <i>representation</i> , relationship
Comp. creativity	<i>music</i> , <i>image</i> , <i>story</i> , games, agents, words, actions, poems, character, blending

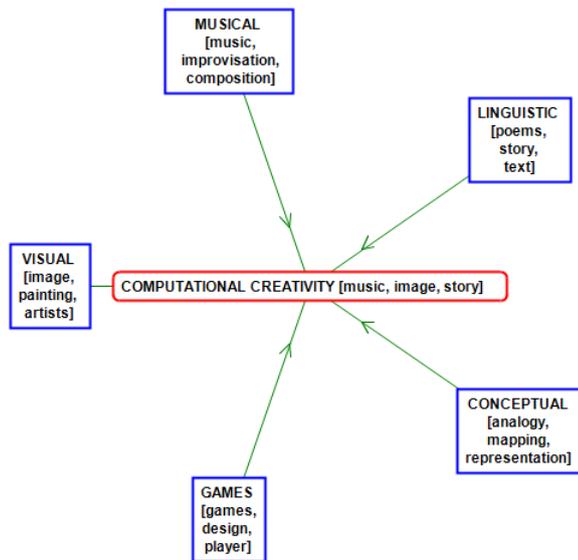


Figure 3: CC conceptualization through automated clustering, concept naming and manual keywords selection.

- Creativity in Story Generation From the Ground Up: Non-deterministic Simulation driven by Narrative (León and Gervás, 2014)

Games:

- Computational Game Creativity (Liapis, Yannakakis, and Togelius, 2014)
- Ludus Ex Machina: Building A 3D Game Designer That Competes Alongside Humans Michael (Cook and Colton, 2014)
- Knowledge-Level Creativity in Game Design (Smith and Mateas, 2011)

Music:

- Automatic Generation of Music for Inducing Emotive Response (Monteith, Martinez, and Ventura, 2010)
- A Vision of Creative Computation in Music Performance Roger (Dannenberg, 2011)
- Generative Music for Live Musicians: An Unnatural Selection (Eigenfeldt, 2015)

Analysis of CC domain development in time

In this section, we investigate temporal aspect of ICCC proceedings. First, we use OntoGen by which we first categorize the articles into editions and then observe the words

distinguishing these categories. Second, the frequency analysis of terms was used to identify terms that are characteristic only for first or last editions. Last but not least, we use the copulas for investing the time dependency between the content of different conference editions.

Distinctive keywords by years using OntoGen OntoGen (Fortuna, Grobelnik, and Mladenić, 2007) is capable of supervised categorization of documents in predefined categories. For this experiment, we used conference years as categories and extracted characteristic keywords for each year. The first set of *descriptive* keywords (KeyW in Table 5) is extracted using document centroid vectors, while the second set of *distinctive* keywords (SVM in Table 5) is extracted from the SVM classification model dividing documents in the topic from the neighbouring ones (Fortuna, Mladenić, and Grobelnik, 2006). Table 5 shows both sets of words for each year.

Unsurprisingly, descriptive keywords overlap different years: words recurring most across years are “creativity, design, modelling, system”. More interesting are the distinctive keywords: ICCC-2015 might be characterized by “bots”, ICCC-2014 and ICCC-2011 by “games”, 2014 by “metaphors” and “stereotypes”, ICCC-2012 by “melodies” and “associations”, and ICCC-2010 by “analogies”.

Categorization by year can also be used for specific topics: in the ontology in Fig. 4, we can split a selected topic into year categories and observe the distinctive (SVM) keywords. The papers representing the concept of Musical creativity in 2015 contain words such as “musebot, pc, unnatural...”, but in 2010, “chord, improvisation, jazz...”.

Terminology distribution by years The frequency distributions of the top 1,000 terms obtained by the terminology extraction process described earlier in the paper are an indicator to detect terms that recently occurred or were present only in the early editions. Examples of terms that appear in 2014 and 2015 and not before are: “game jam, co-creative system, concept invention, generative software, curation coefficient, procedural generation, player goal, network analysis, simulation model” (30 terms in total). In contrast, the terms that were used in 2010 and 2011 are: “fractal representation, fractal feature, basel problem, sensory system, fractal algorithm”.

Copula-based analysis of dependencies between ICCC proceedings

This section describes measuring the detected dependencies between different years of ICCC proceedings. As in the previous section, we counted the frequencies of automatically extracted terms in the ICCC pro-

Table 4: Categories and keywords of the first layer of the semi-automatically constructed CC ontology.

Category	Automatically extracted keywords
Musical	<i>music</i> , chord, <i>improvisation</i> , melodies, harmonize, <i>composition</i> , accompaniment, pitch, emotions, beat
Visual	<i>image</i> , <i>painting</i> , darci, artifacts, <i>collage</i> , adjectives, associations, rendered, colored, artists
Linguistic	<i>story</i> , <i>poems</i> , actions, character, <i>words</i> , agents, narrative, artefacts, poetry, evaluating
Games	<i>games</i> , <i>design</i> , <i>player</i> , games_design, angelina, agents, code, jam, filter, gameplay
Conceptual	<i>analogy</i> , <i>blending</i> , mapping, conceptual, objective, <i>associations</i> , team, graphs, concepts, domain
Evaluation	music, poems, improvisation, <i>evaluating</i> , interactive, poetry, <i>creativity system</i> , musician, <i>participants</i> , behavioural
Comp. creativity	<i>music</i> , <i>image</i> , <i>story</i> , games, agents, words, actions, poems, character, blending

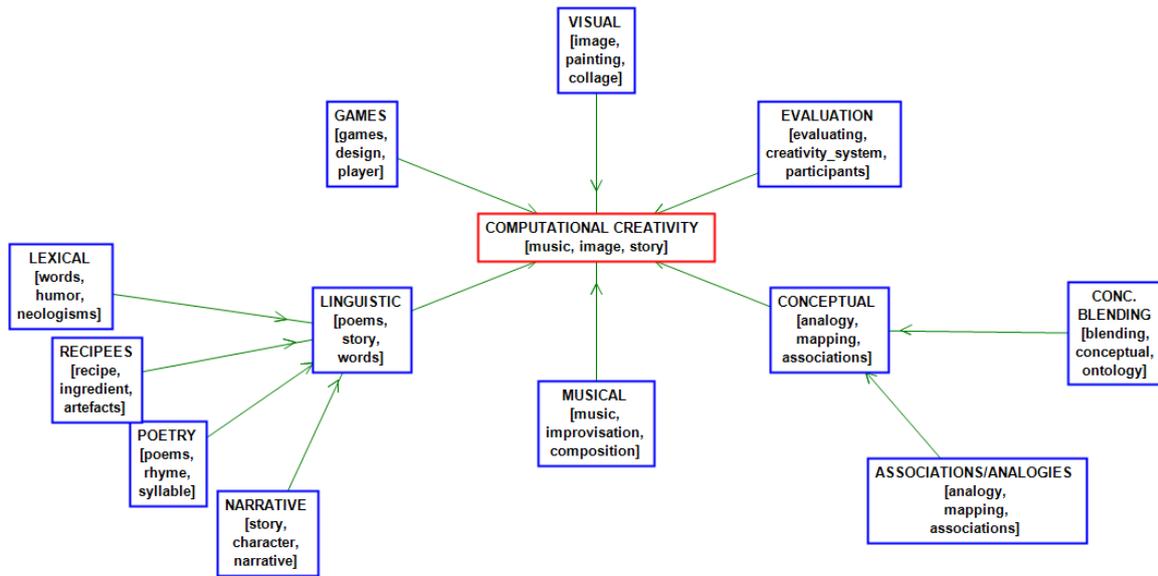


Figure 4: Semi-automatically generated conceptualization of the CC domain, with concept naming and subconcept creation.

Table 5: Keywords and distinctive (SVM) words by year.

Year	Category	Words
2015	KeyW:	creativity, generated, image, work, human, blending, design, based, music, words
	SVM:	<i>blending, humor, bots, choice, vectors, musician, jam, conceptual, colour, participants</i>
2014	KeyW:	creativity, computer, process, modelling, evaluating, words, agents, story, domain, based
	SVM:	<i>games, agents, story, artists, adjectives, ontology, domain, motifs, poems, actions</i>
2013	KeyW:	modelling, process, figure, image, design, performs, based, levels, interactive, concepts
	SVM:	<i>robot, metaphor, motion, surprising, evolved, image, composition, mechanism, stereotypes, fictional</i>
2012	KeyW:	creativity, system, computer, music, evaluating, user, human, figure, work, set
	SVM:	<i>melodies, associations, accompaniment, character, template, shape, player, monotone, text, cluster</i>
2011	KeyW:	creativity, system, story, design, modelling, results, games, music, set, problem
	SVM:	<i>story, games, movements, playing, graphs, games_design, darci, actions, identical, strategies</i>
2010	KeyW:	generated, system, user, set, idea, design, emotions, analogy, developments, based
	SVM:	<i>analogy, chord, emotions, improvisation, genes, filter, lives, team, jazz, songs</i>

ceedings of each year from 2010 until 2015. This information was used as input for the copula-based dependency analysis between six ICC proceedings, described below.

The scatter plot of the terms that occur in 2010 in comparison with terms that occur in years 2011 to 2015 are given

in Fig. 5. The number of occurrences of the specific term in 2010 are given on the x axis, while the number of occurrences of the same term in other years is represented on the y axis. The scatter plot graphically shows a positive dependence between different years as data points are clus-

Table 6: Results from bi-variate copulas for all terms.

No.	Copula type	Coupling of domains	θ
1	Best Clayton	2010-2012	3.1941
2	Best Frank	2010-2012	9.1507
3	Worst Clayton	2010-2015	2.4010
4	Worst Frank	2010-2015	7.3955

Table 7: Two Archimedean copulas.

Copula type	$C_\theta(u, v)$
Clayton	$[\max(u^{-\theta} + v^{-\theta} - 1, 0)]^{-1/\theta}$
Frank	$-\frac{1}{\theta} \ln \left(1 + \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^{-\theta} - 1} \right)$

tered in a band running from lower left to upper right. In the next step, we quantify the dependency between the different years. The most commonly used measure for dependency is correlation. The correlation between two variables (e.g., the ICCC proceedings of two distinctive years) can be measured by means of the Pearsons correlation coefficient. It is a dimensionless quantitative measure of statistical relationships between two (or more) variables. It measures the degree of linear correlation; however the two variables may have different functional dependency. For this reason we apply the copula functions as a tool for studying and measuring the dependences of random variables (Sklar, 1959).

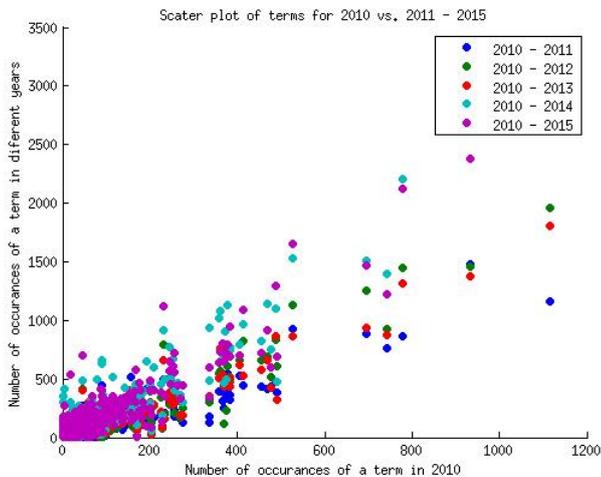


Figure 5: The scatter plot of the terms that occur in 2010 in comparison with terms that occur in years 2011 to 2015.

Copulas are functions that formulate the multivariate distribution in such a way that various general types of dependences including the non-linear one may be captured. We focus on two families of bi-variate Archimedean copulas: Clayton and Frank. Their usage is mainly motivated by their convenient properties, such as symmetry and associativity. Their mathematical forms are presented in Table 7. The parameter θ is estimated from the data. Higher values of θ

mean higher dependence between the two variables.

We explored the dependencies between pairs of proceedings. For this purpose we built Clayton and Frank bi-variate copulas. The results of best copulas (the most dependent pairs of proceedings) and the worst copulas (the least dependent pairs) are provided in Table 6. It can be observed that the ICCC-2010 and ICCC-2015 proceedings are contents-wise the least connected, while the most dependent proceedings are those from 2010 and 2012, where both conferences were organized in Europe.

Conclusions

In the paper, we have presented the conceptualization of the computational creativity domain by semi-automated topic ontology construction based on the corpus of ICCC proceedings. We analysed automatically extracted keywords and subconcepts for CC domains (Visual, Musical, Linguistic, Conceptual creativity, Creativity and games and Evaluation). In addition we analysed characteristics of different editions of CC conferences and used copulas to measure the dependency between proceedings of different editions. As result of this research, we make available for further research a) the ICCC proceedings corpus in .txt format with and without reference sections, b) automatically extracted ICCC terminology that can be used for future efforts in creating a CC glossary, c) fully automated topic ontology with automatic keywords extraction, as well the semi-automated CC ontology, which is the result of manual manipulation of the automatic ontology. Ontologies are available in .png and .rdf formats. All the resources are available publicly⁵

In future, as these techniques develop to full automation, and the amount of data increases with successive conferences, it will be possible to construct a timeline of the conceptual development of the field of computational creativity, using the objective analysis of the literature. It will be possible to trace the rise and fall of trends, and their success or failure, and to identify the development of core CC science, as proposed by (Lakatos, 1970). This activity will be unique in science, and will support an unprecedented level of unbiased philosophical reflection on the field of computational creativity.

Acknowledgements

We acknowledge the support of the Slovenian Research Agency and European projects Prosecco (grant nb. 600653) and ConCreTe, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

- Agres, K.; McGregor, S.; Purver, M.; and Wiggins, G. 2015. Conceptualizing creativity: From distributional semantics to conceptual spaces. In *In the Sixth International Conference on Computational Creativity, ICCC 2015*.

⁵http://kt.ijs.si/senja_pollak/CC_resources/.

- Anderson, A.; McFarland, D.; and Jurafsky, D. 2012. Towards a computational history of the acl: 1980-2008. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, ACL '12, 13–21. Stroudsburg, PA: ACL.
- Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. Routledge.
- Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In de Raedt, L.; Bessiere, C.; Dubois, D.; and Doherty, P., eds., *Proc. ECAI Frontiers*.
- Cook, M., and Colton, S. 2014. Ludus ex machina: Building a 3d game designer that competes alongside humans. In *Proceedings of the Fifth International Conference on Computational Creativity, ICCCC2014*, 54–62. ACC.
- Dannenberg, R. B. 2011. A vision of creative computation in music performance. In *Proceedings of the Second International Conference on Computational Creativity*, 84–89.
- Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.
- Eigenfeldt, A. 2015. Generative music for live musicians: An unnatural selection. In *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, 142–149. Park City, Utah: Brigham Young Uni.
- Fortuna, B.; Grobelnik, M.; and Mladenčić, D. 2007. Ontogen: Semi-automatic ontology editor. In *Human Computer Interface (Part II) (HCI 2007)*, LNCS 4558, volume 4558, 309–318.
- Fortuna, B.; Grobelnik, M.; and Mladeni, D. 2006. Semi-automatic data-driven ontology construction system. In *PASCAL EPprints (2006)* <http://eprints.pascal-network.org/perl/oai2>. Working Group Summary 15.
- Fortuna, B.; Mladenčić, D.; and Grobelnik, M. 2006. *Semantics, Web and Mining: Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3-7, 2005, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg. chapter Semi-automatic Construction of Topic Ontologies, 121–131.
- Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2):199–220.
- Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: a review. *ACM Computing Surveys* 31(3):264–323.
- Jordanous, A., and Keller, B. 2012. Weaving creativity into the semantic web: a language-processing approach. *Proceeding of the Third International Conference on Computational Creativity* 216–220.
- Laclaustra, I. M.; Ledesma, J. L.; Mendez, G.; and Gervás, P. 2014. Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. In *Proceedings of the Fifth International Conference on Computational Creativity*. Ljubljana, Slovenia: Jožef Stefan Institute, Ljubljana, Slovenia.
- Lakatos, I. 1970. Falsification and the methodology of scientific research programmes. In Lakatos, I., and Musgrave, A., eds., *Criticism and the Growth of Knowledge*. Cambridge, UK: Cambridge University Press. 91–196.
- Lavrač, N.; Grčar, M.; Fortuna, B.; and Velardi, P. 2010. *Computational Social Network Analysis: Trends, Tools and Research Advances*. London: Springer London. chapter Exploratory Analysis of the Social Network of Researchers in Inductive Logic Programming, 135–154.
- León, C., and Gervás, P. 2014. Creativity in story generation from the ground up: Non-deterministic simulation driven by narrative. In *5th International Conference on Computational Creativity, ICCCC 2014*.
- Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *Proceedings of the Fifth International Conference on Computational Creativity*. Ljubljana, Slovenia: Josef Stefan Institute.
- Monteith, K.; Martinez, T.; and Ventura, D. 2010. Automatic generation of music for inducing emotive response. In *Proceedings of the International Conference on Computational Creativity*, 140–149. Lisbon, Portugal: Department of Informatics Engineering, University of Coimbra.
- Pérez y Pérez, R.; Ortiz, O.; Luna, W.; Negrete, S.; Castellanos, V.; Alosa, E. P.; and Ávila, R. 2011. A System for Evaluating Novelty in Computer Generated Narratives. In Ventura, D.; Gervás, P.; Harrell, D. F.; Maher, M. L.; Pease, A.; and Wiggins, G., eds., *Proceedings of the Second International Conference on Computational Creativity*, 63–68.
- Pollak, S.; Vavpetič, A.; Kranjc, J.; Lavrač, N.; and Špela Vintar. 2012. NLP workflow for on-line definition extraction from English and Slovene text corpora. In Jancsary, J., ed., *Proceedings of KONVENS 2012*, 53–60. ÖGAI. Main track: oral presentations.
- Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5):513–523.
- Sklar, A. 1959. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris* 8:229231.
- Smith, A. M., and Mateas, M. 2011. Knowledge-level creativity in game design. In *In Proc. of the 2nd International Conference in Computational Creativity, (ICCC 2011)*.
- Smith, B. 2003. Chapter 11: Ontology. In Floridi, L., ed., *Blackwell Guide to the Philosophy of Computing and Information*, volume 7250. Blackwell. 155–166.
- van der Velde, F.; Wolf, R. A.; Schmettow, M.; and Nazareth, D. S. 2015. A semantic map for evaluating creativity. In *Sixth Interantional Conference on Computational Creativity (ICCC 2015)*. Park City, Utah, USA: Brigham Young University.
- Velardi, P.; Faralli, S.; and Navigli, R. 2013. Ontolearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39(3):665–707.
- Vintar, Š. 2010. Bilingual term recognition revisited the bag-of-equivalents term alignment approach and its evaluation. *Terminology* 16:141–159.
- Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems* 19(7):449–458.

An institutional approach to computational social creativity

Joseph Corneli

Computational Creativity Group
Goldsmiths College
London, UK
j.corneli@gold.ac.uk

Abstract

Modelling the creativity that takes place in social settings presents a range of theoretical challenges. Mel Rhodes's classic "4Ps" of creativity, the "Person, Process, Product, and Press," offer an initial typology. Here, Rhodes's ideas are connected with Elinor Ostrom's work on the analysis of economic governance to generate several "creativity design principles." These principles frame a survey of the shared concepts that structure the contexts that support creative work. The concepts are connected to the idea of computational "tests" to foreground the relationship with standard computing practice, and to draw out specific recommendations for the further development of computational creativity culture.

Introduction

One two-part claim is advanced and defended herein: *Elinor Ostrom's theory of institutions can be used to design systems that exhibit computational social creativity, and a culture supports this work.* The contribution takes the form of several candidate "design principles," a literature survey that elaborates them, and an analysis that connects these ideas to common programming practice.

The paper is structured as follows. The "Background" section describes Ostrom's (1990) *Institutional Analysis and Development* (IAD) framework, focusing on her proposed design principles for commons management. To connect these ideas to social creativity, the paper draws on the 4Ps (Person/Process/Press/Product), a model for thinking about creative contexts (Rhodes 1961) that has been brought to bear in theorising computational creativity (Jordanous 2016). This is summarised and slightly adapted. In the subsequent main section of the paper, "Testing for Creativity", Ostrom's design principles are transposed from the world of commons management to the world of computational social creativity. This section looks for ways to connect the proposed creativity design principles to computational methods, and also draws on contemporary thinking in the philosophy of technology, with examples from familiar social computing settings like Wikipedia. A two-part example dealing with both the "soft" culture of the computational creativity community and potential software-based interventions is presented in the "Example" section. Finally, the "Discussion and Conclusions" highlight the relevance of this work for computational creativity culture, systems, and evaluation.

Background

This section summarises the motivation for the paper, introduces Elinor Ostrom's work, and reviews Rhodes's 4P framework. The central parts of this section are Table 1 and 2, which list Ostrom's design principles for managing a commons, and transpose them to creative domains.

Motivation The current investigation is motivated, in part, by the idea of *Ecologically Grounded Creative Practice* (Keller, Lazzarini, and Pimenta 2014). Within a given ecological niche, agents and objects interact; niches can also be brought into relationship in creative ways. The current work has in mind relatively sophisticated agents with their own "contextual maps" and the ability to participate in "reading and writing computational ecosystems" (Antunes, Leymarie, and Latham 2015). Such agents will use, view, critique, and evaluate the work and workflow of other agents. Although computational agents with all of these features do not exist yet in any robust form, we can reason about them, and in so doing, help design the future of *computational social creativity* (Saunders and Bown 2015) – an emerging research area at the nexus of artificial life, social simulation, and computational creativity.

Elinor Ostrom's "design principles" To contextualise this effort, we must begin with a short excursus into economics. Ostrom's work is typically applied to study the management of natural resources. In economics jargon, the specific resources considered are *rivalrous* and *non-excludable*. This means that consumption by one party precludes consumption by a rival, and that it is not directly possible for anyone to block others' access to the resource. Economic goods with these two properties are referred to as *common pool resources* (CPRs); see (Ostrom 2008). Fisheries and forests are important examples. Economic actors have incentives to exploit these resources, however, there are natural limits on total consumption. In principle, a CPR might be gobbled up due to individual greed: this is the so-called tragedy of the commons, and one does not have to look too far for examples. However, in practice, the tragic outcome does not always transpire. Ostrom's theoretical and empirical perspective helps understand why, and emphasises:

(1) the importance of group attributes and institutional arrangements in relation to the structure of incentives and utilities for individual decision making; and (2) the likelihood of a broader set of possible outcomes, including user-group institutional solutions (McCay and Acheson 1990, p. 23)

Ostrom's ideas have recently been applied to analyse Wikipedia, considered as an "expressive commons" (Safner 2016). Wikipedia is *non-rivalrous* in consumption, if we accept the metaphor "to read is to consume." However, *contribution* to Wikipedia presents a range of salient social dilemmas, and efforts to manage them are reflected, for example, in the *Neutral Point Of View* (NPOV) policy, which helps produce "articles that document and explain major points of view, giving due weight with respect to their prominence in an impartial tone."¹

IAD focuses on *action situations*, framed in three phases: *context*, *action*, and *outcome*. Importantly, this part of the theory is not linked to the particular details of CPRs. Ostrom uses the term *institution* to refer to the "shared concepts used by humans in repetitive situations organized by rules, norms, and strategies" (Ostrom 2010). We will return to these concept categories later and consider them further from a computational perspective. For now, our way into thinking in terms of IAD will be by way of several *design principles* for the successful management of CPRs that Ostrom described; see Table 1. These principles work together to support institutions that maintain the integrity of the commons – for example, by ensuring that behaviour is monitored, that knowledgeable and concerned parties are the ones who make specific rules, and that conflicts do not get out of hand (Ostrom et al. 2012, p. 79).

The four Ps We can bootstrap our contextual understanding of creativity with the help of an existing model. Rhodes (1961, pp. 307-309) intends "the four Ps" to refer to the following facets of creativity, which are familiar from everyday experiences of creativity in society.

Person – *personality, intellect, temperament, physique, traits, habits, attitudes, self-concept, value-systems, defense mechanisms, and behavior.*

Process – *motivation, perception, thinking, and communication.*

Product – *an idea embodied into a tangible form.*

Press – *the relationship between human beings and their environment.*

We will shortly use these concepts to rewrite the items in Table 1, replacing the focus on *appropriation* with a focus on *contribution* that befits a theory of social creativity.

Jordanous makes a case for thinking about computational creativity using Rhodes's 4P's, starting with a critique of the strategies used in the evaluation of computational creativity, which, she emphasises, is "traditionally considered ... from

¹https://en.wikipedia.org/wiki/Wikipedia:Five_pillars

Ostrom's design principles

1A. User boundaries

"Clear boundaries between legitimate users and nonusers must be clearly defined."²

1B. Resource boundaries

"Clear boundaries are present that define a resource system and separate it from the larger biophysical environment."

2A. Congruence with local conditions

"Appropriation and provision rules are congruent with local social and environmental conditions."

2B. Appropriation and provision

"The benefits obtained by users from a common-pool resource (CPR), as determined by appropriation rules, are proportional to the amount of inputs required in the form of labor, material, or money, as determined by provision rules."

3. Collective-choice arrangements

"Most individuals affected by the operational rules can participate in modifying the operational rules."

4A. Monitoring users

"Monitors who are accountable to the users monitor the appropriation and provision levels of the users."

4B. Monitoring the resource

"Monitors who are accountable to the users monitor the condition of the resource."

5. Graduated sanctions

"Appropriators who violate operational rules are likely to be assessed graduated sanctions (depending on the seriousness and context of the offense) by other appropriators, by officials accountable to these appropriators, or both."

6. Conflict-resolution mechanisms

"Appropriators and their officials have rapid access to low-cost local arenas to resolve conflicts among appropriators or between appropriators and officials."

7. Minimal recognition of rights to organise

"The rights of appropriators to devise their own institutions are not challenged by external governmental authorities."

8. Nested enterprises

"Appropriation, provision, monitoring, enforcement, conflict resolution, and governance activities are organised in multiple layers of nested enterprises."

Table 1: Ostrom's design principles as expressed in the meta-review carried out by Cox, Arnold, and Tomás (2010) the perspective of the creative output produced by a system" (Jordanous 2016).

Ecological thinking suggests that that it is quite limited to take the final product as the sole term of analysis. At least we might like to introduce the "embedded evaluation" of creative products into the creative process, and build agents that are aware of some contextual features of their environment. For example, these agents might ask: *How similar or how different is my generated artwork to an existing artwork, or*

²Repetition is *sic*, the point being that the boundaries must be both distinct and explicitly defined.

Proposed creativity design principles

- 1A. The population of Producers who can add to or alter the resource is clearly defined.
- 1B. The boundaries of the Place must be well defined.
- 2A. The Process is related to local conditions.
- 2B. Contributing to the Product has benefits for the Producer that are proportional to the efforts expended.
3. Most Producers who are affected by the rules governing contribution can participate in modifying the operational rules.
- 4A. Tests document the interaction of Producers and Place.
- 4B. Tests can be modified by Producers or their representatives.
5. Producers who violate operational rules in the domain will be assessed sanctions by other Producers.
6. Producers have rapid access to low-cost local arenas to resolve conflicts.
7. The rights of Producers to devise institutions governing their contributions are not challenged by external authorities.
8. Contribution, testing, enforcement, conflict resolution, and governance and are organised in multiple layers of nested Places and agencies.

Table 2: “Creativity design principles” formed by switching the polarity of entries in Table 1 to emphasise contribution rather than appropriation, and using the concept of “tests” to connect to computing practice

to the components thereof, or to the initial conception for the work? This route is quite close to Ritchie’s (2007) empirical criteria for judging a final product against an “inspiring set” – but now makes evaluation an explicit part of the creative process. Some recent work in computational creativity emphasises embedded evaluation (Gervás and León 2014). However, as Jordanous argues, creative products are just one part of the overall creative process – and the 4Ps help expose the other features.

Unfortunately, however, Rhodes’s thinking and terminology is too anthropocentric for our current purpose. As Ostrom describes it, action situations are to be understood using seven clusters of variables: *participants, positions, potential outcomes, action-outcome linkages, participant control, types of information generated, and costs and benefits assigned to actions and outcomes* (Ostrom 2009, p. 14). Nowhere does this mention a “Person”. Continuing the adaptations begun by Jordanous (2016), the four Ps will be rendered here as Producer/Process/Product/Place. It is important to emphasise that these labels are strictly more inclusive than Rhodes’s, and more abstract. In particular, the Place corresponds to Ostrom’s action situation, structured in advance by contextual features. This adapted 4P model is reminiscent of the *Domain-Individual-Field Interaction* (DIFI) model due to Feldman, Csikszentmihalyi, and Gardner (1994), if we understand Domain \approx Place, Individual \approx Producer, and Field \approx (a collection of) estab-

lished Processes. Note that contextual theories, broadly construed, pose a long-standing challenge for computing, partly because “what context is changes with its context” (Gundersen 2014, p. 343). One possible working definition is that: “Context is what constrains a problem solving [scenario] without intervening in it explicitly” (Brézillon 1999). Another relevant remark is that context is “defined solely in terms of effects in a given situation” (Hirst 2000).

In developing an *institutional approach to computational social creativity*, we will look for the rules, norms, and strategies that can be used to establish suitable and effective contextual relationships between Process(es), Place(s), Producer(s), and Product(s).

Transposing the design principles into “creativity design principles” and translating them into technical terms *Software testing* is embodied in the formal ideas of *assertions, advice, and contracts*. Related programming methodologies aim to build *executable specifications* and may make use of *test-driven development* (TDD). These techniques provide various ways for (evolving) programs to interact with their context. These ideas can help us translate Table 1 into technical terms. To get started, Table 2 uses the 4P terminology and the generic notion of a test to transpose Ostrom’s design principles into “creativity design principles.”

Testing for creativity

The current section elaborates the candidate creativity design principles outlined above, expanding each with relevant literature and examples, and seeking the ways in which each principle could be applied within a software system.

1A. User boundaries

“The population of Producers who can add to or alter the resource is clearly defined.”

In user-oriented computing, this principle is often addressed using *Access Control Lists* (ACLs) or other permissions mechanisms. The corresponding tests are relatively simple: either each modifiable object in the system has a piece of metadata about it that says who can modify it, or each user has a piece of metadata attached to his or her user account that says which resources they can modify.

Before granting access to a resource, we may require that a Producer implements certain protocols. In a client-server architecture, the client generally communicates using an existing API and may need to implement a certain set of call-back functions or adhere to other restrictions. Noncompliant user behaviour after access has been granted may result in access being revoked. Thus, for example, even though Wikipedia is “the encyclopedia anyone can edit,” violating the site’s principles may lead to a IP-based block, or a username-based ban.

1B. Resource boundaries

“The boundaries of the Place must be well defined.”

The source of this well-definedness may come from “either side.” That is, the Place may advertise its definition in terms of its APIs and other criteria (as above) together with

guarantees on output behaviour in the style of “Design by Contract” (Mitchell and McKim 2002); alternatively, **Producers** may implement tests that restrict the **Places** that they will engage with.

In a simple example of the latter sort, a game-playing agent might resign if it estimates that its position is unwinnable. The fact that different participants can have different perspectives points to an interesting special case in which the (shared) definition of the **Place** arises in an emergent manner. This phenomenon is especially important if we “[take] a broad view of creativity as any process in which novel outcomes emerge” (Saunders and Bown 2015).

2A. Congruence with local conditions

“The **Process** is related to local conditions.”

In Ostrom’s original formulation, local conditions were broken down along axes of “time, place, technology, and/or quantity of resource units” (Ostrom 1990). With respect to theorising the local conditions of creativity, we can gain a useful perspective by turning briefly to the psychoanalyst Winnicott’s treatment of “the exciting interweave of subjectivity and objective observation” which takes place in an “area that is intermediate between the inner reality of the individual and the shared reality of the world” (Winnicott 2002, p. 86). We are then led to consider those local conditions that exist in the “interweave” of **Place** and **Producer**. For example, roboticist Andy Clark proposes a theory of extended cognition, in which enminded beings “use” the environment to self-program and are not just programmed by the environment (Clark 1998). However as Clark points out elsewhere, “it becomes harder and harder to say where the world stops and the person begins” (Clark 2001). In short, the mind is not separated from the body or environment but grounded in perception (Ingold 2000).

A corresponding computational test is found in the earlier example of embedded evaluation, in which existing artefacts are employed as a virtual sensorium. More broadly, this principle concerns making sense of, or “parsing”, the **Place** (and the other **P**’s). This **Process** is well described by Steigler’s notion of *grammatisation*: “processes by which a material, sensory, or symbolic flux becomes a gramme,” or, more simply, “the production and discretisation of structures” (Tinnell 2015). Remember that while a given agent is trying to make sense of the world, others are likely trying to make sense of that agent as well. Framed as dilemma, the last word would likely be: “program or be programmed” (Rushkoff 2010) – but reflecting on Clark’s comment above, we see that this can become somewhat complex.

2B. Appropriation and provision

“Contributing to the **Product** has benefits for the **Producer** that are proportional to the efforts expended.”

The usual way of thinking about computers – as non-agentive machines – would render the above-stated principle perfectly meaningless. In connection with principle 2A, we should here remark: “That an object is more profitable or effective is only a secondary consequence of its refinement” (Chabot 2013, p. 12). In any case, before we can

think about “benefits” in the case of a non-human (and non-living) **Producer**, the phrasing of the current principle leads us to ponder the cost of their “efforts.”

It may be best to change tack, and ask, with Terrance Deacon, “In what sense could a machine be alive?” (Deacon 2014). If a machine were responsible for maintaining its own energy supply, its features of outward-orientation might give cause to say that the machine has a “self” (Deacon, Haag, and Ogilvy 2011). Consider for example the Ethereum project, which provides protocols for distributed computing and the creation of “decentralized autonomous organisations” – whose organisation relative to the outside world is mediated by cryptocurrency, referred to as “fuel” (Wood 2014).

From a testing standpoint, the key requirements are: an ability to judge whether a given option can be (tentatively) thought of as beneficial, and, ideally, a memory that can compare these judgements with iterations of similar situations later on. In this way we would recover the foundations of reinforcement learning, and, as Ostrom points out, the core logic behind the development of new institutions:

“How about if you do *A* in the future, and I will do *B*, and before we ever make a decision about *C* again, we both discuss it and make a joint decision?” (Ostrom 2009, p. 19)

3. Collective-choice arrangements

“Most **Producers** who are affected by the rules governing contribution can participate in modifying the operational rules.”

Let us reflect in more detail on the *rules* that comprise – along with biophysical and material conditions and community attributes – the locally-contextual variables which determine or constrain an action situation (Ostrom 2009, p. 15). At their simplest, these rules are “if-then” statements giving instructions that determine the behaviour of persons in certain roles. As such, each rule contains a logical test, and changing the rules means writing new tests.

Ostrom develops a grammar around this idea, and defines *regulatory rules* with the following formula:

ATTRIBUTES of participants who are OBLIGED, FORBIDDEN, OR PERMITTED to ACT (or AFFECT an outcome) under specified CONDITIONS, OR ELSE. (Ostrom 2009, p. 187)

Norms and *strategies* are defined using a simplified formula, also cast in terms of *attributes*, *deontics*,³ *aim*, and *conditions* (Ostrom 2009, p. 140). The prescriptive terms may be assigned a particular weight, and actions and consequences may also be assigned a particular cost or value (Ostrom 2009, p. 142). Some relevant actions are: *be* in a position, *cross* a boundary, *effect* a choice, *jointly exercise* partial control together with others, *send or receive* information, *pay out or receive* costs or benefits, and *take place* (for outcomes) (Ostrom 2009, p. 191).

³I.e., the prescriptive valence – obliged, forbidden, or permitted, as above – for norms, not for strategies.

Something more needs to be said about the assertion that Producers “can” participate in changing (or creating) rules, norms, and strategies. In practice, participatory systems tend to be lossy. Changes to rules and structures will tend to be carried out by those Producers who are *most* affected – and *thus* most knowledgeable; cf. Ostrom et al. (2012, p. 79). The structure of new rules is predicted by Conway’s Law:

[T]here is a very close relationship between the structure of a system and the structure of the organization which designed it. (Conway 1968)

Specifically, the proposed relationship is “homomorphism”: following Conway, any Product will mirror the hyper-local conditions that describe the Producers’ social context. Furthermore, it seems likely that Products will mirror local environmental conditions in the Place. This points to importance of a broad class of tests that would be described as environmental “sensors”. This theme will be developed more fully below.

4A. Monitoring users

“Tests document the interaction of Producers and Place.”

The straightforward view suggested by the idea of “monitoring” is to deploy some global functionality that keeps track of the actions of all participating Producers within a Place. But this function can be broken up and distributed out among the Producers themselves. In the first instance, what a Producer produces is sensory data. Sensors are generally deployed along with effectors or (more broadly) transducers that translate the sensory information into action. So, monitoring is important for modelling any action or interaction whatsoever. For example, The Painting Fool compares an initial altered snapshot (sensory data) to the painted image that it generates in response to that snapshot, and judges the quality of its output on that basis (Colton and Ventura 2014). This example could be extended to theorise “proprioceptive” sensing and judgement about effected actions more broadly. Filtering upstream data is another simple application of sensors, which Keller (2012) describes as an “ecompositional technique.” In short, an ecological view on monitoring suggests that it can be distributed out among participants and that this is vital for social creativity.

4B. Monitoring the resource

“Tests can be modified by Producers or their representatives.”

The environment itself also filters and selects (Kockelman 2011). Some of these conditions are fatal for living beings in the environment, and more broadly may provide terminating conditions for the constituent Processes in a Place. It would be too much to say that *all* tests can be modified by Producers. Rather, Producers may have programmatic access to those tests which transform potentially fatal (or at least fateful) features of the Place and participating Producers into *data*. This opens up the possibility of directly modifying decision making processes on the one hand, or of passing along information about the fitness landscape to fu-

ture generations of Producers in a (co-) evolutionary framework on the other (DeLanda 2011).

Simply put, *data* is lack of uniformity within some context (Floridi 2016). In the case of monitoring the extractive use of CPRs, direct and compelling feedback about instances of non-uniform or otherwise aberrant resource usage define critical (i.e., decisive) points within a resource management structure. In creative contexts “critique” is no less important.

5. Graduated sanctions

“Producers who violate operational rules in the domain will be assessed sanctions by other Producers.”

Economic sanctions are generally punishments, which are presumed to have a clear meaning or a direct impact on behaviour. However, there are other cases in which Producers’ interactions with other Producers will not be punitive so much as, for example, educative or otherwise formative.

In an artistic context, “sanctions” may range from constructively critical reviews to outright condemnation to no response at all. The *Iterative Development-Execution-Appreciation* (IDEA) cycle (Colton, Charnley, and Pease 2011) introduces *well-being* and *cognitive effort* ratings from which several derived measures of audience response can be computed (e.g., by averaging across audience members). This can readily be extended to a developmental or peer production context. “Audience” might be re-thought as a “public,” or as Rhodes’s “press” (as originally formulated) to capture the idea that its response has a direct effect on the Producer. Inasmuch as the Producer is produced, feedback from the “parent” Producer(s) is especially important to this formative Process.

6. Conflict-resolution mechanisms

“Producers have rapid access to low-cost local arenas to resolve conflicts.”

Wikipedia’s edit wars provide a familiar example (Viegas et al. 2007; Yasseri et al. 2012). These are carried out on the pages of the encyclopedia itself, and resolved using supplementary pages. Machine-generated metadata is relied upon throughout. These mechanisms are low cost: the “stigmergic” self-organisation patterns exemplified by open online communities make fairly minimal demands on participating agents (Heylighen 2015). Nevertheless, structure matters: cases of direct and unresolvable conflict must usually be referred a higher authority, e.g., sitewide guidelines and policies, or available arbitration committees. Opportunities to jointly exercise partial control are, again, often Products, and the creation of a communication channel – a Place within a Place – is another formative Process, which Jakobson (1960, p. 355) calls the “phatic function.”

The theme of local scale suggests more and less representative examples. For instance, academic research is currently organised in a much more segmented and localised format than Wikipedia. *Modularity* is one of three features that are hypothesised to support *commons based peer production* (CBPP) (Benkler 2002). However, CBPP requires not just decomposability into modules but relatively fine *granularity*

of these modules, and as well as a *low cost of integration* to bring disparate pieces of work together once they are completed – possibly “subsidised” by an assistive technology, like Wikipedia’s metadata systems. Creative and scientific writing, at the level of individual papers or books, tends to miss features that would allow this work to scale up (Kim, Cheng, and Bernstein 2014) – even though science and literature represent impressively huge “virtual” collaborations.

The most straightforward test related to this theme is that a **Producer** needs to be able to *detect* conflict, either between itself and other **Producers**, or between incompatible goals. In order to resolve a conflict – or to organise work on a project to avoid conflicts in the first place – a **Producer** will probably need to reason about the project’s structure.

7. Minimal recognition of rights to organise

“The rights of **Producers** to devise institutions governing their contributions are not challenged by external authorities.”

The foremost external authority to be concerned about in a computational creativity setting is the programmer. A “mini me” critique can readily be levelled by CC sceptics (Colton 2012). We are still in early days for autonomous creative systems and general AI, and involvement of programmers and others in teaching systems how to devise institutions is at least as relevant as teaching them how to conform to pre-given instructions.

Keeping in mind the earlier reflections on Winnicott, a relevant set of tests would compare the frequency of user or programmer-generated changes in the system, with the frequency of changes coming from the system itself. This is the thrust of the diagrammatic formalism of creative acts developed by Colton et al. (2014): with considerable further work we could expand the ability of computer systems to participate in, or fully automate, such modelling activity. A basic challenge in applying the formalism from Colton et al. is to identify the individual “creative acts” that a given **Producer** has made. The tests that would reveal these acts in a given stream of **Products** tend to be domain-specific.

8. Nested enterprises

“Contribution, testing, enforcement, conflict resolution, and governance and are organised in multiple layers of nested **Places** and agencies.”

That the **Place** or the **Producer** would be layered isn’t a surprise; many systems have a hierarchical aspect. What is perhaps more surprising is that many of the features that make up a “creative ecosystem” must themselves be *produced*, which points to the inherent multiplicity of **Producers**. Here, **Producers** are seen as self-organising the structure of their interrelationships and interconnections at various levels. Developing a computational treatment of such a system divorced from real world applications would be a thankless and ultimately futile task. Effort may be better spent on developing programs that model and participate in existing creative ecosystems. In such cases, there would be real-world empirical tests of success, coming from users.

Example

This section uses the creativity design principles discussed above to describe some of the creativity-supporting institutions in place at the Seventh International Conference on Computational Creativity (ICCC 2016), and to explore potential additions and adaptations for future ICCCs.

I. The crucible for the current paper was a unique set of ongoing discussions (see “Acknowledgements”) (2A). At first, the hope was to co-author the paper with one of these discussants, but due to time constraints this was not possible, so it became a single-author paper (1A). The ICCC call helped motivate writing up the ideas (2B), partly because the conference is open to papers that are informed by and contribute to various disciplines at varying degrees of formality. However, ICCC enforces rigorous academic standards, using slightly different evaluation criteria for papers submitted to each of five “tracks” (1B). Reviewers used the EasyChair website to bid for papers to review, and to share discussions and debate about these papers in case of disagreement (6). Papers that were seen as less relevant were rejected outright, or potentially (as a norm) allocated briefer slots in the conference schedule (5). The current paper was conditionally accepted, which meant that it entered into a “shepherding” process, whereby a senior programme committee member could check (4A) whether the author followed through on specific reviewer requirements (4B). By and large authors are given free rein to write papers about any topic relevant to computational creativity, if they do so in a rigorous academic style (7). This entails reflecting on certain themes-held-in-common – but the conference seems to lose some opportunities for structuring engagement more deeply, e.g., around common tools or challenge problems (8?). Presumably only the conference steering committee can change the conference’s overall rules; however, it should be noted that reviewer requirements constitute fine-tuned rule-setting at the level of individual papers (3?).

II. The reflections above begin to suggest ways in which we might make better use of software systems in creative partnership. One realistic idea would be to use computer programs to help with paper review tasks. Essay grading software is now mainstream, and services like WriteLab can help authors simplify their writing and catch grammar and logic errors.⁴ Agent-based reviews or a shift to *post-publication review*, in which reviews are offered “after an article is published, much like commentary on a blog post” (Ford 2013, p. 316) would change the population of reviewers (1A). Moving beyond blogs to wikis, lists of open problems from prior publications could be collected, compared, and explicitly referenced with semantic links (1B) (Tomlinson and others 2012). This could begin to make explicit the ways in which a given paper constitutes an advance (2A, 2B). The development, use, and maintenance of shared tools (APIs, open source software) and design patterns for computational creativity could be encouraged (3). A standardised testing approach based on challenge problems, as in the

⁴<https://writelab.com> offers a freemium service for students, but is “always free for instructors.”

recently announced OpenAI Gym,⁵ with worked examples, explicit evaluation metrics, and variant versions (4A, 4B) could help the community move towards, and enforce, standards of *replicability* and *generalisability* (5). Partial “wikification” and semantisation of the research area is already underway with systems like FloWr (Charnley, Colton, and Llano 2014) and ConCreTeFlows (Znidaršič et al. 2014), but it is unclear whether these systems will merge, or diverge, or if a new standard will come along (6?). Once shared technologies and datasets are in common use, computational agents will be better able to contribute to the field (7). It is to the advantage of computational creativity researchers to develop applications and application environments that we – and others – agree are useful (8).

Discussion and Conclusions

This section reviews the contribution above, beginning with a link to related work. Specifically, Ostrom’s high-level Institutional Analysis and Development (IAD) framework can be regarded in parallel with the high-level outline of the *Standardised Procedure for Evaluation of Creative Systems* (SPECS) (Jordanous 2012). SPECS suggests that, in order to evaluate creativity, it is necessary to put forth a definition of “what it is to be creative,” and then to specify criteria by which creativity will be measured before formulating the evaluation. IAD suggests that institutions operate within a certain context, which afford certain kinds of actions, and that these lead to certain observable outcomes. To wit:

IAD	SPECS
Context	Definition
Action	Criteria
Outcome	Evaluation

In IAD, context can be thought of as a collection of “exogenous variables,” (Ostrom 2009, p. 13, esp. Figure 1.1) including pre-defined rules, that shape what happens in the action situation at the heart of the analysis. We have described several candidate design principles that outline potential rules for guiding action in creative settings. This suggests the possibility of recording a definition and set of criteria for evaluating social creativity in a general domain. Pragmatically, this definition might unpack the 4Ps in terms of Ostrom’s variables (participants, positions, etc.).

SPECS could be criticised for being overly abstract: in other words, for simply describing good practice in any empirical investigation. IAD adds many more specifics, which have necessarily been presented in a compressed form here. It is hoped that this first attempt to use IAD to theorise computational social creativity will motivate future explorations that further unpack social creativity using Ostrom’s ideas.

The creativity design principles offer guidelines (and with minor changes, hypotheses) for members of the computational creativity community to test out in practice. More empirical work is needed to validate (or improve) these principles. On the cultural side, more attention should be given to the fact that our institutions – including institutions for building institutions – are analysable in programmatic terms.

⁵<https://gym.openai.com/>

Acknowledgements

This work was supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553 (COINVENT).

Ongoing conversations with Anna Jordanous, Steve Corneli, and Rasmus Rebane about computational creativity, economics, and semiotics (respectively) were key drivers in this paper’s evolution. The anonymous reviewers made many useful comments on matters of presentation.

References

- Antunes, R. F.; Leymarie, F. F.; and Latham, W. 2015. On writing and reading artistic computational ecosystems. *Artificial life* 21(3).
- Benkler, Y. 2002. Coase’s Penguin, or Linux and the Nature of the Firm. *Yale Law Journal* 112:369.
- Brézillon, P. 1999. Context in problem solving: a survey. *The Knowledge Engineering Review* 14(1):47–80.
- Chabot, P. 2013. *The philosophy of Simondon: Between technology and individuation*. Bloomsbury.
- Charnley, J.; Colton, S.; and Llano, M. T. 2014. The FloWr Framework: Automated Flowchart Construction, Optimisation and Alteration for Creative Systems. In Ventura et al. (2014).
- Clark, A. 1998. *Being there: Putting brain, body, and world together again*. MIT press.
- Clark, A. 2001. *Natural-born cyborgs?* Springer.
- Colton, S., and Ventura, D. 2014. You Can’t Know my Mind: A Festival of Computational Creativity. In Ventura et al. (2014).
- Colton, S.; Pease, A.; Corneli, J.; Cook, M.; and Llano, T. 2014. Assessing Progress in Building Autonomously Creative Systems. In Ventura et al. (2014).
- Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*, 90–95.
- Colton, S. 2012. The Painting Fool: Stories from Building an Automated Painter. In *Computers and creativity*. Springer. 3–38.
- Conway, M. E. 1968. How do committees invent? *Datamation* 14(4):28–31.
- Cox, M.; Arnold, G.; and Tomás, S. V. 2010. A review of design principles for community-based natural resource management. *Ecology and Society* 15(4):38.
- Deacon, T.; Haag, J.; and Ogilvy, J. 2011. The emergence of self. In Wentzel Van Huyssteen, J., and Wiebe, E. P., eds., *In Search of Self: Interdisciplinary Perspectives on Personhood*. Wm. B. Eerdmans Publishing Co.
- Deacon, T. W. 2014. In what sense could a machine be alive? Thursday 3rd April, 9:30-10:30, AISB’50, Convention Plenary, Goldsmiths College, University of London.
- DeLanda, M. 2011. *Philosophy and simulation: the emergence of synthetic reason*. Continuum.

- Feldman, D. H.; Csikszentmihalyi, M.; and Gardner, H. 1994. *Changing the world: A framework for the study of creativity*. Praeger Publishers/Greenwood Publishing Group.
- Floridi, L. 2016. Semantic conceptions of information. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab, Center for the Study of Language and Information (CSLI), Stanford University.
- Ford, E. 2013. Defining and characterizing open peer review: A review of the literature. *Journal of Scholarly Publishing* 44(4):311–326.
- Gervás, P., and León, C. 2014. Reading and Writing as a Creative Cycle: the Need for a Computational Model. In Ventura et al. (2014).
- Gundersen, O. E. 2014. The role of context and its elements in situation assessment. In Brézillon, P., and Gonzalez, A. J., eds., *Context in Computing*. Springer. 343–357.
- Heylighen, F. 2015. Stigmery as a Universal Coordination Mechanism: components, varieties and applications. *Human Stigmery: Theoretical Developments and New Applications*.
- Hirst, G. 2000. Context as a spurious concept. In Gelbukh, A., ed., *International Conference CICLing-2000: Conference on Intelligent Text Processing and Computational Linguistics (Proceedings)*, 273–287.
- Ingold, T. 2000. *The Perception of the Environment: Essays on Livelihood, Dwelling and Skill*. Routledge.
- Jakobson, R. 1960. Linguistics and Poetics. In Sebeok, J., ed., *Style in Language*. MIT Press. 350–377.
- Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation* 4(3):246–279.
- Jordanous, A. 2016. Four PPPPerspectives on Computational Creativity in theory and in practice. *Connection Science* 28:194–216.
- Keller, D.; Lazzarini, V.; and Pimenta, M. S. 2014. Ubimus Through the Lens of Creativity Theories. In *Ubiquitous Music*. Springer. 3–23.
- Keller, D. 2012. Sonic ecologies. In Brown, A. R., ed., *Sound musicianship: Understanding the crafts of music*. Cambridge Scholars Publishing. 213–227.
- Kim, J.; Cheng, J.; and Bernstein, M. S. 2014. Ensemble: exploring complementary strengths of leaders and crowds in creative collaboration. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 745–755. ACM.
- Kockelman, P. 2011. Biosemiosis, technocognition, and sociogenesis: Selection and significance in a multiverse of sieving and serendipity. *Current Anthropology* 52(5):711–739.
- McCay, B. J., and Acheson, J. M. 1990. *The question of the commons: The culture and ecology of communal resources*. University of Arizona Press.
- Mitchell, R., and McKim, J. 2002. *Design by Contract, by Example*. Addison-Wesley.
- Ostrom, E.; Chang, C.; Pennington, M.; and Tarko, V. 2012. *The Future of the Commons: Beyond Market Failure and Government Regulation*. Institute of Economic Affairs.
- Ostrom, E. 1990. *Governing the commons: The evolution of institutions for collective action*. Cambridge: Cambridge University Press.
- Ostrom, E. 2008. The challenge of common-pool resources. *Environment: Science and Policy for Sustainable Development* 50(4):8–21.
- Ostrom, E. 2009. *Understanding institutional diversity*. Princeton University Press.
- Ostrom, E. 2010. Institutional analysis and development: Elements of the framework in historical perspective. In *Historical Developments and Theoretical Approaches in Sociology*, volume 2. EOLSS. 261–288.
- Rhodes, M. 1961. An analysis of creativity. *The Phi Delta Kappan* 42(7):305–310.
- Ritchie, G. D. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.
- Rushkoff, D. 2010. *Program or be programmed: Ten commands for a digital age*. Or Books.
- Safner, R. 2016. Institutional Entrepreneurship, Wikipedia, and the Opportunity of the Commons. <http://ssrn.com/abstract=2564230>.
- Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial life*.
- Tinnell, J. 2015. Grammatization: Bernard Stiegler’s theory of writing and technology. *Computers and Composition* 37:132–146.
- Tomlinson, B., et al. 2012. Massively distributed authorship of academic papers. In *CHI’12 Extended Abstracts on Human Factors in Computing Systems*, 11–20. ACM.
- Ventura, D.; Colton, S.; Lavrač, N.; and Cook, M., eds. 2014. *Fifth International Conference on Computational Creativity, ICCO 2014*. Association for Computational Creativity.
- Viegas, F. B.; Wattenberg, M.; Kriss, J.; and Van Ham, F. 2007. Talk before you type: Coordination in Wikipedia. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, 78–78. IEEE.
- Winnicott, D. W. 2002. *Playing and reality*. Routledge. (Original work published 1971.)
- Wood, G. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*.
- Yasserli, T.; Sumi, R.; Rung, A.; Kornai, A.; and Kertész, J. 2012. Dynamics of conflicts in Wikipedia. *PLOS ONE* 7(6):e38869.
- Znidaršič, M.; Miljković, D.; Perovšek, M.; Pollak, S.; Kranjc, J.; Cherepnalkoski, D.; and Lavrač, N. 2014. First report on framework and data. Technical Report D6.2, ConCrete: Concept Creation Technology, Project Number 611733. ICT – Future and Emerging Technologies (FET).

Understanding Musical Practices as Agency Networks

Andrew R. Brown

Griffith University
Brisbane, Australia

andrew.r.brown@griffith.edu.au

Abstract

This position paper proposes that creative practices can be usefully understood as agency networks. In particular it looks at interactive algorithmic musical practices and the takes a distributed view of the influences involved in such music making. The elements involved include humans, tools, culture and the physical environment that constitute a system or network of mutual influences. Such an agency network perspective is intended to be useful for the pragmatic tasks of designing new interactive music systems and developing new musical practices that utilise them. Drawing on previous research into generative music and computational creativity, various views on interactive music systems are canvassed and an approach to describing these as agency networks is developed. It is suggested that new human-machine musical practices may arise as a result of adopting an agency network perspective and that these, in turn, can drive cultural innovations.

Introduction

There have been many attempts at defining creativity in either humans, computational systems or co-creative interactions between them. In this position paper I propose that creative acts may, instead, be understood as networks of agency. This approach may be useful in computationally creative systems research in particular where philosophical questions about self-awareness, intentionality, and embodiment of machines can become problematic.

Definitions of computational creativity that focus on the outcomes provide quite some latitude for the effect of devices on this outcome independent of human influence. For example Boden states “Computational creativity (CC, for short) is the use of computers to generate results that would be regarded as creative if produced by humans alone” (Boden 2015:v). Other definitions have been more ambitious (e.g., Wiggins 2006:451) by implying a stronger sense of computer autonomy than suggested by Boden’s phrase, “use of computers”. Rather, these

definitions suggest that the goal of computational creativity is for computational behavior itself to be deemed creative by human standards.

This definitional preoccupation can create confusion and disagreement amongst the field and, perhaps of more concern, it may limit avenues of research and development in human-computer artistic co-creation by discouraging pragmatic investigations. As Boden acknowledges, “Whether computers can ‘really’ be creative isn’t a scientific question but a philosophical one, to which there’s no clear answer. But we do have the beginnings of a scientific understanding of creativity” (Boden 2014:23).

Acknowledging my motivation toward the pragmatic production of interesting music and in the interests of promoting intellectual frameworks that stimulate artistic co-creation research, I suggest that agency networks (in the spirit of actor network theory) can usefully account for the contribution of people, machines, and cultural contexts to musical activities and outcomes. An agency network perspective is a distributed view of the influences involved in music making, or other creative tasks. The elements involved in the network include humans, tools, cultural conventions, and the physical environment; these constitute a system or network of mutual influences on creative processes and outcomes.

Notions of agency in creative tasks can provide a useful common ground between the intentional stance attributed to humans in such actions and the functionality and constraints attributed to tools and environments, particularly because when we look intently into creative action “the line between human intention and material affordances becomes all the more difficult to draw” (Malafouris 2008:33). In short, the agency network approach to displays of musicality defers claims to creativity and shifts evaluative judgements toward the pragmatics of personal or cultural value.

This perspective bears some relationships to Oliver Bown’s suggestion that we can evaluate creativity as “actors forming temporary networks of interaction that

produce things” (Bown 2015:21). The agency network approach supports his view that creative authorship can be distributed to varying degrees between humans, tools culture and environment. Inherent in this perspective is that these creative relationships can be symmetrical in their influence (i.e., coupled) but may not be symmetrical in their contribution (i.e., varying roles and degrees of attribution). Bown proposes that such a view takes us beyond the consideration of either humans or machines as “islands of creativity” to a more nuanced evaluation of creativity. In this position paper I propose to additionally suggest that a view of creativity as a network of agencies may also have an epistemological claim to understanding, and perhaps even be the basis for generative processes for the design of human-computer co-creative systems.

In this article I will focus on music because that is the domain I am most familiar with; it may be that similarities can be found with other creative arts activities or even in other endeavours. I will be particularly interested in co-creativity within interactive music systems, but suggest that human-machine relationships are unavoidable even in what appear to be autonomous human or machine creative acts.

The article begins by examining the effects of algorithmic technologies on musical practice and musical culture, and investigates the making of music with generative computational systems as an emerging creative practice. It explores the impact that cybernetic interactions between musicians and algorithmic media have on conceptions of creativity and agency, and the potential to influence cultural evolution.

Background

As computing systems have become more powerful in recent years, real time interaction with ‘intelligent’ computational processes has emerged as a basis for innovative creative practices. Examples of these practices include: interactive digital media installations, generative art works, live coding performances, virtual theatre, interactive cinema, and adaptive processes in computer games. In these types of activities, computational systems have assumed a significant level of agency, or autonomy, provoking questions about shared authorship and originality, about aspects of musicianship with interactive technologies, and about the future of musical genres where these practices are employed. These issues are redefining our relationship with technology and fomenting new debates about human capabilities, values and the meaning of productive activities.

Cybernetic interactions—those between people and technologies—have been recognised, periodically, as having the potential to influence musical developments (Machover and Chung 1989; Pressing 1990; Rowe 1993; Miranda 2000; Dean 2003; Pachet 2002; Gifford and Brown 2013). Recent theoretical advances in understanding the relationship between embodied cognition and music technologies lay the groundwork for the next stage of these developments (Leman 2008, Borgo 2012). These ideas are manifest in creative practices and, by using those insights to elaborate notions of musical agency, we may be better able to appreciate co-creation with generative media.

At the heart of all creative interactions is a sense of creative agency—the opportunities and responsibilities for decisions and actions in creative activities. Cybernetic co-creation, where creative control is shared with technologies, challenges our understanding of agency—both human and non-human. Research has examined how expert musicians manage these collaborations (Winkler 1998; Brown 2003; Collins 2006; Gurevich 2014). To date, researchers have mostly focused on individual instances of algorithmic music in experimental music contexts, but opportunities are increasing to study virtuosic practices in mainstream practices. This work has helped to identify the salient features of music interactions with algorithmic media and to use them to account for theories of co-creation and musical agency, in order to inform future cultural innovation and development. Musical practices that include algorithmic media—typically computers running interactive and/or generative software—and our interactions with them have been studied in recent years by this author (Brown 1999; 2001; 2005; Brown, Gifford and Wooller 2010; Brown, Gifford and Voltz 2013) and a number of others (Pressing 1990; Rowe 1993; Cope 2000; Pachet 2002; Nierhaus 2010).

In previous work I, and co-authors, have argued that to build and use “generative software that operates appropriately in a creative ecosystem, we must secure some understanding of how we interact with our existing partners and tools, and how they interact with us” (Jones, Brown and d’Inverno 2012:200). An underlying proposition in that work is that music made with interactive software constitutes its own form of musical practice and that opportunities for stimulating cultural development result from these new creative relationships. It is also important to appreciate how this interactive practice builds on a long history of technological usage more broadly. In the language of the philosophy of

technology, tools (including musical instruments) may be engaged with as ready-to-hand, under conscious utilitarian control, or as present-at-hand, experienced as an embodied engagement or in ‘flow’ (Heidegger 1977, Ihde 1979). Experiences with automated media transcend this duality in that technologies appear to us as musical partners with their own agency. This type of human-machine discourse—where “two entities are acting reciprocally upon one another”—has been labelled Interactionism (Agre 1997:53). Specifically this kind of internationalism involves moving from a technological representation of music, such as notated scores and recorded audio data, to a technological simulation (generation) of musical actions and outcomes. Generative algorithms might simulate compositional processes, human behaviours, or sociocultural conditions. Margaret Boden suggests that computer artists value the degree of machine autonomy that such automation provides; they find it, Boden suggests, aesthetically more interesting than when the computer is treated as ready-to-hand, or as a “slave” (Boden 2010:190).

Investigations into music making with automated media, such as those described in previous surveys of the field in Joel Chadabe’s (1997) *Electric Sound* and Roger Dean’s (2003) *Hyperimprovisation*, highlight the historical explorations in interactive algorithmic music and, in particular, the role of chance in providing novelty, and of improvisation (especially by the human being) in adapting to changing or unexpected events. These researchers also underscore the stylistic innovation associated with algorithmic musical practices over past decades, particularly the aesthetic connections with electroacoustic music, sound art and, more broadly, with experimental music.

Human-machine co-creation

In her book on computer art, Boden defines creativity as “the generation of novel, surprising and valuable ideas” and explicitly includes musical concepts and artefacts within the term ‘ideas’ (Boden 2010:1). She outlines three types of creativity; combinatorial, exploratory, and transformational. Of particular interest here is that, firstly, computers seem quite capable of these processes (perhaps with some limitations in assessing value) and, secondly, that her definition leaves open the possibility of also adopting Mihaly Csikszentmihalyi’s assertion that creativity “arises from the synergy of many sources and not only from the mind of a single person” (1996:1). Co-creation between musician and algorithmic media meets this criterion and resonates with the associated theory of

distributed cognition, which acknowledges that our competence is reliant on support from the world around us (Merleau-Ponty 1962; Perkins 1993; Clark 1997). Just as in the past, when musicians have relied on each other and acoustic instruments for enhanced musical expression, so today and in the future, algorithmic computer systems do and will play their part. How these interactions operate for effective musical outcomes can be usefully understood, I propose, by thinking about them as networks of elements with particular agency. Different musical practices will arise from different configurations of agency networks.

Examples of musical practices that include algorithmic media are: *Generative Music* (Eno 1996), *Live Algorithms* (Blackwell, Bown and Young 2012), *Live Coding* (Collins et al. 2003), *Interactive Music Systems* (Rowe 1993), *Mobile Music Making* (Tanaka 2004), and *Algorithmic Composition* (Cope 2000). These involve the kinds of interactions typical of most human musical collaborations, such as synchronisation and coordination, outlined as crucial by David Borgo (2005) in his interrogation of musical improvisation amongst jazz musicians. To date, algorithmic musical practices have been employed predominantly in experimental or avant-garde musical genres.

Less obviously, perhaps, automated media have played a part in the rise of contemporary electronic (dance) music since the latter part of the 20th century (Kirn 2011). Software sampling and sequencing technologies have been significant in the development of these genres. In general, technologies such as step sequences and parameter control, while ‘automated’, are not generally characterised as algorithmic, although algorithmic processes have been increasingly present in commercial music technologies in recent years (e.g., Apple Logic Pro’s ‘Drummer’). Some notable EDM artists, including Aphex Twin and Autechre, have taken advantage of algorithmic techniques. Driven by technological and cultural transfer from academic and experimental practices—like those described above—to popular music, the need to appreciate and articulate the characteristic of interaction with algorithmic music processes is all the more pressing. Models of interactive music practices as an agency network ‘system’ can play a part in assisting the understanding and design of these new musical practices.

The emergent behaviour of human-machine co-creation practices implies that we consider the human and machine components as part of a creative system, a perspective that is particularly favoured in the field of

cybernetics. The uses of Cybernetic principles within digital arts I have previously reviewed (Gifford and Brown 2013). A more detailed overview of Cybernetics is provided by Andrew Pickering's (2010) history of the field, which includes some references to its use in the arts, and extends his earlier work exploring human interactions with the materiality of the world, specifically in the field of scientific discovery.

Musical co-creation between humans and computationally creative software has accelerated in recent decades as the computing tools for real-time interactive media and the means of audience interaction through mobile devices have become ubiquitous. It is timely that an agency network perspective be catalyst for re-examining these interactions and, in particular, exploring their use in contemporary culture. The focus of such a perspective, as proposed here, is a better appreciation of the concept of musical agency as it applies to all elements of the co-creative 'system'.

Toward a networked approach to musical agencies

Agency can be simply defined as the ability to produce an effect. This definition is often constrained further to the production of an *intended* effect. Human beings have always been accepted as having agency, especially through their ability to act intentionally to satisfy needs and desires. Ascribing non-human agency, however, requires intellectual care. Going even further, to describe algorithmic media as "creative machines" (Lewis 2011:460)—as we might wish to do in situations of co-creation—is particularly precarious as debates within the computational creativity community attest.

For the purposes of this article I will refer to the capacity of human beings or technologies to generate music as their *musical agency*. It might seem controversial to ascribe agency to non-living things; however, inspired by the work of anthropologist Alfred Gell (1998) it seems reasonable to say that artefacts and machines have (at least) a relational agency that depends upon their interaction with human intentions and cultural conventions. Gell suggests that inanimate artefacts (like works of art) can be influential and 'cause things to happen' within a cultural context. It seems less controversial, then, to suggest that 'animated' machines capable of generating sound automatically (such as generative computer music software), might have musical agency. This arises, following Gell's logic, because of their relationships or interaction with human makers,

performers, and audiences—a cultural context that contains intention and meaning—as part of the to-and-fro of creative collaborations (Brown 2012; Brown, Gifford and Voltz 2013). Lambros Malafouris further suggests that such interaction itself may not 'say much' about the agency of interacting elements, but he suggests that we look to see what "constitutes a meaningful event in the larger enchainment of events that constitute the activity" for greater insights into the presence of 'pragmatic agency' (Malafouris 2008:25).

An early application of the notion of agency to music appeared in Timothy Taylor's book *Strange Sounds* (2001), where he focused on the influence of electronic and digital technologies on musical culture. He did not, however, examine the impact of algorithmic approaches. With a not-dissimilar cultural agenda, the proposition I pose here is that understanding creativity as a network of agencies may influence the ways algorithmic technologies are integrated into musical practice. Like many of the relevant writers in this field, Taylor considers musical culture to be a "system" made up human, technical and social forces—the position most famously suggested by Bruno Latour in his Actor Network Theory (Latour 2007). While generally supportive of the role of technologies in moving musical culture forward, Taylor often characterises technologies as constraining. In celebrating human re-use, or misuse, of a technology for new musical purposes—as when DJs repurpose turntables—he suggests this is evidence that "Human agency struck back" (Taylor 2001:204) against the 'resistance' of technical design. My view of this interaction is more optimistic than Taylor's.

Also drawing on Actor Network Theory as a model, Pickering examined how people and material things are interrelated and each has an effect on how activities (in his case, science) play out. "The basic metaphysics of the actor-network is that we should think of science (and technology and society) as a field of human and nonhuman (material) agency. Human and nonhuman agents are associated with one another in networks, and evolve together within those networks. The actor-network picture is thus symmetrical with respect to human and nonhuman agency" (Pickering 1995). Pickering's more recent book, *The Cybernetic Brain* (2010), extended this view of material agency within an historical survey of the pioneers of cybernetics, some of whom explored cybernetic principles in audio-visual contexts, and Pickering himself has a growing interest in the connection of material agency to the arts (personal correspondence).

Theories of material agency have been applied to artistic contexts such as making pottery (Malafouris 2008). Recently, Chris Salter applied the notion of material agency directly to musical processes, in particular sound installations. Salter is especially concerned with the materiality of sound and sonic environments and the way in which artists and audiences interact with this materiality. Like Gell, Malafouris and Pickering he attributes the agency of objects to their contextuality: "... agency is not located in objects or things but situated in practice, it is 'in the flow of the activity itself'" (Salter 2015:40). With possible extensions of this view, the position taken in the article is that it might be helpful to attribute agency more directly to non-human actors, or as an emergent property of interaction between them, within musical practices.

The term 'musical agency' has been used on previous occasions; for example Blackwell, Bown and Young defined it as "the influence someone or something has on a body of music" (2012:164). This definition is one similar to that applied to agency in general, but constrained to the musical context; it indicates, however, an explicit acknowledgment of human and non-human agency. In his more recent writing Bown makes an even more explicit claim along these lines that "All human creativity occurs in the context of networks of mutual influence" (Bown 2015:17).

Such literature attests to a growing interest in the issue of agency as an explanatory theory about the operation of creative practice. I suggest, however, that there are problems in using underspecified terms too liberally, and in directly applying to artistic contexts, those concepts (such as material agency) that have been worked out in other domains. Therefore, it is proposed that there is a need to explore alternatives that might lead to more detailed and appropriate definitions and understandings of musical agency.

In the case of automated media, such as algorithmic music software, there might be more to agency than 'reflected glory' during interaction. This is not only because of the generative capability of computer systems, but perhaps also because agency need not be simply 'present' or 'absent'. Instead, there can be degrees of agency, and a non-human agent might have limited, or partial, agency within the network of co-creative relationships. Some could argue that agency is only awarded by the transferred intentionality of its designer/programmer; the hypothesis, explored here, is that algorithmic agency is an inherent potential and

independent of human intentionality. A potential that can be realised (or emerge) through interactivity.

The idea of partial agency or, perhaps, dimensions of agencies may appear somewhat intuitive, but was formally proposed by Victor Kaptelinin and Bonnie Nardi (2006). To some degree, in opposition to Pickering (and Latour), they suggest that agents in a network of interaction might have asymmetric degrees of agency. That is, the human might have more, or different, agency than a computer system but it would still make sense to talk of the computer as having agency in that limited way. A "more expansive treatment of agencies is needed", suggest Kaptelinin and Nardi, "to capture the complexity of phenomena related to modern technologies, especially intelligent machines" (2006:243). In particular, I suggest, we need to consider how ideas about networks of musical agency may lead to a better understanding of the dynamics of creative musical practices (and creativity more generally), especially practices with algorithmic systems. This perspective resonates with George Lewis' view: "Understanding computer-based music-making as a form of cultural production obliges a consideration of the discourses that mediate our encounters with the computer itself" (Lewis 2011:457). It follows then that, not unlike Salter (2015) suggests, theories about musical agency may provide insights into musical practices that employ algorithmic processes, and might open new opportunities for evolving musical culture.

Cultural evolution with algorithmic media

There is a common narrative around technology-driven human development. Daniel Pink provides a succinct summary, writing; "Last century, machines proved they could replace human backs. This century, new technologies are proving they can replace human brains" (Pink 2005:44). Musical examples of this include Colon Nancarrow's Studies for Player Piano (1948-1992), where machine performance challenged the physical limits of human performative capability, and the software Shazam that can 'listen to' and identify musical works even when our own memory fails us.

Pink cites the defeat of chess champion Garry Kasparov as a case in point of cognitive skill replacement. His recipe for moderating interpretations of this as technological determinism, is to add "the capacity for art and heart to our penchant for logic and analysis" (Pink 2005:222). This is not such a new prescription. A more authoritative source is the philosopher Martin Heidegger who, in his essay *The Question Concerning Technology*,

observed that “the essence of technology is nothing technological” and went on to suggest that “essential reflection upon technology and decisive confrontation with it must happen in a realm that is, on the one hand, akin to the essence of technology and, on the other, fundamentally different from it. Such a realm is art” (Heidegger 1977:35). It is in this spirit of adopting a poetic orientation towards the technological that an agency network view of musical practices with computational media is proposed. A poetic (aesthetic) view corresponds, also, with a more pragmatic understanding of agency networks as an evaluative frame for co-creative music making.

Rather than being drawn into pessimism due to technological determinism, there are reasons to be optimistic about algorithmic music as a creative force and stimulus for cultural development; indeed there are pockets of society in which this is already occurring. Specifically, there are notable individuals who have worked diligently to bring together the skills required to make this practice a success. The musicians cited throughout this article are some of these. If history is any guide, then cultural leaders should pave the way for this practice to become more mainstream. At present, success requires persistence and passion. Fortunately, both music and computing are pursuits that people become passionate about, and where the pursuit of virtuosity—either as a performer or as a software developer (hacker)—is desirable and exemplars well documented (Turkle 1984; Pachet 2012).

Adopting an agency network approach to creativity research and system development may provide a more comprehensive picture of emerging cultural practices, taking care to account for the complexities of these creative acts within our current technoculture. Observing the mutual influences of musicians, technologies and cultures should help refine notions of musical agency. Such an approach takes account of the dynamics of cultural developments arising from musical interactions with computational media, so that new understandings might lead to a better appreciation of these practices, and provide some predictive power to inform the design of future interactive music systems and music activities with them.

Conclusion

The work reviewed here supports the position of the article that creative practices can be usefully understood as an agency network. This position shifts the focus of

attention from individual objects, actors or elements as being (or not) creative, and moves our gaze toward a distributed view of interactions and relations amongst participating influences.

Agency networks are systems of participating elements that have varying types and degrees of agency. Elements in the systems are ‘coupled’ such that they are mutually influencing, but their contributions to the musical outcome are not the same, and generally not considered equal. As emphasised by Kaptelinin and Nardi, agency varies in different dimensions (yet to be fully worked out), and the relationship between agencies is dynamic and changes over time. In the language of Pickering and Malafouris, within the ‘dance of agencies’ different elements may take the lead at different times.

The perspective provided by considering creative systems as agency networks is useful for the pragmatic tasks of designing new interactive music systems and developing new musical practices that utilise them.

Some may consider that an agency network approach to describing creativity simply side-steps the issue of creativity altogether, perhaps it does. But if one’s objective is to improve artistic and innovative outcomes using computational systems, then the development of theoretical positions that provide more diversity and nuance, such as describing types and degrees of agency, may well stimulate new approaches and tactics. If one’s objective is purely philosophical, to understand or computationally model creativity, then it may be that reconfiguring theoretical discussions around agency may not suffice. Also, there remain questions of perceived autonomy, and of human predilection to seeking relationships of cause and effect in the world—even where none exist. An agency network perspective may not directly address these issues but its foregrounding of the distributed nature of influences in music making systems opens up questions for further consideration by computational creativity researchers, designers of computer music systems, and musicians who interact with those systems.

References

- Agre, P. E. 1997. *Computation and Human Experience*. San Diego: University of California.
- Blackwell, T., Bown, O., & Young, M. 2012. Live Algorithms: Towards autonomous computer improvisers. In *Computers and Creativity* (pp. 147–174). London: Springer.

- Boden, M. A. 2010. *Creativity and Art: Three roads to surprise*. Oxford, UK: Oxford University Press.
- Boden, M.A., 2014. Computer models of creativity. *AI Mag* 30.
- Boden, M. A., 2015. How Computational Creativity Began, in: Besold, T.R., Schorlemmer, M., Smaill, A. (Eds.), *Computational Creativity Research: Towards Creative Machines* (pp. v–xiii). Atlantis Press, Barcelona, Spain.
- Borgo, D. 2005. *Sync or Swarm: Improvising music in a complex age*. New York: Continuum.
- Borgo, D. 2012. Embodied, situated and distributed musicianship. In A. R. Brown (Ed.), *Sound Musicianship: Understanding the Crafts of Music* (pp. 202–212). Newcastle upon Tyne: Cambridge Scholars Publishing.
- Bown, O., 2015. Attributing Creative Agency: Are we doing it right? In *Proceedings of the Sixth International Conference on Computational Creativity*, p. 17-22.
- Brown, A. R. 1999. Tools and Outcomes: computer music systems and musical directions. In *Imaginary Space: The Australasian Computer Music Conference* (pp. 16–22). Wellington: The Australasian Computer Music Association.
- Brown, A. R. 2001. How the computer assists composers: A survey of contemporary practice. In G. Munro (Ed.), *Waveform 2001: The Australasian Computer Music Conference* (pp. 9–16). Sydney: The Australasian Computer Music Association.
- Brown, A. R. 2003. *Music Composition and the Computer: An examination of the work practices of five experienced composers*. (PhD thesis). The University of Queensland, Brisbane.
- Brown, A. R. 2005. Generative Music in Live Performance. In T. Opie & A. R. Brown (Eds.), *Australasian Computer Music Conference* (pp. 23–26). Brisbane, Australia: ACMA.
- Brown, A. R. 2012. Creative Partnerships with Technology: How creativity is enhanced through interactions with generative computational systems. In *Proceedings of the Eighth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Stanford, CA: AAAI.
- Brown, A. R., Gifford, T., & Wooller, R. 2010. Generative Music Systems for Live Performance. In *First International Conference on Computational Intelligence* (p. 290). Lisbon, Portugal: Springer.
- Brown, A. R., Gifford, T., & Voltz, B. 2013. Controlling Interactive Music Performance (CIM). In M. L. Maher, T. Veale, R. Saunders, & O. Bown (Eds.), *Proceedings of the Fourth International Conference on Computational Creativity* (p. 221). Sydney: The Association for Computational Creativity.
- Chadabe, J. 1997. *Electric Sound: The past and promise of electronic music*. Upper Saddle River, NJ: Prentice-Hall.
- Clark, A. 1997. *Being There: Putting brain, body, and world together again*. Cambridge, MA: The MIT Press.
- Collins, N., McLean, A., Rohrhuber, J. & Ward, A. 2003. Live Coding in Laptop Performance. *Organised Sound*, 8(3), 321–330.
- Collins, N. 2006. *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems*. Cambridge University, Cambridge. Retrieved from <http://www.informatics.sussex.ac.uk/users/nc81/thesis.html>
- Cope, D. 2000. *The Algorithmic Composer*. Madison: A-R Editions.
- Csikszentmihalyi, M. 1996. *Creativity: Flow and the psychology of discovery and invention*. New York: Harper Collins.
- Dean, R. 2003. *Hyperimprovisation: Computer-Interactive Sound Improvisation*. A-R Editions, Middleton.
- Eno, B. 1996. Generative music: evolving metaphors, in my opinion, is what artists do. *Motion Magazine*, July, 7.
- Gell, A. 1998. *Art and Agency: An anthropological theory*. Oxford: Clarendon Press.
- Gifford, T. & Brown, A. R. 2013. Cybernetic Configurations: Characteristics of Interactivity in the Digital Arts. In K. Cleland, L. Fisher, & R. Harley (Eds.), *Proceedings of the 19th International Symposium of Electronic Art* (pp. 1–3). Sydney: ISEA International.
- Gurevich, M. 2014. Skill in Interactive Digital Music Systems. In K. Collins, B. Kapralos, & H. Tessler (Eds.), *The Oxford Handbook of Interactive Audio*. Oxford: Oxford University Press.
- Heidegger, M. 1977. *The Question Concerning Technology and Other Essays*. New York: Harper & Row.
- Ihde, D. 1979. *Technics and Praxis*. Dordrecht, Netherlands: D. Reidel Publishing.
- Jones, D., Brown, A. R. & d' Inverno, M. 2012. The Extended Composer: Creative reflection and extension with generative tools. In J. McCormack & M. d' Inverno (Eds.), *Computers and Creativity* (pp. 175–203). London: Springer.

- Kaptelinin, V. & Nardi, B. A. 2006. *Acting with Technology: Activity Theory and Interaction Design*. Cambridge, MA: The MIT Press.
- Kirn, P. 2011. *The Evolution of Electronic Dance Music*. Milwaukee, WI: Backbeat Book.
- Latour, B. 2007. *Reassembling the social: An introduction to actor-network-theory*. Oxford, UK: Oxford University Press.
- Leman, M. 2008. *Embodied Music Cognition and Mediation Technology*. Cambridge, MA: The MIT Press.
- Lewis, G. E. 2011. Interactivity and Improvisation. In R. Dean (Ed.), *The Oxford Handbook of Computer Music* (pp. 457–466). Oxford: Oxford University Press.
- Machover, T. & Chung, J. 1989. Hyperinstruments: Musically Intelligent and Interactive Performance and Creativity Systems (pp. 186–190). In, *International Computer Music Conference*. ICMA, San Francisco.
- Malafouris, L. 2008. At the potter's wheel: An argument for material agency, in: Knappett, C., Malafouris, L. (Eds.), *Material Agency: Towards a Non-Anthropocentric Approach* (pp. 19–36). New York: Springer.
- Merleau-Ponty, M. 1962. *Phenomenology of Perception*. London: Routledge and Kegan Paul.
- Miranda, E. R. 2000. *Readings in Music and Artificial Intelligence*. Harwood Academic Publishers, Amsterdam.
- Nierhaus, G. 2010. *Algorithmic Composition: Paradigms for Automated Music Generation*. New York: Springer.
- Pachet, F. 2002. Playing with Virtual Musicians: the Continuator in Practice. *IEEE Multimedia* 9(3), 77–82.
- Perkins, D. 1993. Person-plus: A distributed view of thinking and learning. In G. Salomon (Ed.), *Distributed Cognitions: Psychological and educational considerations* (pp. 88–110). Cambridge: Cambridge University Press.
- Pickering, A. 1995. *The Mangle of Practice: Time, Agency and Practice*. Chicago: The University of Chicago Press.
- Pickering, A. 2010. *The Cybernetic Brain: Sketches of another future*. Chicago: The University of Chicago Press.
- Pink, D. H. 2005. *A Whole New Mind: Moving from the information age to the conceptual age*. Crows Nest, NSW: Allen & Unwin.
- Pressing, J. 1990. Cybernetic issues in interactive performance systems. *Computer Music Journal* 14(1), 12–25.
- Rowe, R. 1993. *Interactive Music Systems: Machine listening and composing*. The MIT Press, Cambridge, MA.
- Salter, C. 2015. *Alien Agency: Experimental encounters with art in the making*. Cambridge, MA: The MIT Press.
- Tanaka, A. 2004. Mobile music making. In *Proceedings of the 2004 conference on New interfaces for musical expression* (pp. 154–156). National University of Singapore.
- Taylor, T. D. 2001. *Strange Sounds: Music, Technology and Culture*. New York: Routledge.
- Turkle, S. 1984. *The Second Self: Computers and the Human Spirit*. New York: Simon and Schuster.
- Wiggins, G. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems*, 19(7), 449–458.
- Winkler, T. 1998. *Composing Interactive Music: Techniques and ideas for using Max*. Cambridge, Massachusetts: The MIT Press.

A History of Creativity for Future AI Research

Arthur Still

Durham University

UK

awstill@btinternet.com

Mark d’Inverno

Goldsmiths, University of London

UK

dinverno@gold.ac.uk

Abstract

We look at two traditions for talking about creative activity, one originating in the classical Latin use of the word “creare” as a natural process of bringing about change, the other in Jerome’s later use in the Vulgate bible, referring to the Christian God’s creation of the world from nothing but ideas. We aim to show that because the latter tradition has predominated recently in the fields of Psychology and Artificial Intelligence these academic fields have been limited in scope to the Western culture of individualism and progress. We argue that the former tradition is a more general and useful notion as it applies more readily to describing human experience and activity as well as applying equally to other non-western cultures. Furthermore, because both traditions are still alive, and since they are both referred to through the use of this word “creativity”, there is chronic confusion in everyday modern discourse as well as in Psychology and Artificial Intelligence. We outline these two traditions in order to understand and unpick this confusion and discuss implications for future research.

Introduction

The modern science of creativity started in 1950 when JP Guilford published his paper “Creativity” which he had read that year as the presidential address to the American Psychological Association. This word soon replaced the established concept of “creative imagination” (Engell, 1981) which was studied by a wide range of Psychologists interested in creative activity. Guilford was an expert in Psychometrics, the measurement of mind, and he offered “creativity” as a measurable psychological power or propensity, distinct from the familiar “intelligence”. It was presented as a power that would explain the products of “creative genius”, of Einstein and Picasso, as well as more mundane inventions in industry and war, and the imaginative productions of children and adults. Guilford defined the word by explaining that “The creative person has novel ideas” (Guilford 1950:452), and “creativity refers to the abilities that are most characteristic of creative people”.

It was soon recognised that it was not enough just to have new ideas; they have to result in something of value: “The creative work is a novel work that is accepted as tenable or useful or satisfying by [sic] a group in some point in time” (Stein, 1953: 311). This definition of creativity, involving novelty and value has dominated both Psychology and Artificial Intelligence (AI), as in Margaret Boden’s “the ability to

generate novel, and valuable, ideas” (Boden, 2009:24). The scope of this modern concept (from child art to Michelangelo) carried with it the old mystery of how the human mind, the product of evolution, could be behind the astonishing achievements of “creative genius”. The task of Psychology and AI therefore, has been to understand scientifically the mechanisms underlying these achievements.

Until around the 1920s the word “creativity” was rarely used, but when it did appear it did not refer to a psychological propensity, but to new productions and changes in a culture or an individual, as in “the cycle of creativity, in which the languages from which our present tongues are derived, were formed” (Stuart-Glennie, 1874). or “the period of Shakespeare’s [sic] dramatic creativity which produced Cymbeline and The Tempest” (Ward, 1899: 240).

The word “creative” on the other hand was common during the early 20th century. “Creative imagination” had been around since the 18th century, and this was the name given to the process of thinking underlying creative activity, studied especially by Developmental Psychologists interested in the ability of children to think and act imaginatively (Ribot 2006). But “creative” was used in a different sense by John Dewey in the title of the book he edited in 1917 called “Creative Intelligence”. Dewey, and the other authors of this book, used the word “creative” to express their belief that intelligence is inherently a process of inquiry and reflection that comes through a strong sense of “being in the world”, in direct opposition to the more mechanical conception of intelligence contained in IQ tests. As a practical expression of this belief “Creative writing” was pioneered in schools that were based on Dewey’s principles (Mearns, 1925), and several significant works on *creative activity* appeared, where the emphasis is on the ongoing experience of a human being in the world. These included Dewey’s seminal *Art and Experience* (1934) and Wertheimer’s *Productive Thinking* (1945), two of the most important psychological works on this topic that have ever been written. Wertheimer’s chapter on Einstein succeeded in showing both Einstein’s brilliance and the grounding of his thought in experience; without any of the mystery about it stirred up by treating creativity as a power in the mind

The reason that “creativity” started to become more popular seems to have been Alfred North Whitehead’s use of the word for the process of generating novelty in the theory

of physical evolution that was the mainstay of his process metaphysics (Whitehead, 1976, first published 1929). After the publication of Whitehead's philosophy, the word "creativity" became more common in academia, and towards the end of the 1940's it turned up as a buzz word in the fields of marketing and self-help. It figured prominently in Alex Osborn's best seller, *Your Creative Power*, where it replaced his earlier use of "creative imagination" which was the standard term used by Psychologists investigating creative activity. Osborn was an advertising executive and the inventor of Brainstorming as a way of releasing creative power from social inhibitions. Guilford may well have been encouraged by this to develop his own version of "creativity" as a mental power, with its definition, essentially that of Alex Osborn, of creativity as the generation of valuable novelty. This definition certainly works well for Osborn's usage in marketing and it understandably thrives in a capitalist economy that depends upon the never-ending supply of new commodities. It works less well when the product is not a marketable commodity, but an activity like dance, or jazz improvisation, or the traditional painting of icons and illuminated manuscripts. For instance the Lindisfarne Gospels from the early 8th century were recently put on display in the Bishop Cosin library under the shadow of Durham Cathedral in the Northeast of England. The manuscripts were extraordinary and beautiful, high in quality and appropriate to their original purpose. It is, as the Exhibition Guide points out

*. . . one of the greatest landmarks of human cultural achievement. Created by the community of St Cuthbert on Lindisfarne ? an outstanding example of creativity and craftsmanship from medieval times.*¹

Most people would agree that the author of the guide was right to consider them an example of creativity, and stretching a point we may say they were novel, nothing exactly like it had been done before. But this stretches too far since it is not "novelty" that the authors of the guide are referring to when they speak about creativity but something much deeper. The manuscripts are not so much novel as the high point of a tradition, like the Alhambra in Granada, a John Coltrane Solo, a Ming vase, or Bach's B minor Mass.

But in spite of these difficulties, the definition of creativity as novelty has stuck and, we believe, obscured the pioneering work of Dewey and colleagues starting with his concept of creative intelligence. This, we argue, deserves to play a much stronger role in thinking about our approach to designing novel computational systems. In this paper we try to show historically how and why creativity has become such an overloaded term, as a way of disambiguating two versions which provides us with a fuller understanding about how we approach the design of novel computational systems in general.

The profligacy of "creativity"

After 1950, the word became Protean in its scope. Astonishingly for a term of scientific enquiry, and following on directly from Guilford, the word took on a wide variety of

¹Exhibition Guide, Lindisfarne Gospels in Durham, 2013 p.3

meanings. It was at once a psychological power, and a process in the mind, as well as also being the product of that process (Eysenck, 1995). In addition, it has retained some of the pre-Guilford older meanings of an activity taking place in the world. With creativity it seemed there was room for everyone.

In a much quoted paper Rhodes (Rhodes, 1961) translated the 4 P's of successful marketing - "Price, Product, Promotion, and Place" - into the 4 P's of creativity, "Person, Process, Product and Press", where Press refers to the environmental determinants. Creativity is the result of the 4 P's. There is nothing special about this, since the same scheme of 4 P's could be used for "achievement" or "innovation" or "depression". What is odd and different in the case of "creativity" is that the word is not only the result of the 4 P's, but *can refer to each of the first three of the four P's, as well as to the system that incorporates all 4 of them*. Whilst it would be absurd to identify any of the 4 P's of marketing with successful marketing itself, this is what occurs in the case of "creativity". Creativity needs Creativity to explain itself. It is no surprise therefore to find that one leading Theologian has identified it with God. Gordon Kaufman, the late Mallinckrodt Professor of Divinity at Harvard Divinity School substituted "creativity" for "the word" in the opening verses of the Gospel of John: "in the beginning was creativity, and the creativity was with God, and the creativity was God." (Kaufman, 2004: ix).

Major contradictions are not hard to find. It is uniquely human ("the distinctively human capacity to generate new ideas, new approaches, and new solutions" (Hennessey and Amabile, 2010: 570)), but it also occurs in animals (Bateson and Martin, 2013); and it is an unmitigated good ("positive value is a crucial part of the definition of creativity"; (Boden, 1994: 558)) but takes on a malevolent form in the hands of bad people like terrorists (Cropley et al, 2010). No one takes much notice of such anomalies since the word is so malleable that it can readily be shaped to fit every situation.

Given all this, it is not surprising that early on in the modern career of "creativity" Liam Hudson could write:

This odd word ['Creativity'] . . . applies to all those qualities of which psychologists approve. And like so many other virtues . . . it is as difficult to disapprove of as to say what it means. As a topic for research, "creativity" is a bandwagon; one which all of us sufficiently hale and healthy have leapt athletically abroad. (Hudson 1966: 100-101).

and little has changed since Hudson wrote this

The word ["creativity"] has, historically, undergone several shifts in meaning, and it continues to mean different things to different people. (Cardosa et al 2009: 21).

and such a word, vague but redolent with promise and progress, is a gift to politicians, for Kennedy in 1962

. . . we are coming to understand that the arts incarnate the creativity of a free society. We know that a totalitarian society can promote the arts in its own way—that it can arrange for splendid productions of opera and

*ballet, as it can arrange for the restoration of ancient and historic buildings. But art means more than the resuscitation of the past: it means the free and unconfined search for new ways of expressing the experience of the present and the vision of the future. When the creative impulse cannot flourish freely, when it cannot freely select its methods and objects, when it is deprived of spontaneity, then society severs the root of art.*² (Kennedy, 1962)

and President Obama in his 2011 State of the Union address

What we can do - what America does better than anyone else - is spark the creativity and imagination of our people (quoted in Bateson and Mason, 2013: 85)

Scientists themselves have been eager to lend support to this patriotic mission. As Guilford declared soon after the launch of Sputnik by the USSR, when the anxiety about “Falling Behind”³ was at its height:

The preservation of our way of life and our future security depend upon our most important national resources: our intellectual abilities and, more particularly, our creative abilities. It is time, then, that we learn all we can about those resources (Guilford, 1959: 469).

And 50 years later, when the clash of civilizations had replaced that of political ideologies, Hennessey and Amabile wrote, in the 2010 Annual Review of Psychology:

If we are to make real strides in boosting the creativity of scientists, mathematicians, artists, and all upon whom civilization depends, we must arrive at a far more detailed understanding of the creative process, its antecedents, and its inhibitors. The study of creativity must be seen as a basic necessity. (Hennessey and Amabile, 2010:570)

In the past, the beginning of a new science has always been marked by a new precision in the use of key concepts, a process analysed in detail by the French historian of science Gaston Bachelard (Tiles, 1984). But precisely the opposite has happened in the case of creativity. We have seen a reasonably precise and developing discourse on creativity and creative activity before 1950 turn into semantic chaos. The earlier science of Dewey and Wertheimer was ignored, or even denied after Guilford had led the way by declaring himself “appalled” at the historical neglect of creativity. But we are not here to blame Guilford. What happened goes deeper. He unwittingly amalgamated two very powerful traditions in thinking about creative activity, and much of our present

²John F. Kennedy: Magazine Article “The Arts in America.”, December 18, 1962. Online by Gerhard Peters and John T. Woolley, The American Presidency Project

³Teitelbaum (2014). It continues: “It is the education, skill, creativity, and entrepreneurship of a country’s population that will determine whether it will prosper or fall behind in the twenty-first century” (Teitelbaum, 2014:1). See also Cohen-Cole (2009, 2014) for a detailed and critical historical account of how the ideal of an open-minded, creative individual became a defining feature of cold war politics, and shaped the progress of cognitive science.

confusion about creativity follows from their unmarked mingling in both everyday talk and academic discourse.

Here we illustrate this by outlining the history of these two traditions, and end by using the distinction between them to point towards a possible way forward for Computational Creativity.

The history of two traditions

In the classical Latin of Cicero and Lucretius, “creare” had meant bringing about or having an impact through natural forces. In Lucretius’ first century poem *On the Nature of Things* (Lucretius, 1992), where he described a version of evolution which was materialistic but not strictly mechanistic, because his atoms were liable to chance “swerves”. Create was part of the natural (including human) world of creation and dissolution, as in the birth of a child (the father being the creator, mother creatrix), or the growth of plants. A little earlier Cicero had used create to refer to the founding of Rome by Romulus and the appointment of a consul. Create was different from “facere”, to make out of available materials, as in the world-making of the old creator Gods described in Plato’s *Timaeus* (Plato, 1977), and facere was used in early Latin versions of the Bible.

But in St Jerome’s 4th century version, known as the Vulgate Bible, the word create was used instead of facere, and this was taken to mean creation out of nothing but ideas in God’s mind. Once done the creation was distinct from Himself but what He actually did to bring it about was a matter of debate. William Harvey, who discovered the circulation of the blood, suggested that the Creation came into being with a nod from God, giving the matter some scientific credibility, since Harvey falsely believed that the words ‘nod’ and ‘neurone’ were etymologically connected (Kassler, 1991).

This new Christian meaning of create was a mixture of the old create and facere, so that the word “create” means to bring about by making. But it has never succeeded in obliterating the older Pagan meanings, in which create (to bring about) is distinct from facere (to make). Both meanings of create, Christian and Pagan, often occur together in English. For instance, both senses occur in Mary Shelley’s 1818 novel on the scientific creation of a human being, *Frankenstein*. Dr Frankenstein writes (drawing on the Christian sense) “I began the creation of a human being” (Shelley, 1985: 52). Later in the book, having murdered Frankenstein’s child, the monster uses the Pagan meaning:

I too can create desolation; my enemy is not invulnerable; this death will carry despair to him, and a thousand other miseries shall torment and destroy him. (Shelley 1985: 138).

Both meanings are present 200 years later:

*Reggae-Jazz crossover SKAMEL arrived, almost unanimously, with spectacular facial hair creations . . . they were impressive but in no way prepared us for the musical spectacular with which they laid waste to an excellent, enthusiastic and appreciative audience, thereby creating a long-memorable evening of superb jazz.*⁴

⁴<http://jatpjazz.blogspot.co.uk/p/been-gone-and-wow.html>

To distinguish these two meanings we will call Jerome's (Christian) meaning of "creare" G-creative (where G stands for God, but could also be linked to Genius or Guilford). And the older (Pagan) meaning of "creare" N-Creative, where N stands for nature, as for Lucretius creare was the unfolding of natural processes. In this scheme SKAMEL's hair creations were G-creative and creating a long-memorable evening N-creative.

The cognates "create", "creation", and "creative" have become increasingly popular over the centuries. "Creativity", on the other hand remained rare until the 20th century. In the 13th century Duns Scotus used it as though it were a kind of power possessed by God. William of Ockham ridiculed Scotus for this (Leff, 1975: 148), as though you were to take the general word (collective noun) for horses, "equinity", and treat it as an independent entity with causal powers over and above individual horses, and say "equinity enables us to travel faster than walking". If God created the world (which Ockham didn't doubt) it was because that is what God did, not because of some mysterious power called "creativity".

G-Creative

During the Renaissance the biblical sense of God's Creation, the inventing of a world out of nothing but ideas, was taken as analogous to artistic productions, and the analogy was common in discussions of art and poetry in Italy in the 15th and 16th centuries (Charlton, 1913; Panofsky, 1968), and in later discussions of "creative genius" (Abrams 1953: 381), a notion that became common during the 18th century. This analogy with God applied to "high art", great paintings, poetry and music, and it led to a contrast with crafts or low art made out of given materials, and relying only on skill rather than ideas.

The analogy was in the background when modern Psychology arose from the problem of accounting for mental processes in the mechanistic world of 17th century Physics. The most pressing problem was to explain how works of creative genius (by definition original and non-mechanical) can emerge from the association of ideas, in the mechanistic systems put forward by the two great English philosophers of the 17th century, Hobbes and Locke. Their associationist theories formed a closed system in that outputs are explained by lawful processes that occur within a mind that is separate from the world, from which it passively receives inputs, like messages. This creates a mystery, sometimes explained by inspiration, and this pattern of thought has continued to this day. As Herb Simon wrote over three centuries later:

The notion that creativity requires inspiration derives from puzzlement about how a mechanism (even a biological mechanism like the brain), if it proceeds in its lawful mechanistic way, can ever produce novelty (Simon 1995: 945).

Both Hobbes and Locke had recognised the problem, and neither were inclined to appeal to supernatural inspiration. Hobbes suggested what we might refer to as the "Spaniel Search Metaphor" as a solution. In this account, the inventive mind searches through ideas in imagination and memory like a spaniel that "ranges the field, till he finds a scent"

(Hobbes, 1962: 22). This search metaphor was used to account for creation by the poet and playwright John Dryden, and in outline has remained the preferred cognitive process account for invention of all kinds. In the 18th century the answer to the problem of invention in art and science became "creative imagination", starting with Addison's Pleasures of the Imagination, where he wrote that imagination "has something in it like creation ... it makes additions to nature" (Engell, 1981: 36-37) and the analogies of God and spaniel remained in the background as the issue of creative genius merged into the beginnings of psychology.

Leibniz used a more complex version of this to explain theodicy, which was his attempt to reconcile evil with the goodness of God. His argument started from the premise that a world in which everything was perfect would be static and boring, and therefore unsatisfactory. To make it better God was obliged (given his good intentions) to bring in variation and change, which precludes a steady state of perfection. So God could not make it perfect, only as good as possible, and to achieve this Leibniz envisaged a combinatorial process in which God arrives at the best of all combinations. In 1740 Leibniz's model of God's creation was used by the Swiss writers Johann Bodmer and Johann Breitinger to account for poetic creation within the framework set up by Addison (Abrams 1953:276).

Work on creative imagination continued during the 19th century, and in France Alfred Binet and Theodor Ribot began experimental studies, especially in children. This cognitive work based on the association of ideas continued in Britain during the 20th century after the publication in 1906 of Ribot's Creative Imagination. Osborn's "Your Creative Power" in 1948 referred to this work, but began to use "creativity" in place of "creative imagination" (although this was not the meaning given it by Whitehead), and this opened the way for Guilford's use in 1950. Osborn had seen it as a mystery beyond scientific explanation. Guilford agreed that it was a mystery, but one that will be cleared up by science, presumably like the mystery of Life, or the Universe. Creating an extraordinary mystery that only science can solve is a smart tactic to start a bandwagon.

N-Creative

At the end of the 18th century another theoretical approach emerged, an alternative to the tacit analogy with God's creation and this new theory followed from a return to the older pagan, materialist meaning of creare, N-creative rather than G-creative. Instead of treating the mind as a closed system, isolated from the world of nature, but adding to nature by a God-like process of creation, it began to treat mind and nature as inseparable parts of an open system, drawing on earlier work by Shaftesbury and Leibniz (in his philosophy of nature rather than his theodicy). There were two aspects of this, one very sober, starting with Hume, and the other more revolutionary. The revolution came from German philosophy and then philosopher-poets, such as Schiller, Novalis and Goethe; it was brought to Britain and taken further by Coleridge and Wordsworth and their circle. For these writers human creating is like the growth of a plant, flourishing in its special environment, rather than caused by the operations

of an associative mechanism.

In a version of Hobbes' spaniel metaphor, Hume in 1739 described how "the imagination suggests its ideas, and presents them at the very instant, in which they become necessary or useful". It is as though "the whole intellectual world of ideas was at once subjected to our view" and we just pick out what is needed. But this is not what happens since only the needed ideas are present which "are thus collected by a kind of magical faculty in the soul, which, tho' it always be most perfect in the greatest geniuses . . . is however inexplicable by the utmost efforts of human understanding" (Hume, 1978:24). Hume did not believe in magic, and "a kind of magical faculty" is a way of expressing his recognition that the steps of a mechanistic account using association of ideas will not explain creative thinking, even when the spaniel metaphor is added.

The way forward was to make human invention a product, not of human mind alone, but of human mind acting on the environment, and this insight proved one of the great achievements of the Scottish Enlightenment following Hume. In 1774 in Aberdeen, Alexander Gerard described invention as the result of conjecture and experimentation, and 75 years later another Aberdonian, Alexander Bain, refined this by introducing "trial and error" (Bain, 1855). The answer to the psychological problem had begun to change, and this became a revolution in the hands of poet-philosophers in Germany and Britain around 1800. Instead of asking "How can a mechanistic mind generate new ideas?", the question became "How does a person engage with her physical and social surroundings in order to create her own world, which is a pre-requisite to human creation and invention?". This goes beyond the concept of mind as isolated mechanism, and invites a systemic approach involving organism and environment. In accord with this, the metaphor of mechanism was replaced with one of growth, and this was to some extent a return to the earlier meaning of create as *creare* in Lucretius (Nisbet, 1986; Bell, 1994). *Creare* was to bring about or have an impact through *facere*, doing or making, and the two processes are typically linked in movement or flow, exemplified by growth.

This was expressed by a number of important writers in Germany and Britain, and drew explicitly on their own experience. Amongst writers in English the most important were Wordsworth and Coleridge in Britain, who were followed by Emerson and Thoreau in the States. Coleridge provided the philosophy, struggling for an organic metaphor to replace Locke's psychic mechanism, and finding it in his account of poetry. Writing of his friend Wordsworth's poetry in their Lyrical Ballads he describes it as aiming:

to give the charm of novelty to things of every day . . . by awakening the mind's attention from the lethargy of custom, and directing it to the loveliness and the wonders of the world before us (Coleridge, 1983: 7)

And in a later work he wrote that in poetry "Nature [is] idealized through the creative power of a profound yet observant meditation", and through science poetry is "substantiated and realised" (quoted in Corrigan, 1982: 131). In the Idiot Boy, for instance, the "lethargy of custom" led many to

report disgust at Wordsworth's portrayal of a mother's love of her son, but Wordsworth's "observant meditation" went much deeper than the lethargy of custom to a human reality, the mother's love. Both poetry and science reveal reality through the power of "observant meditation", exemplified for science in Coleridge's friend Humphrey Davy. In his poetry Wordsworth began to use "creative" for a way of living in the world. In the 1805-6 edition of his long autobiographical poem, *The Prelude*, Wordsworth wrote:

*The exercise and produce of a toil,
Than analytic industry to me
More pleasing, and whose character I deem
Is more poetic as resembling more
Creative agency.*

In these ways "creative" began to refer to a particular way of acting in the world, mindful and inquiring, rather than being defined in terms of a product, the "Creation". It involved an immersion in the life around in order to bring about the world from which art and science could emerge. As Wordsworth's younger contemporary John Keats wrote in 1817, "That which is creative must create itself" (White, 2012:73). It can occur in nature herself, as when Thoreau wrote on the shape of snow crystals: "How full of the creative genius is the air in which these are generated?" (Searls, 2009: 354). But it usually referred to immersion in an activity, as in Emerson's "creative reading" in 1837 (Emerson, 1975), and Matthew Arnold's "creative criticism" of 1865 (Arnold, 1914).

This newer meaning of "creative" (N-creative) did not replace the older G-creative (which entails a specific product), and they existed alongside each other, as in "creative genius" or "creative invention" (e.g. Ward, 1899). But it is the newer conception of N-creative that led directly to the work of John Dewey and G.H. Mead in the early 20th century. Dewey wrote on both Arnold and Emerson, and is known to have been strongly influenced by Wordsworth (Gale, 2010). His reflex arc paper of 1896 (Dewey, 1896), in which stimulus and response form a feedback loop inseparable from ongoing activity (instead of a link in causal chain), provides an organic unit of the kind Coleridge struggled to find. His friend and colleague the social psychologist G.H. Mead, had contributed one of the chapters in Dewey's *Creative Intelligence* of 1917 where he had written, echoing Keats, "The individual in his experiences is continuously creating a world which becomes real through his discovery". (Mead 1964).

This word "creativity"

It was soon after Dewey's publication of *Creative Intelligence* that Whitehead used the word "creativity" in his philosophical writings while at Harvard in the 1920's, in which he attempted to replace the old clockwork mechanism of Laplace with a universe (like that of Lucretius) incorporating chance. He used the word "creativity" in his own special way, as change or "passing on" that is inherent in the world, and which he referred to as "the principle of novelty" (Whitehead, 1976: 21). He wrote that he meant it in the dictionary sense of the verb *creare*, "to bring forth, beget, produce" (Whitehead, 1976: 213), which we have glossed in

our account of N-creative as 'bring about' in nature, rather than through the power of a mind, though this is one form of it. According to Whitehead, "originality" emerges out of this; it belongs to life, as living organisms act on their environments and break away from the "line of their ancestry" (Whitehead, 1976: 104).

Whitehead's lectures at Harvard soon appeared in print. They were closely argued and difficult, but had an immediate impact, especially in the States. Dewey discussed them with Mead (Cook, 1979) and the word "Creativity" appeared in section headings in the latter's best known book (Mead 1934). Even though these headings were added by the editor (Meyer, 2005), they reflect Mead's intentions, and probably played their part in the burgeoning familiarity of the word, since the book was widely read amongst Psychologists and Sociologists. Dewey himself wrote on Whitehead (Dewey, 1937) and later started to use the word "creativity" in a way that draws on both Whitehead and William James.

In 1948 he wrote of "the life factor that varies from the previously given order, and that in varying transforms in some measure that from which it departs, even in the very act of receiving and using it. This creativity is the meaning of artistic activity - which is manifested not just in what are regarded as the fine arts, but in all forms of life that are not tied down to what is established by custom and convention. In re-creating them in its own way it brings refreshment, growth, and satisfying joy to one who participates." (Dewey, 1948). This N-Creative activity is activity that can have an impact by sustaining or changing the established order that has guided the individual and the society to which she belongs; and it is present in everyday activities, such as gardening or cooking, where there is a state of creative intelligence and a readiness for inquiry. He had already spelt this theory out in *Art as Experience* (1934).

At the same time, 1948, Alex Osborn was using "creativity" in a very different way to mean the power released by brainstorming, and two years later Guilford introduced his own G-creative version of this as the essence of creativity.

What does this mean for future research?

These two strands in history, G-creative and N-creative, define two distinct theories of creative activity that stem from two distinct meanings of "creative". N-creative is a way of living and acting in the world and it is inherent in all activity unless constrained by authority, or by self-imposed routine. It goes with a concept of intelligence based on attentive inquiry, rather than a mental power. G-creative is based on the power to generate valuable novelty, and it is distinct from intelligence, which in the IQ testing tradition is a relatively mechanical process of knowledge and problem solving.

These two theories have been living alongside each other for 75 years. They have shared the same name, and accordingly have been treated as the same. G-creative has fitted most readily into Psychology and AI, with several dissenting voices (Howe, 2001; Sawyer, 2006). N-creative has been more at home in Humanistic Psychology (Maslow, 1968; Rogers, 1954) and Education (Holbrook, 1964; Woods and Jeffrey, 1996). But there has been much mingling and overlap and this is what has caused much of the confusion. Ani-

mals are N-creative but not G-creative, so if we have only the one term "creativity" they both do and do not show creativity. Can we prise the two theories apart and decide between them as the basis for future research, or do we need both?

One important criterion is range. We have seen that G-creative is aimed at art works and inventions in the Western world, that exist, like God's creation, independently of the creator. But this is limited since "valuable novelty" and the focus on products belong to a world of profit and economic growth, to museums and concert halls. It does not readily include improvised dance, music and story-telling that plays a predominant part in earlier or non-Western traditions, and there is a kind of missionary zeal in the spread of G-creative around the world. It is clearly on the side of progress, whereas N-creative is more universal, and involves change within a tradition. Traditions themselves may be worth preserving, especially if they are necessary for creative activity to take place (Hallam and Ingold, 2007: 48, 113). But unlike G-creative there has been little attempt to formalise the assumptions underlying N-creative. How could we do this?

According to G-creative theory, the mind, like that of God, generates novel ideas which result in valuable products. In the field of AI (and explicitly its subfield computational creativity) this has been explored by trying to simulate creativity using criteria similar to those of the Turing test (Turing, 1950; Boden, 1994). More recently there have been questions raised about the feasibility of the Turing test in relation to creativity (Bedworth and Norwood 1999), and Negrotti (1991) has suggested that AI (and by extension CC) can treat the intelligence and creativity of machines as of interest in themselves, rather than as a way of understanding human intelligence or creative activity. The Turing test would then become superfluous, except as entertainment, but it would mean dropping the "value" requirement for creativity, insofar as this is measured against the evaluation of human products. Recognising this, Dorin and Korb have proposed "creativity that is independent of notions of value or appropriateness" (Dorin and Korb, 2012), and Colton and Wiggins suggest replacing "value" in the definition of creativity with "impact" (Colton and Wiggins, 2012).

This use of "impact" makes it similar to the creare of N-creative and the historical distinction between N and G-creative could be used to define this concept. We can take this definition further in terms of the "social interactions between self-motivated autonomous agents" (d'Inverno and Luck, 2013), and the proposal for "artificial creative systems composed of intrinsically motivated agents engaging in language games to interact with a shared social and cultural environment" (Saunders (2012:216). We would then be close to modelling precisely the structures implied in John Dewey's N-creative theory of art and invention. The N-creative theory is made up of two components. An autonomous agent acts on its social world by constructing, making, talking, playing music, telling jokes, inquiring, etc. These are *facere*, and *facere* brings about change or has an impact (*creare*), to a varying degree, on the world, other agents, or itself. Put together, this system, as a formal statement of Dewey's definition of creativity in 1948, would embody N-creative.

If we are right in the arguments of this paper, it is history that will have helped us to see that there are alternative ways of conceptualising what is nowadays included under the blanket term “creativity”, a term that has become so embedded in our language that even-handed debate on the matter may have become impossible.

That is why, to go further, we need to formalise the language of both theories, G-creative and N-creative, in order to prise them apart and to decide which version of creativity is most useful to us. This attempt may point the way to the design of new technologies, though it may well turn out that it will be the success or failure of such new technologies that will enable us to decide between the theories. But crucially we will have a concept of “creativity” that starts with Whitehead’s theory of creativity and change, which as Bown puts it “forces us to think about creativity as a general process that can be applied wherever new things come into existence (Bown 2012:361). This is remarkably close to what Whitehead had in mind when he introduced the word “creativity” in the 1920’s. At a more complex level, it gives rise to biological and social models of creativity as living organisms act on their environments and break away from the “line of their ancestry” (Whitehead, 1976: 104). Based on this, Dewey was working on the psychological theory of creative activity we have described shortly before he died in 1952. This would have been an extension to Psychology of the “general process” suggested by both Whitehead and Bown. But Dewey’s pioneering work was forgotten in the excitement around Guilford’s (in our view muddled and backward-looking) definition of creativity as an inner process and a measurable propensity in 1950. We end with six specific points for the future of AI research

1. Develop a profound skepticism of “creativity” as a mental entity.
2. Question why we would ever want to build artificial G-creative systems.
3. Increase our shared awareness of the N-creative work of Dewey and Whitehead that has been overshadowed by Guilford and his concept of creativity.
4. Build formal, computational models of N-creative systems and use them to build software that can support and refute these models.
5. Adopt an N-creative approach to designing systems supporting *being* in the world; enhancing and supporting human creative activity in all of its forms. (d’Inverno and McCormack, 2015).
6. Use human experience as the starting point for future system design. (Yee-King and d’Inverno, 2016)

References

Abrams, M. 1950. *The Mirror and the Lamp*. New York: Norton.

Arnold, M. 1914. *Essays by Matthew Arnold*. London: Oxford University Press.

Bain, A. 1855. *The Senses and the Intellect*. London: Longmans, Green.

Bateson, P., and Martin, P. 2013. *Play, Playfulness, Creativity and Innovation*. Cambridge University Press.

Bedworth, J., and Norwood, J. 1999. *The Turing Test is Dead*. ACM Press. 193–194.

Bell, M. 1994. *Goethe’s Naturalistic Anthropology: Man and other plants*. Oxford: Clarendon Press.

Boden, M. 1994. Precis of the creative mind: Myths and mechanisms. *Behavioral and Brain Sciences* 17(3):519–570.

Boden, M. 2009. Computer models of creativity. *AI Magazine* 2009:23–39.

Bown, O. 2012. Generative and adaptive creativity: A unified approach to creativity in nature, humans and machines. In McCormack, J., and d’Inverno, M., eds., *Computers and Creativity*, 361–381. Springer Berlin Heidelberg.

Burwick, F. 1997. *Mimesis and its Romantic Reflections*. Pennsylvania State University Press.

Cardoso, A.; Veale, A.; and Wiggins, G. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* Fall 2009:15–22.

Charlton, H. 1913. *Castelvetro’s Theory of Poetry*. Manchester: Manchester University Press.

Cohen-Cole, J. 2014. *The Open Mind: Cold war politics and the sciences of human nature*. Chicago: University of Chicago Press.

Coleridge, S. 1983. *Biographia Literaria, Volume 2*. Princeton, NJ: Princeton University Press.

Colton, S., and Wiggins, G. 2012. Computational creativity: the final frontier. 21–26.

Cook, J. 1979. Whitehead’s influence on the thought of G. H. Mead. *Transactions of the Charles S. Peirce Society* 15(2):107–131.

Corrigan, T. 1982. *Coleridge, Language, and Criticism*. Athens, GA: University of Georgia Press.

Cropley, D.; Cropley, A.; Kaufman, J.; and Runco, M. 2010. *The Dark Side of Creativity*. Cambridge University Press.

Dewey, J. 1896. The reflex arc concept in psychology. *Psychological Review* 3:357–370.

Dewey, J. 1910. *Educational Essays*. London: Blackie & Son.

Dewey, J. e. a. 1917. *Creative intelligence; essays in the pragmatic attitude*. New York: Henry Holt.

Dewey, J. 1934. *Art as Experience*. New York: Perigree Books.

Dewey, J. 1937. Whitehead’s philosophy. *The Philosophical Review* 46(2):170–177.

Dewey, J. 1948. Foreword. In Schaefer-Simmern, H., ed., *The Unfolding of Artistic Activity*. Berkeley, CA: University of California Press.

d’Inverno, M., and Luck, M. 2012. Creativity through autonomy and interaction. *Cognitive Computing* 4(3):332–346.

- d'Inverno, M., and McCormack, J. 2015. Heroic versus collaborative AI for the arts. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, 2438–2444. AAAI Press.
- Dorin, A., and Korb, K. B. 2012. Creativity refined: Bypassing the gatekeepers of appropriateness and value. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*, 339–360. Springer Berlin Heidelberg.
- Emerson, R. 1837 (19750). The american scholar. In Van Doren, M., ed., *The Portable Emerson*, 23–46. Harmondsworth, Middlesex: Penguin Books.
- Engell, J. 1981. *The Creative Imagination: Enlightenment to Romanticism*. Cambridge: Harvard University Press.
- Gale, R. 2010. *John Dewey's Quest for Unity*. Amherst, NY: Prometheus Books.
- Gerard, A. 1774. *An Essay on Genius*. London: W. Strahan, T. Cadell.
- Guilford, J. P. 1939. *General Psychology*. Princeton NJ: D Van Nostrand.
- Guilford, J. 1950. Creativity. *American Psychologist*. 5(9):444–454.
- Guilford, J. P. 1959. Three faces of intellect. *American Psychologist*. 14(8):469–479.
- Hallam, E., and Ingold, T. 2007. *Creativity and Cultural Improvisation*. Oxford: Berg.
- Hobbes, T. 1996. *Leviathan*. Cambridge: Cambridge University Press.
- Holbrook, D. 1964. *The Secret Places*. London: Methuen.
- Howe, M. 2001. *Genius Explained*. Cambridge: Cambridge University Press.
- Hudson, L. 1966. *Contrary Imaginations*. London: Methuen.
- Hume, D. 1978. *Treatise of Human Nature*. Oxford: Clarendon Press.
- Kassler, J. 1991. The paradox of power: Hobbes and stoic naturalism. In Gaukroger, S., ed., *The Uses of Antiquity: the scientific revolution and the classical tradition*, 53–78. Dordrecht: Reidel.
- Kaufman, G. 2004. *In the Beginning— Creativity*. Augsburg Fortress Publishers.
- Leff, G. 1975. *William of Ockham: The Metamorphosis of Scholastic Discourse*. Manchester: Rowman and Littlefield.
- Lucretius. 1992. *On the Nature of Things*. Cambridge, MS: Harvard University Press.
- Maslow, A. H. 1968. *Toward a Psychology of Being*. New York: Wiley.
- Mead, G. 1934. *Mind, Self and Society*. Chicago: University of Chicago Press.
- Mead, G. 1964. *Selected Writings*. Chicago: University of Chicago Press.
- Mearns, H. 1925. *Creative Power: the education of youth in the creative arts*. New York: Doubleday.
- Meyer, S. 2005. Introduction: Whitehead now. *Configurations* 13(1):1–33.
- Negrotti, M. 1991. Alternative intelligence. In Negrotti, M., ed., *Understanding the Artificial: On the Future Shape of Artificial Intelligence*, 55–75. London: Springer-Verlag.
- Nisbet, H. 1986. Lucretius in eighteenth-century germany. (with a commentary on goethe's "metamorphose der tiere"). *The Modern Language Review* 81(1):97–115.
- Osborn, A. 1948. *Your Creative Power: How to Use Imagination*. New York: C. Scribner's Sons.
- Panofsky, E. 1968. *Idea: a concept in art theory*. New York, NY: Harper & Row.
- Plato. 1977. *Timaeus and Critias*. London: Penguin Books.
- Rhodes, J. 1961. An analysis of creativity. *The Phi Delta Kappan* 42(7):305–310.
- Ribot, T. 1906. *Essay on the Creative Imagination*. Chicago: Open Court.
- Rogers, C. 1954. Toward a theory of creativity. *ETC: A Review of General Semantics* 11(4):249–260.
- Saunders, R. 2012. Towards autonomous creative systems: A computational approach. *Cognitive Computation* 4(3):216–225.
- Sawyer, R. 2006. *Explaining Creativity: The science of human innovation*. New York: Oxford University Press.
- Searls, D. 2009. *The Journal of Henry David Thoreau, 1837-1861*. New York: NYRB Classics.
- Shelley, M. 1985. *Frankenstein*. London: Penguin Books.
- Simon, H. 1995. *Explaining the Ineffable: Al on the Topics of Intuition, Insight and Inspiration*. San Francisco: Morgan Kaufmann. 939–948.
- Stein, M. 1953. Creativity and culture. *Journal of Psychology*, 36:311–322.
- Stuart-Glennie, J. 1874. The Examiner May 23rd issue.
- Teitelbaum, M. 2014. *Falling Behind?: Boom, bust and the global race for scientific talent*. Princeton, NJ: Princeton University Press.
- Tiles, M. 1984. *Science and Objectivity*. Cambridge: Cambridge University Press.
- Turing, A. 1950. Computing machinery and intelligence. *Mind* 59:433–460.
- Ward, A. 1899. *A History of English Dramatic Literature to the Death of Queen Anne, rev. ed. volume 2*. London: Macmillan.
- Wertheimer, M. 1945. *Productive Thinking*. New York: Harper.
- Whitehead, A. 1978. *Process and Reality*. New York: The Free Press.
- Wordsworth, W. 1986. *The Prelude: a parallel text*. London: Penguin Books.
- Yee-King, M., and d'Inverno, M. 2016. Experience driven design of creative systems. In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016)*. Université Pierre et Marie Curie.

VISUAL ARTS



Deep Convolutional Networks as Models of Generalization and Blending Within Visual Creativity

Graeme McCaig
Simon Fraser University,
Canada
gmccaig@sfu.ca

Steve DiPaola
Simon Fraser University,
Canada
sdipaola@sfu.ca

Liane Gabora
University of British Columbia,
Canada
liane.gabora@ubc.ca

Abstract

We examine two recent artificial intelligence (AI) based deep learning algorithms for visual blending in convolutional neural networks (Mordvintsev et al. 2015, Gatys et al. 2015). To investigate the potential value of these algorithms as tools for computational creativity research, we explain and schematize the essential aspects of the algorithms' operation and give visual examples of their output. We discuss the relationship of the two algorithms to human cognitive science theories of creativity such as conceptual blending theory and honing theory, and characterize the algorithms with respect to generation of novelty and aesthetic quality.

Introduction

It has been suggested for some time that neural networks are appealing models for aspects of creativity such as the ability to blend concepts based on their wider context and associations (Boden 2004). The recent deep learning trend (Bengio et al. 2013) utilizes multi-layer artificial neural networks, is inspired in part by theorized principles of human brain function, and has produced impressive practical results in visual processing. In particular, Deep Dream (Mordvintsev et al. 2015) and Neural Style Transfer (Gatys et al. 2015) are algorithms, based on deep-learning convolutional neural networks (CNNs) (Krizhevsky et al. 2012), that blend visual qualities from multiple source images to create a new output image. (We adopt the shortened name "Deep Style" for Neural Style Transfer—Deep Dream and Deep Style are then abbreviated DD and DS.) Since their introduction, DD and DS have evoked public attention and speculation about their level of creativity and their ability to replace human artists. Figure 1 provides an example of our results combining DD with additional painterly (stroke-based) rendering.

Here we consider how DD/DS and related CNN-based algorithms can play a role in computational creativity research as cognitively inspired mechanisms for visual blending and imagination. Both computer cognitive modeling research and results-focused computational creative systems (e.g. generative art systems) could be targets for DD/DS type algorithms. First, we examine how DD/DS fit into certain existing cognitive theories of creativity.



Figure 1. Using Deep Dream, the bird image (b) is used as a guide with source (a), thereby conceptually blending the two (c).

Visual Concept Blending and Honing Theory

Conceptual blending (Fauconnier & Turner 1998) is a proposed cognitive mechanism in which new concepts are obtained by integrating multiple pre-existing conceptual spaces. It has been suggested as an important mechanism underlying human creativity (Pereira & Cardoso 2002). Existing approaches to computational modeling of conceptual blending include symbolic AI-based systems (Besold & Plaza 2015) and the approach of Thagard & Stewart (2011) which combines neural patterns using a convolution operation (different from convolution as used in CNNs). Another recent direction in modeling conceptual blending uses a mathematical formalism based on quantum mechanics (Aerts & Gabora 2016). While the contextuality and noncompositionality of concepts makes them resistant to mathematical description, quantum formalism provides a natural spatial representation in which variables are natively context specific.

Theory and modeling of conceptual blending often includes *visual blends*, in which the novel output of a blending operation is expressed as an image. Visual blending is often framed in terms of combining high-level, verbalizable concepts (Xiao & Linkola 2015; Martins et al. 2015; Confalonieri et al. 2015). However, the strength of DD/DS lies in their ability to combine lower-level yet abstract im-

age qualities such as shapes, textures and arrangements, extracting such qualities from input images without explicit labelling. Lower-level image-quality blending is likely a component of visual *imagination* (Richardson 2015; Heath et al. 2015), and fits the idea of preconceptual creativity discussed in (Takala 2015). Thus, we suggest that DD/DS are promising as computational tools for exploring preconceptual visual blending and imagination, which are relevant to creativity and art creation.

Another theoretical framework for cognitive creativity is honing theory (Gabora 2005). Honing theory predicts that creativity involves the merging and interference of memory items resulting in a single cognitive structure that is *ill-defined*, and can be said to exist in a state of potentiality, and which can be formally described as a superposition state. The idea becomes increasingly well-defined, and transforms from potential to actual through interaction with internally and externally generated contexts. The idea could actualize in different ways depending on the contexts the idea interacts with, or perspectives it is viewed from. Innovative concept combinations are thought to take place in an *associative* mode of thought, in which more features of the object of thought are held in mind, and thus there are more associative routes available to items that have previously gone unnoticed (Gabora & Ranjan, 2013). The transition between a tightly focused *analytic* mode of thought and a widely focused associative mode of thought typically takes place many times in the creative process, and facility in making this transition known as *contextual focus* is important to creative ability (DiPaola and Gabora 2009).

As we will discuss and illustrate, DD and DS share qualities with mechanisms proposed in honing theory such as:

- A repeated dual-phase search process in which images are analyzed from low-level to a collection of abstract, higher-level perspectives (features) – resemblance or emphasis is found at this higher level, and then transferred back to the pixel level
- The interaction of visual concepts in working memory with a previously learned network of visual features built up over time
- Identifying points of resonance or visual metaphors across diverse images based on points of similarity in abstract feature association space

Deep Learning Convolutional Neural Nets

Deep learning is a collection of network-based machine learning methods that are notable for their power and breakthrough performance in tasks such as object recognition (Krizhevsky et al. 2012), as well as their similarities to aspects of human vision and brain function (DiCarlo et al. 2012). When network training is complete, running novel input data through the network models perception, and some network types are capable of using feedback connections to create novel data generalized from what has been learned (Salakhutdinov and Hinton 2009). Deep generative models have been proposed as useful for cognitive modelling (Zorzi et al. 2013). A deep belief net is used as a perception module in Spaun Cognitive Architecture (Stewart

et al. 2012), and a deep Boltzmann machine used as a model for Charles Bonnet Syndrome (hallucination) by Reichert et al. (2013).

Machine learning has traditionally depended on hand-engineering of features. Recent techniques focus on representation learning, which strives to automatically "extract and organize relevant information from the data", a step toward human-like AI (Bengio et al 2013). Deep learning techniques, which compose multiple non-linear transformations of the data, have become an important focus in representation learning. Much of the power of deep networks comes from their ability to naturally represent abstraction. Abstraction occurs when a certain symbol or encoding element stands for a broader range of specific instantiations. The nested, hierarchical structure of abstraction maps directly to the connection structure of a deep net.

The convolutional neural network (CNN) (LeCun et al. 1998) is a deep feedforward neural net architecture usually trained with backpropagation. The CNN architecture aids generalization, efficient training, and invariance to input distortions by incorporating reasonable assumptions about the input image domain through mechanisms of local receptive fields, shared weights, and sub-sampling. "Core object recognition" in the human brain is thought (DiCarlo et al. 2012) to use an alternating structure of selectivity and tolerance transformations, similar to convolutional nets.

Visual Blend CNNs: Deep Dream & Style

Using our code and scripts based on these algorithms as well as a simple selfie source of one of the authors we explore and explain the principles of Deep Dream (Mordyintsev et al. 2015), focusing on guide-image mode, and Deep Style (Gatys et al 2015). DD was developed by Google researchers and introduced via a blog-post and open source code (github.com/google/deepdream). The purposes of DD are not only to "check what [a] network learned during training" but also to provide "a new way to remix visual concepts—or perhaps even shed a little light on the roots of the creative process in general" (Mordyintsev et al 2015). DS grew out of research in texture-transfer and texture-synthesis—the authors view the algorithm as a way to separate out the "content" from one image and the "style" from another, fusing them in a novel image. These two algorithms both use pre-trained CNNs to generate a transformed version of a source image, emphasizing certain semantic and/or stylistic qualities.

The two algorithms are similar in many respects, as Figure 2 illustrates. Each begins with a style-source image, which is propagated from the lowest (pixel) layer to a selected set of higher layer(s)—the higher layers give a more generalized encoding of the image in terms of abstract features. This encoding is stored as a guide-style tensor. In DS, the guide-style tensor is accompanied by a guide-content tensor, found by propagating the content-source image through the network (to a layer higher than that of the guide tensor). The lack of guide-content tensor in DD means that DD is more divergent, wandering farther from the source image over successive iterations into patterns

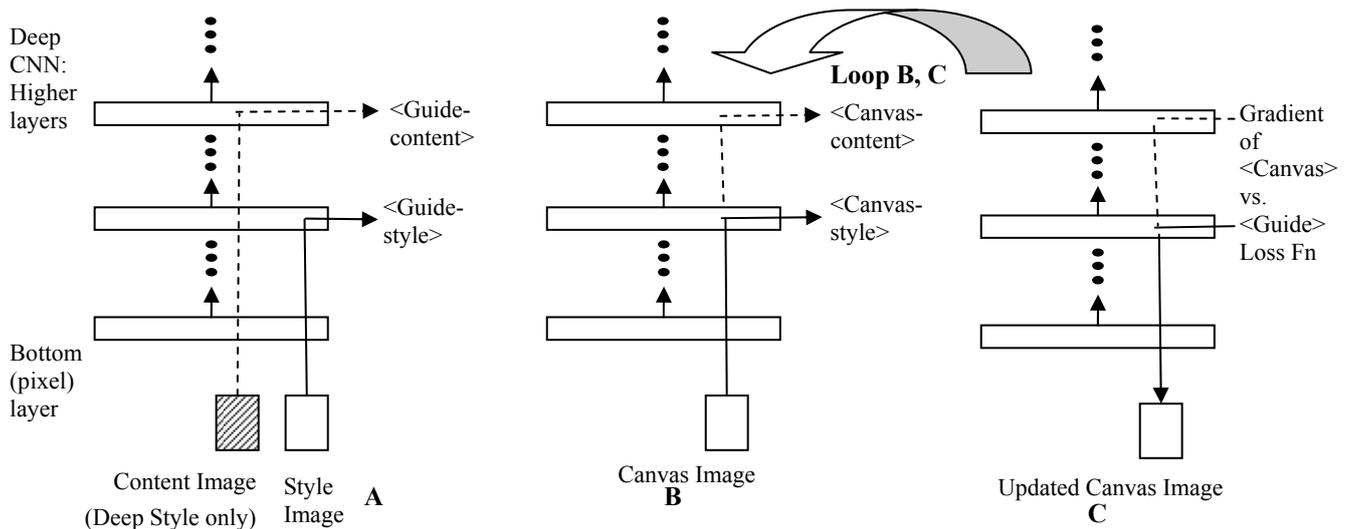


Figure 2. Schema illustrating the operation of Deep Dream and Deep Style algorithms. Dashed lines apply only to Deep Style.

and associations that depend more on the network’s learned biases and less on what was originally present.

At this point, the algorithms initialize the set of pixels that will be gradually transformed into the output image (we call this set the canvas image due to its dynamically updated nature). In DD, the canvas image is set equal to the content-source image, while in DS the canvas image is initialized as random noise. The canvas image is propagated up through the network to the same layers as the guide-style tensor and guide-content tensor, respectively, yielding the canvas-style tensor and the canvas-content tensor.

A loss function is defined to measure the similarity of the canvas-style tensor vs. guide-style tensor (and also canvas-content tensor vs. guide-content tensor for DS). From this loss function a gradient is found, indicating how the canvas tensors should be incrementally changed to be more similar to the guide tensors. This gradient is back-propagated to the lowest layer, and a small step in the desired direction is applied as a change to pixel values. Repeating the cycle of (update pixels, propagate forward, find gradient, propagate downward) constitutes a gradient ascent process.

The nature of the guide tensors and loss functions determines which visual attributes are drawn from the source and guide images in a blend. DD maximizes a dot product between feature vectors from the canvas activation and the feature vectors taken from *best-matching locations* in the guide activation. Thus a DD image will tend to pick up on certain types of shape and texture, found within the guide image, that bear similarity to its own shapes and textures.

In DS, the guide-style tensors are computed as Gram matrices, a set of correlations among the features within each layer. Compared to the dot-product method this approach seems to more faithfully capture the look of specific shapes and textures from the guide image, particularly with respect to color schemes. Some generalization does take place, allowing regions of the canvas to be *similar* to the guide rather than slavishly copying patches. This use of

Gram matrices also encourages the canvas as a whole to capture different aspects of the guide image rather than fixate on a particular visual component as DD often does. DS combines the Gram matrix approach with a simpler vector-similarity measure used for the content-guide, ensuring that the recognizable content (spatial layout and high-level features) of the source image is maintained.

The two algorithms can be applied to various CNN architectures and training sets. In this paper we employ GoogLeNet (Szegedy et al. 2015) for DD and VGG (Simonyan & Zisserman 2014) for DS, as did the algorithms’ designers. The network weights come from training on the ImageNet dataset (except in the experiment illustrated in Figure 3f, using a network trained on the Comprehensive Cars dataset). Our implementations of DD and DS are built on the Caffe CNN library (Jia et al. 2014).

Deep Dream can alternatively be run in a no-guide-image mode (what we might call “free hallucination”). In this mode, the quantity optimized is the L2-norm of activation for one selected network layer (meaning whichever of that layer’s nodes are most strongly activated will tend to become more so).

Generalization and Two-Phase Aspects

As Figure 2 helped to illustrate, DD and DS incorporate mechanisms proposed to play an important role in creative cognition. Crucially, to create a blend of two images, these algorithms first generalize each image by propagating it through a deep CNN and representing it according to the resulting tensor encoding at a certain network layer(s). Depending on the height and type of the network layer, the given encoding analyzes the image according to a particular set of visual features. Much as in the parable of the blind men who each describe an elephant in a different way, different layers “see the image” in different ways, the visual blend depends on points of similarity between guide and source, as viewed from the perspective of a certain network layer encoding. In turn, the nature of a layer en-

coding depends on both the network architecture and the original training data, which caused features to be developed as a form of long-term memory.

To enhance similarity between the source image and the guide, the algorithms use a reiterative two-phase creative process of alternating divergence and convergence; similarities found at a high/abstract level are manifested back at the pixel level.

Algorithm Input/Output Examples

This section provides examples of images processed by the DD and DS algorithms, and discusses them in terms of creative cognitive mechanisms.

Figure 3 presents results obtained using Deep Dream in no-guide-image mode. This illustrates the type of visual features according to which different network layers encode and interpret the image. We note that such visualization does not capture the entire range of visual shapes/textures potentially encoded by a given layer. Rather it tends to display a kind of layer bias—a strong regime of activation (in the L2-norm sense) into which the layer tends to “find its way” using gradient ascent search.

Certain characteristics of the output of Deep Dream (guided or not) as well as Deep Style are particularly evident in Figure 3. Firstly, the algorithm does not merely superimpose layer-specific features in a random way over the image; rather features tend to be emphasized and grown starting from those image regions that already contain said features. For example, a layer that emphasizes circle and arc shapes tends to place them in pre-existing arc-shaped parts of the image, such as the curved orbital region around a subject’s eye. On the other hand, if the algorithm is run for many iterations, all image regions will eventually be forced in the direction of a high-activating feature, essentially making something out of nothing.

Another striking aspect of the DD output is that the textures and shapes convey a sense of completion and good continuation, or flow. For example, in Figure 3b, we see a repeated plate or petal pattern in which most of the plates are similar size and shape, not overlapping or being interrupted. This arises from the type of optimization performed by the search process; the total activation of a layer is more enhanced when neighboring features work together without overlapping or disrupting each other.

Two notable “clichéd” aspects of DD deserve comment:

1. Being designed as a bottom-up discrimination network, GoogLeNet discards much of the data about tonic color of regions, but retains color-contrast near edges. When feature detection is optimized, contrasting color bands tend to be created while losing much of the source color information (color remains relevant for objects such as fire trucks where it is an important identity cue).

2. The ImageNet training data contains a bias towards animal types as a large portion of the 1k labelled categories, with a particular emphasis on fine-grained distinction of dog breeds. Hence, a large portion of the network’s capacity has attuned to the task of detecting dog faces and parts. This creates a bias in the encoding space wherein



Figure 3. Comparing no-guide output of Deep Dream for low to high network layers (a-e), and alternate training set (f).

shapes and patterns are likely to be treated as relevant to dog features; hence dog features emerge. Figure 3f shows results obtained using a network trained on car images (CompCars database); indeed, car features emerge.

We now examine results from our DD implementation when a guide image is used, exploring different guide images and sources along with different layers used as target for guide- and canvas- tensors. Many of our results use as input a simple selfie portrait from a front facing iPad (800x600 pixel resolution) in office lighting to show these results from our systems can be obtained from simple non-professional source material.

Figure 4 provides a sample set of input/output images, showing how the algorithm generalizes visual features depending on the network layer. In 4b, butterfly features are represented as grid-like patterns and black/orange/yellow colors, while in 4c more complete shapes such as wings and limbs/antennae emerge. These particular shapes are not directly visible in the butterfly image nor in the face image, but emerge when generalizing from both. In Figures 4d and 4e (obtained using a different source with the same guide) the wings vary in shape depending on the source image as well. These results demonstrate that the wing shape emerges from the particular combination of input images without being explicitly present in either.

Figure 5 explores the difference in source vs. guide image roles by using a certain two images in alternating roles. Attributes generalized from the fire truck include rectangular and curved shapes (and a reddish tinge) at the lower level, while many truck-like components emerge at a higher level. The features abstracted from the face seem to be focused on hair-like patterns, as well as the transformation of the truck’s windows into orbital region- or eye-like curves. Figure 6 shows the different effect of DS: DS captures more specific-looking fragments from the guide image (giving a collaged look), more faithful colors, and a more balanced distribution of colors and shapes from

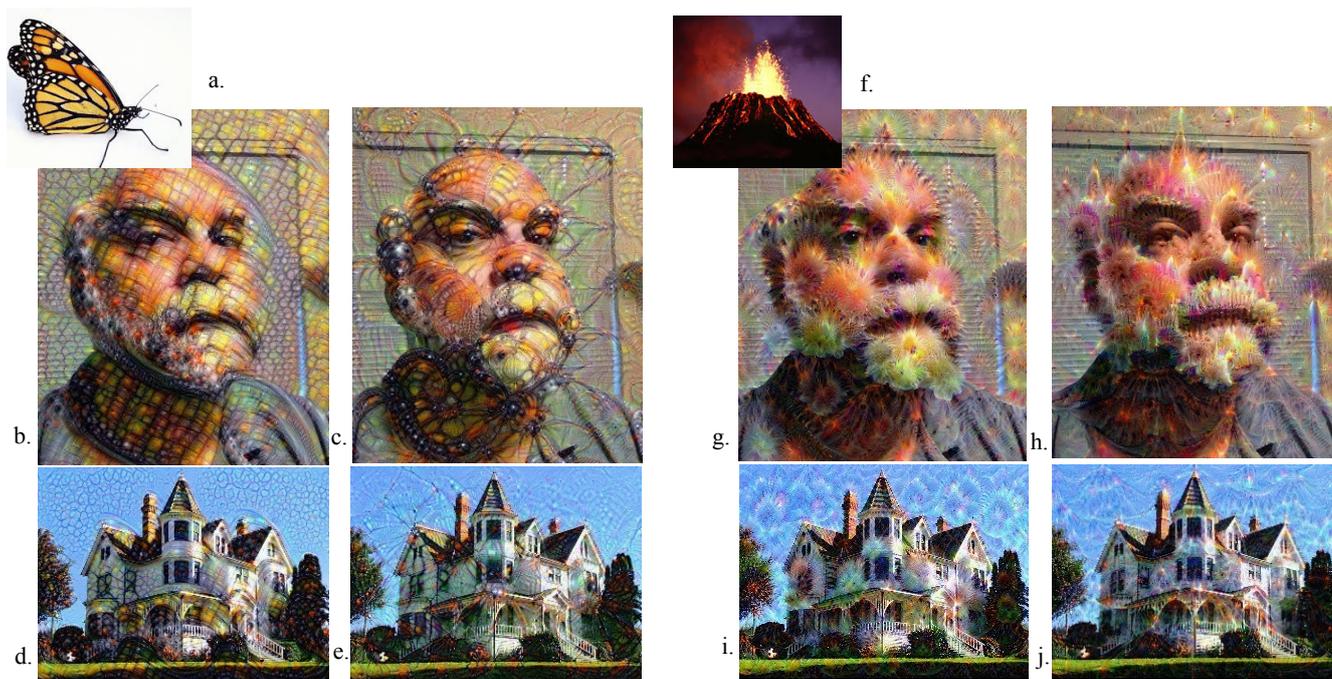


Figure 4. Deep Dream examples using different source and guide images and net layers. Images (b-e) use guide (a), images (g-j) use guide (f). Images (b,d,g,i) use a lower net-layer parameter while (c,e,h,j) use a higher net-layer parameter. Butterfly image by Ano Lobb (Flickr, CC BY 2.0). Original house image by Jan Tik (Flickr, CC BY 2.0). Volcano image courtesy of US Geological Survey.

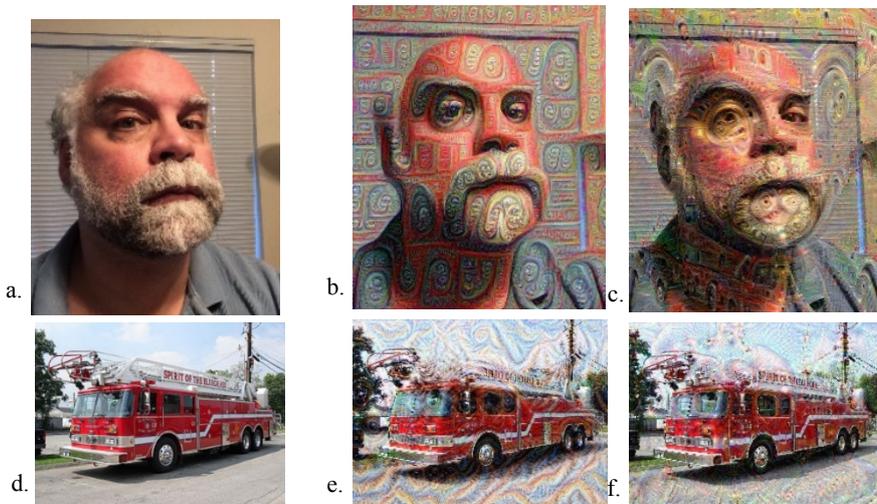


Figure 5. Deep Dream. Top row: face (a) as source with fire engine (d) guide; bottom row: fire engine source with face guide. (b,e) use lower net-layer, (c,f) use higher net-layer.

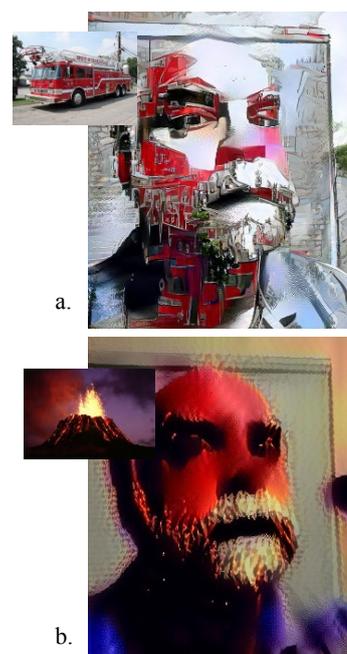


Figure 6. Deep Style using face content and engine, volcano style (a, b).

across all parts of the guide. Figure 7 shows two guide/source combinations that produced unexpected results, illustrating how the learned background knowledge of the network plays a key role in how guide and source image features are interpreted and hence generalized/reapplied. In Figure 7b it is likely the network has not been specifically trained on many angular-origami-pattern images, yet it “understands” the guide image in terms of a

range of patterns and particularly crease-like dark diagonal lines. In Fig 7d, the network features do a good job of representing several bee-like aspects: yellowness, fuzziness, dark eyes and dangling legs. Yet, the algorithm also falls prey to the aforementioned training set bias of seeing dog faces, creating a fanciful 3-way dog/bee/face hybrid. Due to the fact that DS adheres closely to the look of specific guide image fragments, it is quite successful at painting

style transfer, i.e. applying the colorist and brush stroking style from a specific painting to a new source image to imitate, e.g., Rembrandt. Figure 8 shows two examples.

As part of our first extension of the DD and DS algorithms, we aim to situate DD/DS within the wider scope of an artificial painter system. In this type of system, DD/DS or similar systems can play the role of the artist's perception and imagination (for abstraction, narrative, and emphasis), while a further artistic painting algorithm models the phase that occurs in manifesting imagery onto the canvas and dialoging with the artist's materials. Eventually it would make sense to have a cyclical interaction between the perception/imagination phase and the stroke-painting phase. In this first attempt, we apply the DD/DS module to the source photo first, followed by a painting phase.

Figure 9 shows a DD image before and after treatment with the ePainterly cognitively inspired painting system (in digital format, zoom in to appreciate detail). This ePainterly system, an extension to cognitive painting system, Painterly (DiPaola 2009), models the cognitive processes of artists using algorithmic, particle system and noise modules to generate artistic color palettes, stroking and style techniques. It is in the stroke-based rendering subclass of non-photorealistic rendering (NPR) and is used as the final part of the process to realize the internal DD/DS models. In this example, aesthetic advantages include reducing some artifacting of DD output via cohesive stroke-based clustering as well as a better distributed color space.

Evaluating Novelty and Aesthetic Value

We can also analyze DD and DS in terms of other ideas about computational creativity. We will examine to what extent these algorithms are able to pursue generated outputs having novelty and value (widely accepted as the defining characteristics of creative production).

We first inquire whether DD/DS make explicit autonomous evaluations of the novelty and value of their outputs. This question can be viewed as important for creativity: for example, Jennings (2010) includes such autonomous evaluation as a necessary condition for creative autonomy. (He also includes the ability to *change* evaluative standards, a condition not met by DD/DS on their own). In terms of novelty, when used as standalone systems, neither DD nor DS maintain or optimize explicit measures of novelty as an image is produced. In terms of value, the situation is more favorable: DD and DS isolate and maximize subsets of the network features evoked by each input image, resulting in the maintenance or enhancement of certain aspect image qualities (or whole levels of abstraction) at the expense of other qualities. This process can be construed as the computation of one or more aesthetic value metrics. It bears a resemblance to neuroaesthetic principles of art, such as Zeki's (2001) notions of highlighting/stimulating discrete portions of visual processing and translating the brain's abstractions on to the canvas or Ramachandran and Hirstein's (1999) laws of peak-shift and isolation.

Going beyond the ability of DD/DS to explicitly evaluate and optimize novelty and value, we can further ask

whether DD/DS at least tend to generate outputs with high novelty and value. Informally, we do observe that the range of outputs generated by these systems often strike us as surprising while also being visually pleasing. The viewer may be surprised not only by the form of an unexpected visual concept being "imagined" into the original image (in the case of free hallucination with DD) but in the form of an unexpected manner of visual resemblance between selected inputs (e.g. Figure 7). Perhaps the network's support of a large number of multi-level visual features creates a rich space of combinations such that the optima found by DD/DS are not the same as the human viewer anticipates.

In Ritchie's (2007) empirical framework for assessing computational creativity, it is suggested that measures more fine-grained or primitive than novelty are dissimilarity from the "inspiring set" and typicality (relative to the target art form or genre). For DD and DS, we take the inspiring set to include both the original neural network training image set and the current input image(s) to the algorithm. It seems obvious that DD/DS are not simply replicating or near-replicating any of the training set images. By design, DD and DS outputs do bear visual similarity to the input images, but our general impression is that the resemblance is not so slavish as to exclude creativity. We furthermore observe a tendency for DD to find a more abstract or remote similarity with the style-guide image compared to DS, which aligns well with the fact that DS uses a more informationally rich optimization goal for style.

Ritchie regards typicality as a double-edged sword: on one hand, it is an achievement for a computer to generate successful examples of a style, but on the other hand, high typicality suggests low novelty. For DD, the comparison set for typicality is not obvious ("contemporary art" would be one choice), while DS will tend to be compared to the artist or genre of the style-guide image. Thus, to the extent that DS seems often to do a good job of mimicking an artist's style (e.g. Figure 8), Ritchie's approach might lead us to characterize it as impressive by one standard yet prevented from being creative at the highest level. Additional considerations from Ritchie (2007) regarding repetition point to other limitations on DD/DS's level of creativity. Repetition in the output arises when using DD/DS multiple times with the same inputs and parameter settings.

Finally, it is worth considering that DD/DS algorithms may be particularly amenable to future extensions and modifications that would enhance their ability to internally search for novelty/value. Regarding novelty, the vector spaces formed by node activations can lead to distance measures that are closer to human-perceived visual similarity compared to raw-pixel-based measures. Such measures could be combined with storage of images and data clustering techniques to estimate the novelty of a particular generated image compared to training images or compared to a certain corpus (e.g., art of a certain genre). Regarding aesthetic value, ideas from information-based aesthetic theories (Rigau et al. 2008) such as compressibility could be applied as additional optimization constraints, using the node vectors as a basis.

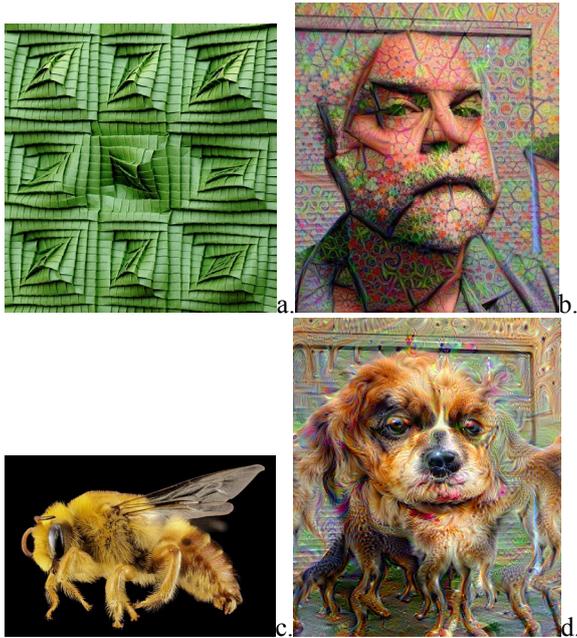


Figure 7. Deep Dream—surprising interpretations of image attributes by network. Image (a) by Goran Konjevod (Flickr, CC BY-NC 2.0)

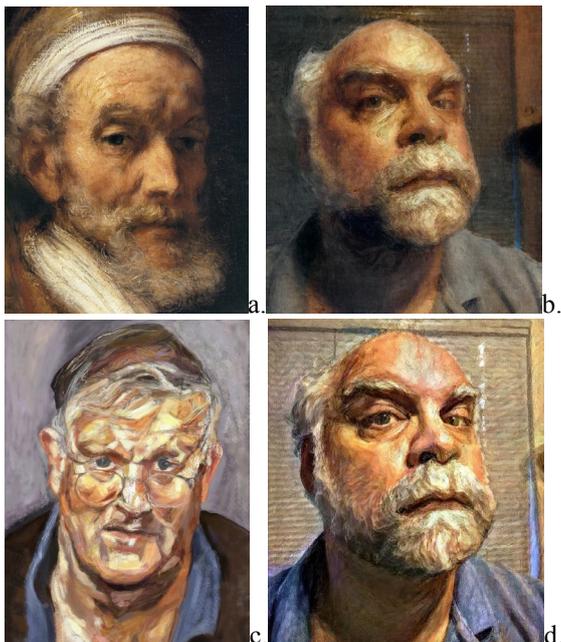


Figure 8. Deep Style using Rembrandt (a,b) and Freud (c,d) paintings as style guides.

Conclusion and Future Directions

We suggest that the use of deep neural networks (with accompanying search/optimization algorithms) to produce creative visual blends and artifacts has entered a new and promising phase, both for models of creativity and for practical art-generating systems. Next steps include:

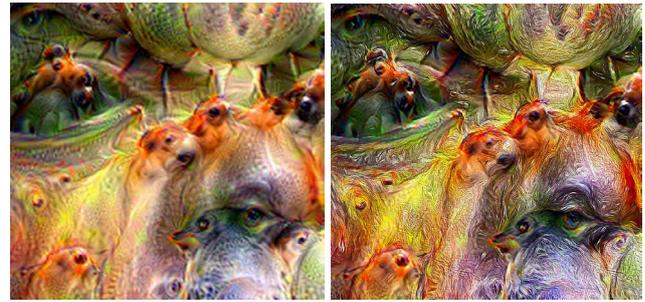


Figure 9. Deep Dream example (zoomed-in crop), before (l) and after (r) cognitive stroke-painting simulation.

1. Addressing flaws and limitations in current network architectures and training sets—some current work aims to increase network ability to generate more realistic images, avoiding the current visible artifacts (Denton et al. 2015). Richer or purpose-built training data should help get around issues of bias such as the “dog faces everywhere” effect.

2. Embedding deep neural nets as perceptual modules in larger architectures for cognitive creativity and art—other modules to add include emotional expression and planning/interaction with art materials. Examples of this direction include (Stewart et al. 2012; Augello et al. 2013).

3. Linking the CNN operation to different training paradigms or text-understanding modules—this could allow the networks to more explicitly represent concepts beyond noun-like object semantics and parts, for example allowing image generation based directly on adjectives at a visual or emotional level (e.g. “jagged”, “joyful”, or “bold”).

4. Exploring augmentations to DD, DS for novelty and aesthetic evaluation as mentioned in the previous section.

Acknowledgements

This work was supported by the National Sciences and Engineering Research Council of Canada (NSERC). Thanks to Daniel McVeigh and Jonathan Waldie for their assistance generating image examples. Our DS implementation is based on: <https://github.com/fzliu/style-transfer>.

References

- Aerts, D., Broekaert, J., Gabora, L. & Sozzo, S. 2016. Generalizing prototype theory: A formal quantum framework. *Frontiers in Psychology* 7.
- Augello, A., Infantino, I., Pilato, G., Rizzo, R., & Vella, F. 2013. Introducing a creative process on a cognitive architecture. *Bio. Inspired Cognitive Architectures* 6:131–139.
- Bengio, Y., Courville, A., & Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Trans. on Pattern Analysis and Machine Intell.* 35(8):1798–1828.

- Besold, T. R., & Plaza, E. 2015. Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *Proceedings of the Sixth International Conference on Computational Creativity*, 150-157.
- Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. Psychology Press.
- Confalonieri, R., Corneli, J., Pease, A., Plaza, E. and Schorlemmer, M. 2015. Using argumentation to evaluate concept blends in combinatorial creativity. In *Proc. of the Sixth Int. Conf. on Computational Creativity*, 174-181.
- Denton, E. L., Chintala, S., & Fergus, R. 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, 1486-1494.
- DiCarlo, J., Zoccolan, D., & Rust, N. 2012. How does the brain solve visual object recognition? *Neuron* 73(3):415-434.
- DiPaola, S. 2009. Exploring a parameterised portrait painting space. *Int. Journal of Arts & Technology* 2(1):82-93.
- DiPaola, S., & Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming & Evolvable Machines* 10:97-110.
- Fauconnier, G., & Turner, M. 1998. Conceptual integration networks. *Cognitive Science* 22(2):133-187.
- Gabora, L. 2005. Creative thought as a non-Darwinian evolutionary process. *J. Creative Behav.* 39(4):262-283.
- Gabora, L., & Ranjan, A. 2013. How insight emerges in distributed, content-addressable memory. In *Neuroscience of Creativity*. Cambridge, MA: MIT Press. 19-43.
- Gatys, L. A., Ecker, A. S., & Bethge, M. 2015. A neural algorithm of artistic style. arXiv:1508.06576.
- Heath, D., Dennis, A. & Ventura, D., 2015. Imagining imagination: A computational framework using associative memory models and vector space models. In *Proc. of the Sixth Int. Conf. on Computational Creativity*, 244-251.
- Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4): 489-501.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM International Conf. on Multimedia*, 675-678.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097-1105.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278-2324.
- Martins, P., Urbancic, T., Pollak, S., Lavrac, N., & Cardoso, A. 2015. The Good, the Bad, and the AHA! Blends. In *Proc. Sixth Int. Conf. on Comp. Creativity*, 166-173.
- Mordvintsev, A., Olah, C., Tyka, M. 2015. Inceptionism: Going deeper into neural networks. URL: googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html
- Pereira, F. C., & Cardoso, A. 2002. Conceptual blending and the quest for the holy creative process. In *Proc. ASIB'02 Symposium for Creativity in Arts and Science*.
- Ramachandran, V. S., & Hirstein, W. 1999. The science of art: A neurological theory of aesthetic experience. *Journal of Consciousness Studies* 6(6-7):15-51.
- Richardson, A. 2015. Imagination: Literary and cognitive intersections. In *Oxford Handbook of Cognitive Literary Studies*. Oxford University Press. 225-245.
- Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67-99.
- Reichert, D. P., Seriès, P., & Storkey, A. J. 2013. Charles bonnet syndrome: evidence for a generative model in the cortex? *PLoS Computational Biology* 9(7):e1003134.
- Riesenhuber, M., & Poggio, T. 1999. Hierarchical models of object recognition in cortex. *Nature Neuroscience* 2(11):1019-1025.
- Rigau, J., Feixas, M., & Sbert, M. 2008. Informational aesthetics measures. *IEEE Computer Graphics and Applications* 28(2):24-34.
- Salakhutdinov, R., & Hinton, G. E. 2009. Deep Boltzmann machines. In *Proc. of the International Conference on Artificial Intelligence and Statistics*, Vol. 5, 448-455.
- Simonyan, K., & Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.
- Stewart, T. C., Choo, F.-X., & Eliasmith, C. 2012. Spaun: A perception-cognition-action model using spiking neurons. In *Proc. Conf. of the Cognitive Science Society*, 1018-1023.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. 2015. Going deeper with convolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1-9.
- Takala, T. 2015. Preconceptual Creativity. In *Proc. of the Sixth Int. Conf. on Computational Creativity*, 252-259.
- Thagard, P., & Stewart, T. C. 2011. The AHA! experience: Creativity through emergent binding in neural networks. *Cognitive Science* 35(1):1-33.
- Xiao, P., & Linkola, S. 2015. Vismantic: Meaning-making with Images. In *Proceedings of the Sixth International Conference on Computational Creativity*, 158-165.
- Zeki, S. 2001. Essays on science and society: Artistic creativity and the brain. *Science* 293(5527):51-52.
- Zorzi, M., Testolin, A., & Stoianov, I. P. 2013. Modeling language and cognition with deep unsupervised learning: a tutorial overview. *Frontiers in Psychology* 4.

X-Faces: The eXploit Is Out There

João Correia and Tiago Martins and Pedro Martins and Penousal Machado

CISUC, Department of Informatics Engineering,
University of Coimbra,
3030 Coimbra, Portugal
{jncor, tiagofm, pjmm, machado}@dei.uc.pt

Abstract

In the combinatorial form of creativity novel ideas are produced through unfamiliar combinations of familiar ideas. We explore this type of creativity in the scope of Data Augmentation applied to Face Detection. Typically, the creation of face detectors requires the construction of datasets of examples to train, test, and validate a classifier, which is a troublesome task. We propose a Data Augmentation technique to autonomously generate new frontal faces out of existing ones. The elementary parts of the faces are recombined using Evolutionary Computation and Computer Vision techniques. The key novel contributions include: (i) an approach capable of automatically creating face alternatives; (ii) the creation and usage of computational curators to automatically select individuals from the evolutionary process; and (iii) an experimentation with the interplay between Data Augmentation and serendipity. The system tends to create a wide variety of unexpected faces that exploit the vulnerabilities of face detectors. The overall results suggest that our approach is a viable Data Augmentation approach in the field of Face Detection.

Introduction

Face Detection (FD) systems are being thoroughly researched due to their wide range of applications, including entertainment services, social networks, search engines, and security systems (Yang, Kriegman, and Ahuja 2002; Zhang and Zhang 2010). Typically, such detectors employ classifiers that are created using example-based learning techniques. In this way, the dataset plays a key role not only for attaining competitive performances but also for assessing the strengths and shortcomings of the classifier. This means that the creation of adequate datasets for training, testing, and validation of the classifier becomes a crucial process.

The creation of a face classifier requires the construction of a dataset composed of positive and negative examples. A positive example consists of an image containing at least one face and the respective bounds; a negative example consists of an image that does not contain faces. Collecting both types of examples presents troublesome endeavours as the annotation of several images containing faces is a tiresome task, and gathering negative examples is far from being a straightforward task due to the inexistence of well-defined guidelines (Sung and Poggio 1998;

Machado, Correia, and Romero 2012b). For these reasons, after gathering a number of examples, Data Augmentation (DA) techniques are usually used to expand the dataset (Chen et al. 2007).

We propose an approach that aims the expansion of the dataset of positive examples through the generation of new examples out of the existing ones. The idea is to recombine the elementary parts of frontal faces, i.e. mouths, noses, eyes and eyebrows, using Computer Vision (CV) techniques. A Genetic Algorithm (GA) is used to automatically recombine these parts and this way create new faces that are different from the original ones. To guide the evolutionary process we resort to an automatic fitness assignment scheme that employs a classifier (Machado, Correia, and Romero 2012a; Correia et al. 2013). The evolutionary engine is designed so the activation response of the classifier is minimised. As such, the evolutionary process tends to evolve *exploits* of the classifier, i.e. faces that are no longer considered as faces by it (see, e.g., figure 1). We have also implemented computational curators to automatically select examples that are evolved during the evolutionary process. We consider the results interesting, diversified, and sometimes peculiar.

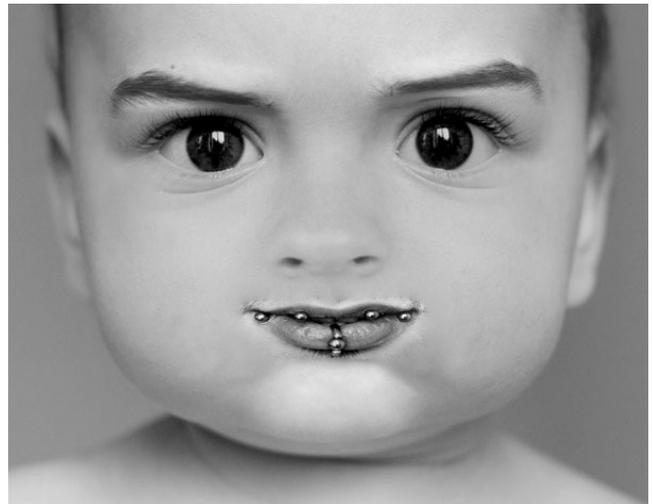


Figure 1: Face evolved by the system that is not classified as so by the classifier.

Our work is motivated by the combinatorial form of creativity, where novel ideas are produced through unfamiliar combinations of familiar ideas (Boden 2009) and by the idea of recognising new invented artifacts even when we are not specifically searching for them (Pease et al. 2013; Schorlemmer et al. 2014). Thus, the main contribution presented herein is an approach capable of automatically generate positive examples out of existing ones. Other contributions include: (i) the combination of CV and Evolutionary Computation (EC) techniques to automatically generate faces; (ii) the analysis of the results evolved using different classifiers trained under different conditions; (iii) the usage of curators to select individuals from the evolutionary process; and (iv) an experimentation with the interplay between DA and serendipity.

The remainder of this paper is organised as follows. We begin by summarising the related work. We proceed to the thorough explanation of the proposed approach. Then, the experimental setup is described and the results are analysed. Finally, conclusions and directions for future work are presented.

Related Work

In this section we analyse contributions on the topic of creation of human faces using EC techniques. We separate the contributions into two groups: the creation of faces out of existing ones; and the creation of faces from scratch.

Johnston and Caldwell (1997) have developed a criminal sketch artist by implementing a GA to recombine regions of existing face images. With the objective of improving a FD system, Chen, Chen, and Gao (2004) have employed a self-adaptive GA to manipulate face images using image segmentation techniques and photometric transformations. The transformations applied in these two approaches can be destructive and for this reason invalid solutions may be created. Frowd, Hancock, and Carson (2004) have used a GA in combination with Principal Component Analysis (PCA) and eigen vectors to create an Interactive Evolutionary Computation (IEC) system aimed to evolve human faces. Limitations of this approach include its dependence on the user guidance and the small variety of results that it can generate.

The second group of contributions include approaches that use a general purpose evolutionary art tool to evolve faces from scratch. Examples of such approaches include the contributions of DiPaola and Gabora (2009) and Ventrella (2010) where the similarity to a target image is used to assign fitness. Both approaches suffer from the limitations of IEC and the solutions tend to be too abstract or too similar to the target image. In previous work, we have also used a general purpose evolutionary art tool and an automatic fitness assignment scheme to evolve images from scratch, including figurative images such as human faces and ambiguous images (Machado, Correia, and Romero 2012a; Correia et al. 2013; Machado et al. 2015). This approach tends to evolve abstract imagery and for this reason it does not guarantee that, from a subjective point of view, the results resemble a face.

We have extracted some observations from our analysis of the shortcomings of the related work that guided the design

of our approach: (i) it should be able to explore the search space in an automatic way, avoiding the user fatigue; (ii) it should guarantee the generation of valid faces; and (iii) it should promote the creation of faces considerably different from the ones contained in the dataset.

The Approach

The proposed approach is part of the Evolutionary Framework for Classifier assessment and Improvement (EFECTIVE) (Machado, Correia, and Romero 2012b; Correia et al. 2013) and is composed of three different modules: an annotation tool, an evolutionary engine, and a classifier. Thus, the approach comprises the following steps: (i) the annotation of training examples by indicating the bounds of the faces and their parts; (ii) the training of a classifier with these examples; and (iii) the automatic evolution of new examples using the classifier to assign fitness. The three different modules are detailed in the following subsections.

Annotation tool

We develop a general-purpose image annotation tool (see figure 2). It allows the user to annotate objects present on images. One can annotate an object by positioning a sequence of points along its contour and by choosing the corresponding category. New categories can be added at any moment. The annotations created by the user are automatically saved in output files, more particularly in one eXtensible Markup Language (XML) file for each image and in one text file for each object category. The tool also exports the mask of each annotated object. When one opens a folder with images, the tool loads the corresponding annotations saved in files if they exist.

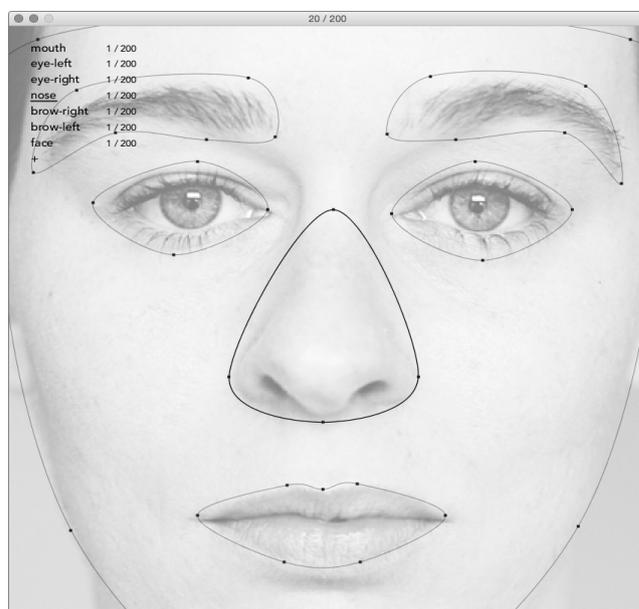


Figure 2: Screenshot of the annotation tool. A demo video can be seen at <http://cdv.dei.uc.pt/2016/x-faces-annotation-tool.mov>.

We use this tool to annotate the elementary parts of faces on a set of images. In this work, each face is annotated by indicating the bounds of its left eye, right eye, left eyebrow, right eyebrow, nose, mouth, as well as the bounds of the face itself.

Classifier

We train a cascade classifier to detect frontal faces based on the work of Viola and Jones(2001). It uses a set of small features in combination with a variant of Adaboost (Freund and Schapire 1995) to attain an efficient classifier, and assumes the form of a cascade of small and simple classifiers that use Multi-scale Block Local Binary Patterns (MB-LBP) features based on the work of Liao et al.(2007). Local Binary Patterns (LBP) features are intensity and scale invariant image descriptors. The original LBP features, introduced by Ojala, Pietikäinen, and Harwood(1996), label the pixels of an image by thresholding a rectangular region (e.g. 3×3) neighbourhood of each pixel with the centre value and considering the result as a binary string or a decimal number. MB-LBP extend the concept of LBP features to different sub-regions (blocks) around a center block, where the average intensity values of all the blocks are calculated. Then the LBP feature is extracted from the averages. The FD algorithm employs this classifier and can be summarised in following steps:

1. Define w and h as the width and height, respectively, of the input image.
2. Define a window of size $w' \times h'$, e.g. 20×20 .
3. Define a scale factor s greater than 1. For instance, a scale factor of 1.2 means that the window will be enlarged by 20%.
4. Calculate all windows with size $w' \times h'$ from the position $(0, 0)$ to $(w - w', h - h')$ with 1 pixel increments of the upper left corner.
5. Apply the cascade classifier for each window. The cascade has a group of stage classifiers, as represented in figure 3. Each stage is composed of a group of MB-LBP features that are applied to the window. If the overall resulting value is lower than the stage threshold, the classifier considers that the window does not contain a face and for this reason terminates the search. If it is higher, it continues to the next stage. If all stages are passed, the window is classified as containing a face.
6. Apply s to w' and h' , and go to step 4 until w' exceeds w or h' exceeds h .

Evolutionary Engine

The evolutionary engine is a conventional GA where the individuals are faces constructed from parts of different faces. Figure 4 explains the genotype of each individual and its phenotype. Each genotype is mapped into a phenotype by creating a composite face, i.e. the face parts encoded in the genotype are placed over a base face that is also encoded in the genotype. This process is accomplished by using a CV clone algorithm that allows the seamless placement of an image upon another (Pérez, Gangnet, and Blake 2003).

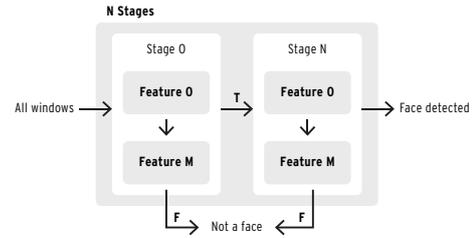


Figure 3: Overview of the FD process using a cascade classifier.

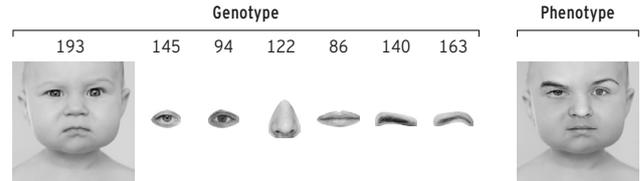


Figure 4: Genotype and phenotype of an individual. The genotype consists of a tuple of integers (*face*, *left eye*, *right eye*, *nose*, *mouth*, *left eyebrow*, *right eyebrow*). Each integer encodes an index of an annotated example. The phenotype consists of a composite of the face parts encoded in the genotype.

To guide evolution we adopt a fitness function that converts the binary output of the face classifier to an output that provides a suitable fitness landscape. This is attained by accessing internal values of the classification task that give an indication of the degree of certainty in the classification. In this work, we are interested in a fitness function that penalises individuals that are classified as faces. As such, the fitness function is defined as:

$$f(x) = (tstg - pstg) + (tstg * ndet) + \frac{1}{1 + stgdif} \quad (1)$$

where $tstg$ is the total number of stages of the classifier, $pstg$ is the number of stages that the input image passes, $ndet$ is a boolean variable that tracks if no face is detected in the image, i.e., if the generated face is not detected by the classifier it yields a value of 1; and $stgdif$ is the difference between the value attained in the last stage that the image passes and the threshold of that stage.

Experimental Setup

Since we are interested in evolving faces from existing ones, in a similar way as in a DA approach, our objective is to evolve positive examples that are misclassified as negative examples, i.e. faces that are classified as not faces. We begin by defining two datasets of positive examples, one with 200 examples and the other with 500 examples. We then use these two datasets to train the classifiers *face200* and *face500*, respectively. The *face200* dataset includes all 200 annotated examples that are used in the GA for recombination. The *face500* dataset contains all the *face200* examples plus 300 other examples. With these two datasets we intend to explore the impact of our approach in a scenario where all

available faces have their parts annotated and in a scenario that only a fraction of the available instances are annotated. Furthermore, we are interested in the analysis of the resulting individuals in both scenarios.

The positive examples that compose both datasets were extracted from the FACITY project, which is a world wide project that gathers the pictures of photographers capturing the multiplicity of human faces from different cities and countries¹. We maintain the same negative examples used in previous experiments, composed of 1905 images from the “Urtho – Negative face dataset”², which contains different types of images including landscapes, objects, drawings, and computer generated images. In the scope of this paper, we intend to study the results of our approach while using *face200* and *face500* to assign fitness.

Table 1: Training Parameters

Parameter	Setting
Example width	64
Example height	64
Number of stages	20
Min. hit rate per stage	0.999
Max. false alarm per stage	0.5
Adaboost algorithm	GentleAdaboost

We use the *opencv traincascade* tool of OpenCV to train each classifier. The main classifier parameters can be consulted in table 1 and were chosen based on the works of Viola and Jones (2001) and Lienhart, Kuranov, and Pisarevsky (2002). As for the FD settings we use the default parameters of OpenCV, which are presented in table 2. The test of each parameter is beyond the scope of this paper and besides that the default parameters enable a compromise between performance and speed of detection (Lienhart, Kuranov, and Pisarevsky 2002).

Table 2: Detection Parameters

Parameter	Setting
Scale factor	1.2
Min. face width	$0.7 \times$ example width
Min. face height	$0.7 \times$ example height

We test two experiments: *exp200* and *exp500*. In *exp200*, *face200* is used to guide evolution and *face500* to curate individuals. In *exp500*, *face500* is used to guide evolution and *face200* to curate individuals. The role of the curator is to observe the evolved individuals and to select relevant faces that it does not classify as faces. We study the behaviour of each curator and its selection of individuals.

The evolutionary engine settings are presented in table 3. In terms of face parts recombination, we maintain the pairs of eyes and the pairs of eyebrows, reducing the genotype length from seven to five. The rationale for this decision

Table 3: Evolutionary Engine Parameters

Parameter	Setting
Number of generations	50
Population size	50
Elite size	1
Tournament size	2
Crossover operator	uniform crossover
Crossover rate	0.8
Mutation operator	gene replacement
Mutation rate per gene	0.15

is related with the fact that most faces have a certain horizontal symmetry that the classifier tends to learn from the positive examples. In preliminary experiments, we have observed that the classifiers struggled on images where the pair of eyes and eyebrows belonged to different faces, leading to an early convergence of the evolutionary process. Besides the technical aspects, the images evolved were unnatural and easily noticeable that were blends.

Experimental Results

In this section we present and analyse the experimental results. We begin by analysing the evolution of fitness in the two experiments. Afterwards, we discuss the progression of detections over the generations. Then, we present and discuss the individuals selected by the curators. Finally, we analyse the visuals of the evolved individuals.

Figure 5 shows the evolution of fitness of the best individuals along the generations in *exp200* and *exp500*. For each experiment, we plot both fitness curves to examine how one affects the other. We can observe that the evolutionary algorithm is able to optimise the fitness function. In both experiments, when the fitness value of the guiding classifier increases, the fitness value of the curating classifier tends to behave similarly. The values reveal that it is easier to satisfy *face200* either when it is evaluating or curating. The observed behaviour in the *exp500* plot suggests that even being the *face500* that is guiding, the fitness values of *face200* are higher than the ones of *face500*. On the other hand when *face200* is guiding, the fitness that uses *face500* also increases, suggesting that *exp200* is also suitable of evolving solutions for the *exp500*. As we are interested in evolving individuals that are not classified as faces, one can say that we are promoting the evolution of examples that are not present in the training datasets. Bearing this in mind, while reflecting on the training datasets of both experiments, the results suggest that the evolved individuals in *exp500* tend to be new in the perspective of *face200*, i.e. different to the ones present in the *face200* training dataset. The behaviour of *face500* in *exp200* suggests that it is able to evolve individuals that are new in the *face500* dataset but at a smaller rate. This can be a consequence of the *face200* training instances being included in the *face500* training.

In figure 6 we observe the average of individuals that are classified as faces throughout generations. The number of detections decreases in both experiments, showing the ability of the approach to evolve faces that are no longer classified as such. In both experiments, *face500* obtains higher

¹FACITY project – <http://www.facity.com/>

²Haartraining negative dataset – <http://tutorial-haartraining.googlecode.com/svn/trunk/data/negatives/>

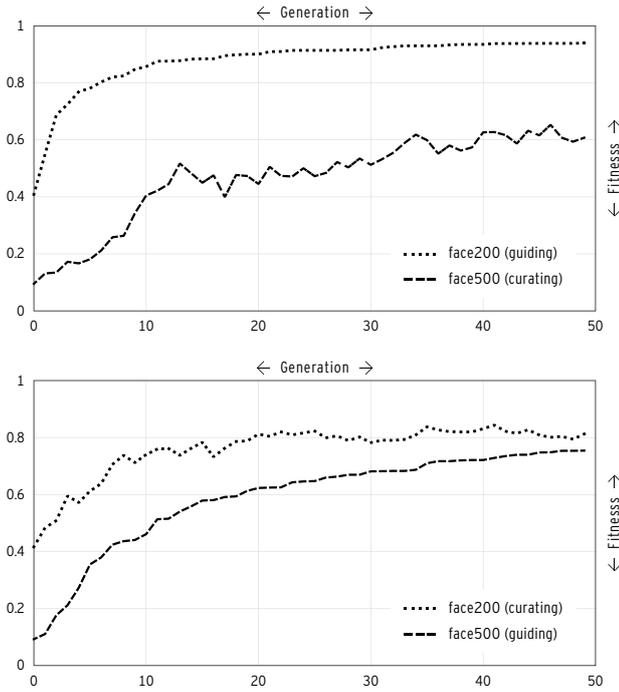


Figure 5: Evolution of the fitness of the best individual throughout generations when using *face200* to guide evolution and *face500* to curate individuals (top); and the other way around (bottom). The visualised results are averages of 30 runs.

number of detections than *face200*. This difference is more pronounced in *exp200*. As for the progression of the curves, one can see that in *exp200*, *face200* decreases at a higher rate. In *exp500*, both curves decrease at a similar rate. One can also say that in *exp200* the evolution promotes the generation of solutions that are classified as faces by *face500*. We consider that this is consistent with the fitness curve behaviour of figure 5, suggesting that the system exploits vulnerabilities common to both classifiers.

Figure 7 depicts in which generation the fittest individual, on average, cease to be classified as face. One can conclude that when *face200* is guiding, it takes less than 10 generations for the best individual to stop being classified as a face. The behaviour of *face500* suggests that the fittest individuals are still classified as faces in the final generations. In contrast, when *face500* is guiding, the results indicate that there are evolutionary runs where the fittest individual is still classified as a face by both classifiers.

Figures 8 and 10 depict a selection of fittest individuals registered in the experiments. As for figures 9 and 11, one can observe some of the curated individuals. The results suggest that although there are individuals in common, the two curators tend to select different individuals. Some of the selected faces share characteristics that we consider as exploits of the classifiers, particularly at the level of the skin tone, contrasts, and size of some facial features. One can conclude that there are overlaps between the fittest and the

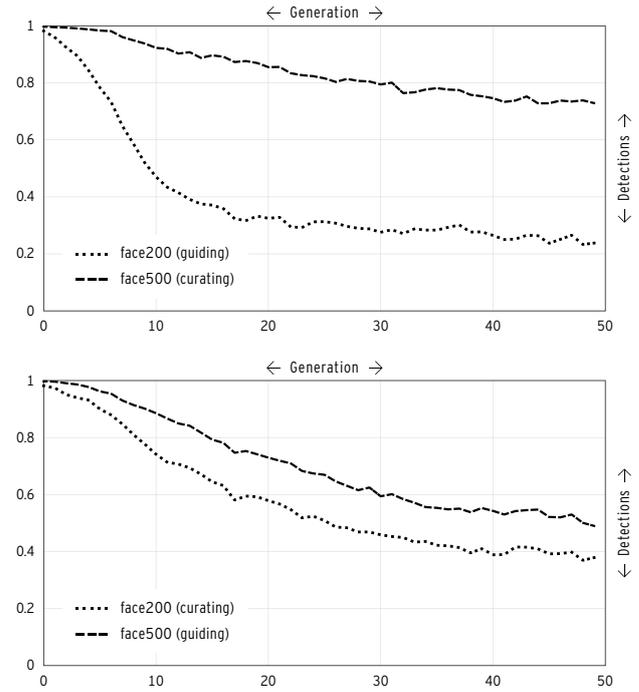


Figure 6: Progression of the average of detections when using *face200* to guide evolution and *face500* to curate individuals (top); and the other way around (bottom). The visualised results are averages of 30 runs.

curated individuals in *exp200* (see figures 8 and 9). When *face500* is curating or guiding, the evolved individuals share more characteristics (see figures 9 and 10). This is consistent with the idea that the exploits of *face500* tend to be also exploits of *face200*.

Figure 12 depicts a selection of individuals evolved in different runs that we found to be interesting and peculiar. This selection shows the ability of the approach to explore the search space and exploit the vulnerabilities of the classifiers in an automatic and tractable way. As such, one could expect simple recombinations of faces that the classifier has not "seen" before or exploits of lighting and contrast conditions. Nevertheless, the system produces atypical faces with unexpected features. For instance, one can see convincing images of babies with piercings, cases of gender ambiguity, and mixtures of interracial attributes that are at least visually uncommon and peculiar. Some of the generated faces are so realistic but disturbing at the same time that one could relate with the uncanny valley problem MacDorman et al. (2009), i.e., the phenomenon where computer generated figures or virtual humanoids that approach photo-realistic perfection make real humans uncomfortable.

Based on the notion of serendipity by Pease et al. (2013), the system with a prepared purpose, based on previous knowledge obtains a new result that is suitable and useful for the system and external sources. The purpose of this DA approach is to generate new useful examples of faces, in this case, faces that are not detected by the classifier that

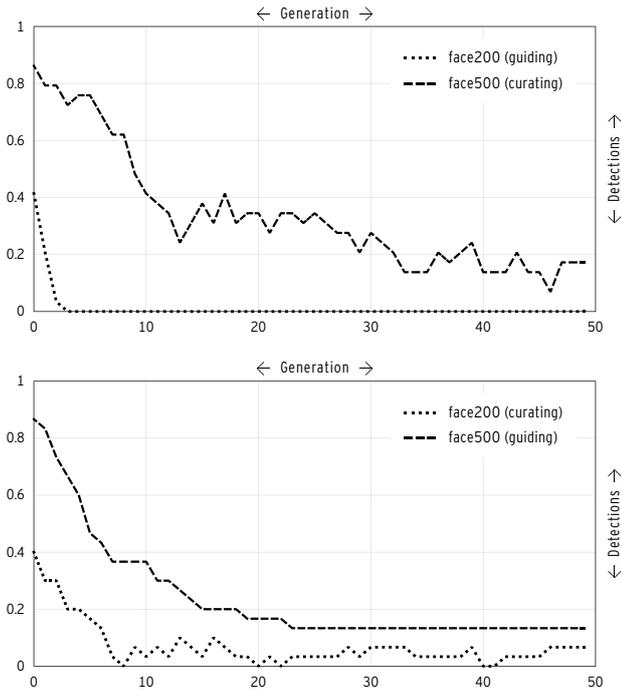


Figure 7: Progression of the detection of the best individual throughout generations when using *face200* to guide evolution and *face500* to curate individuals (top); and the other way around (bottom). The visualised results are averages of 30 runs.

is guiding the evolutionary process. Most of the outputs of this process are evaluated by the guiding classifier, by the computational curator, and by ourselves as new, valuable, interesting and unexpected. Since the system starts with a set of limited pre-defined parts, some individuals evolved suggest the occurrence of serendipity. Based on the ability of the system to generate such examples, one could insert domain knowledge to identify and enforce the generation of these examples. One could also use a curator or multiple curators to take an active part in the generation process, i.e. a component in the fitness function.

Although our approach generates faces that are not detected by the classifiers, the images can be used in other domain of applications. They can be used as inspiration to video games or movie making in the form of characters, and in visual arts to generate crowds with different faces. One could easily adapt the system and consider other items into play. For instance, the placement of the face parts could be modified for a horror scenario, allowing the face parts to be placed in different positions, e.g. replace all face parts with eyes, replace the two eyes with two mouths, and exchange the eyebrows with mouths. Due to the primary goal of this work, we only use human faces and their parts. However it is possible to use parts from other contexts, e.g. add and recombine parts from other animals and objects, allowing the creation of surrealistic artifacts. Our system can use multiple object classifiers to analyse photos and manipulate, mis-



Figure 8: Fittest individual in the last generation for 12 different runs when *face200* is guiding evolution.



Figure 9: Examples of faces curated by *face500* in different runs when *face200* is guiding evolution.

place, and edit the detected objects or introduce new ones.

A final comment goes for the potential use of this approach for DA. Similar to typical bootstrapping, this approach can generate variations or completely new examples from a pre-defined sub-set. The generated examples may be used to further improve the quality of the training dataset and thus the quality of the classifier, a path that is already being pursued.

Conclusions and Future Work

We have described and tested an approach for the automatic generation of faces based on the principles of combinatorial form of creativity. The experimental results demonstrate the ability of this approach to generate a wide variety of faces that test the ability of the classifiers to detect them. As such, we consider the approach proposed herein a viable solution for DA in the field of FD. The results also show the impact



Figure 10: Fittest individual in the last generation for 12 different runs when *face500* is guiding evolution.



Figure 11: Examples of faces curated by *face200* in different runs when *face500* is guiding evolution.

of different classifiers on the evolved faces. Besides fulfilling its purpose, from our perspective, the faces created have interesting and unexpected features.

The proposed approach may benefit from future enhancements, including: (i) the implementation of automatic detection and landmark mechanisms in the annotation tool to assist the annotation of the face parts; (ii) the use of the evolved faces to extract more face parts; (iii) the further integration of the proposed approach with EFECTIVE so the evolved examples are used to (re)train the classifiers in an attempt to improve their performance; (iv) the exploration of different curators; and (v) the expansion of the approach to other problems or scopes.

Acknowledgement

This research is partially funded by: Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grants



Figure 12: Examples of faces evolved in different runs. More faces can be visualised at <http://cdv.dei.uc.pt/2016/x-faces.mov>.

SFRH/BD/90968/2012 and SFRH/BD/105506/2014; and project ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

- [2009] Boden, M. A. 2009. Computer models of creativity. *AI Magazine* 30(3):23.
- [2007] Chen, J.; Wang, R.; Yan, S.; Shan, S.; Chen, X.; and Gao, W. 2007. Enhancing human face detection by resampling examples through manifolds. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 37(6):1017–1028.
- [2004] Chen, J.; Chen, X.; and Gao, W. 2004. Resampling for face detection by self-adaptive genetic algorithm. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, 822 – 825 Vol.3.
- [2013] Correia, J.; Machado, P.; Romero, J.; and Carballal, A. 2013. Evolving figurative images using expression-based evolutionary art. In *Proceedings of the fourth International Conference on Computational Creativity (ICCC)*, 24–31.
- [2009] DiPaola, S. R., and Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines* 10(2):97–110.
- [1995] Freund, Y., and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory, EuroCOLT '95*, 23–37. London, UK, UK: Springer-Verlag.
- [2004] Frowd, C. D.; Hancock, P. J. B.; and Carson, D. 2004. EvoFIT: A holistic, evolutionary facial imaging technique for creating composites. *ACM Transactions on Applied Perception* 1(1):19–39.
- [1997] Johnston, V. S., and Caldwell, C. 1997. Tracking a criminal suspect through face space with a genetic algorithm. In Bäck, T.; Fogel, D. B.; and Michalewicz, Z., eds., *Handbook of Evolutionary Computation*. Bristol, New York: Institute of Physics Publishing and Oxford University Press. G8.3:1–8.
- [2007] Liao, S.; Zhu, X.; Lei, Z.; Zhang, L.; and Li, S. Z. 2007. Learning multi-scale block local binary patterns for face recognition. In *Proceedings of the 2007 International Conference on Advances in Biometrics, ICB'07*, 828–837. Berlin, Heidelberg: Springer-Verlag.
- [2002] Lienhart, R.; Kuranov, A.; and Pisarevsky, V. 2002. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. *Computing* 2781(MRL):297–304.
- [2009] MacDorman, K. F.; Green, R. D.; Ho, C.-C.; and Koch, C. T. 2009. Too real for comfort? uncanny responses to computer generated faces. *Computers in Human Behavior* 25(3):695 – 710. Including the Special Issue: Enabling elderly users to create and share self authored multimedia content.
- [2015] Machado, P.; Vinhas, A.; Correia, J.; and Ekárt, A. 2015. Evolving ambiguous images. In Yang, Q., and Wooldridge, M., eds., *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 2473–2479. AAAI Press.
- [2012a] Machado, P.; Correia, J.; and Romero, J. 2012a. Expression-based evolution of faces. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design - First International Conference, EvoMUSART 2012, Málaga, Spain, April 11-13, 2012. Proceedings*, volume 7247 of *Lecture Notes in Computer Science*, 187–198. Springer.
- [2012b] Machado, P.; Correia, J.; and Romero, J. 2012b. Improving face detection. In Moraglio, A.; Silva, S.; Krawiec, K.; Machado, P.; and Cotta, C., eds., *Genetic Programming - 15th European Conference, EuroGP 2012, Málaga, Spain, April 11-13, 2012. Proceedings*, volume 7244 of *Lecture Notes in Computer Science*, 73–84. Springer.
- [1996] Ojala, T.; Pietikäinen, M.; and Harwood, D. 1996. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition* 29(1):51–59.
- [2013] Pease, A.; Colton, S.; Ramezani, R.; Charnley, J.; and Reed, K. 2013. A discussion on serendipity in creative systems. In *Proceedings of the Fourth International Conference on Computational Creativity*, 64.
- [2003] Pérez, P.; Gangnet, M.; and Blake, A. 2003. Poisson image editing. In *ACM Transactions on Graphics (TOG)*, volume 22, 313–318. ACM.
- [2014] Schorlemmer, M.; Smaill, A.; Kühnberger, K.-U.; Kutz, O.; Colton, S.; Cambourooulos, E.; and Pease, A. 2014. Coinvent: Towards a computational concept invention theory. In *Proceedings of the Fifth International Conference on Computational Creativity, ICC2014*, 288–296. International Association for Computational Creativity.
- [1998] Sung, K.-K., and Poggio, T. 1998. Example-based learning for view-based human face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20(1):39–51.
- [2010] Ventrella, J. 2010. Self portraits with mandelbrot genetics. In *Proceedings of the 10th international conference on Smart graphics, SG'10*, 273–276. Berlin, Heidelberg: Springer-Verlag.
- [2001] Viola, P., and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, I–511–I–518 vol.1.
- [2002] Yang, M.-H.; Kriegman, D.; and Ahuja, N. 2002. Detecting faces in images: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24(1):34–58.
- [2010] Zhang, C., and Zhang, Z. 2010. A survey of recent advances in face detection. Technical report, Tech. rep., Microsoft Research.

Before A Computer Can Draw, It Must First Learn To See

Derrall Heath and Dan Ventura

Computer Science Department

Brigham Young University

Provo, UT 84602 USA

dheath@byu.edu, ventura@cs.byu.edu

Abstract

Most computationally creative systems lack adequate means of perceptually evaluating the artifacts they produce and are therefore not fully grounded in real world understanding. We argue that perceptually grounding such systems will increase their creative potential. Having adequate perceptual abilities can enable computational systems to be more autonomous, learn better internal models, evaluate their own artifacts, and create artifacts with intention. We draw from the fields of cognitive psychology, neuroscience, and art history to gain insights into the role that perception plays in the creative process. We use examples and methods from deep learning on the task of image generation and pareidolia to show the creative potential of systems with advanced perceptual abilities. We also discuss several issues and philosophical questions related to perception and creativity.

Introduction

Some people seem to have a natural talent for drawing, while others only wish they could draw well. Many of these people have turned to books and teachers to help them develop their drawing skills. One of the most widely used and consistently successful books for teaching people how to draw is titled *Drawing on the Right Side of the Brain* (Edwards 1989). This book uses insights from neuroscience to help potential artists improve their drawing skills. One of the main premises in the book is that drawing is not a skill of hand, paper or pencil, but a skill of perception. To quote from the book:

“The magic mystery of drawing ability seems to be, in part at least, an ability to make a shift in brain state to a different mode of seeing/perceiving. When you see in the special way in which experienced artists see, then you can draw... Drawing is not very difficult. Seeing is the problem, or, to be more specific, shifting to a particular way of seeing.”

This idea can extend to any kind of creative ability. Before one can create visual art, compose music, or invent recipes, one must first learn to see, hear/listen, or taste, respectively. Even creative tasks like writing poetry must ultimately be grounded to what has been experienced through perception. Our ability to perceive influences how and what we create. Just as drawing is really about perceptual skills, our ability to think creatively and do creative things heavily depends on how we perceive and understand the world.

In his book, *The Anthropologist On Mars*, Oliver Sacks recounts the story of Shirley Jennings, who had been blind

since early childhood and had surgically regained his sight at the age of 50 (Sacks 1995). After the operation, he could not immediately see, could not recognize his family, could not pick out objects, and struggled with depth perception. Over several months, his brain had to learn how to see and make sense of an incredible amount of new information. It was a slow and difficult process reconciling his non-visual mental model of the world with this new form of perception. As he learned to make use of his new sense, things that were aesthetically beautiful to him differed from those others found pleasing. He eventually took painting lessons and created paintings that demonstrate his unique taste in visual art¹.

Shirley’s case, along with several other vision disorders and anomalies like blindsight (Weiskrantz 1996), Capgras syndrome (Ellis and Lewis 2001), and agnosia (Farah 2004), has helped to uncover the work and learning that our brain undertakes in order for us to perceive and understand the world. In this paper we argue that the ability to perceive is a necessary and influential piece of the creative process. It enables us to learn a mental model of the world, to understand and continually evaluate our own creations, and to infuse what we produce with meaning. Indeed, even perception itself is a creative act that our brains regularly perform, although often subconsciously. The necessity of perception applies to the field of computational creativity, in which one of the goals is to build computational systems that can autonomously create art. Before a system can learn to create art, we argue that it must first learn to perceive.

We proceed by exploring the relationship between perception and creativity and then discuss the role of perception in computational systems. We then consider how state-of-the-art computer vision methods can enhance the creative potential of systems designed for visual art. We demonstrate, using deep neural networks, how perceptual skills facilitate imagination and can lead directly to generating novel images. We then elaborate further on why perception itself is a creative process and demonstrate a form of creative perception, called pareidolia, using deep neural networks. Finally, we discuss philosophical issues and the implications of our ideas and elaborate on what more advanced perceptual abilities could mean for the future of computational creativity.

¹<http://www.atfirstsightthebook.com/shirls-paintings.html>

Perception and Creativity

When talking about visual art, Csikzentmihályi says, "...the aesthetic experience occurs when information coming from the artwork interacts with information already stored in the viewer's mind..." (Csikzentmihályi and Robinson 1990). In other words, the viewer's appreciation (perception) of art is determined by his current mental model of the world. Likewise, the artist has her own mental model of the world and created the artwork to convey meaning according to that mental model. How was that mental model established? It's reasonable to say that it was learned through a lifetime of experiences, and people experience the world through perception. Everything we know and understand about the world has come through our senses. Every memory and every thought we have is in terms of what we have experienced in the past (Barsalou 1999).

It is difficult to comprehend what life would be like without perception because it is so fundamental to how we think. Would it even be possible to think, imagine, or create anything at all without some kind of input? There is no definite answer to that question; however, studies of long term sensory deprivation and solitary confinement suggest significant mental deterioration (Grassian and Friedman 1986; Allen, Celikel, and Feldman 2003). Perception directly influences our ability to think and understand, and the better and more varied our perceptual abilities are, the more we are able to think about, imagine, and ultimately, create. We can take this idea further and say that, with our current senses, there are thoughts we cannot think simply because we lack additional (or adequate) senses to know how to think them. To quote Richard Hamming (Hamming 1980):

"Just as there are odors that dogs can smell and we cannot, as well as sounds that dogs can hear and we cannot, so too there are wavelengths of light we cannot see and flavors we cannot taste. Why then, given our brains wired the way they are, does the remark 'Perhaps there are thoughts we cannot think,' surprise you? Evolution, so far, may possibly have blocked us from being able to think in some directions; there could be unthinkable thoughts."

We need perception (i.e., input) in order to build a mental model that can facilitate thinking, which can then facilitate creativity (Flowers and Garbin 1989). Indeed, as noted earlier in the case of Shirley Jennings (Sacks 1995), the mental model itself is what does the perceiving. Our eyes merely translate light into brain signals, but it is our brain that must learn to make sense of that information, which then allows us to think in those terms.

Imagination is clearly tied to this idea and is closely linked with creativity in cognitive psychology literature (Gaut 2003). Imagination is typically generalized as thinking of something (real or not) that is not present to the senses. Most psychologists agree that our perceptions (senses), our conceptual knowledge, and our memories make up our mental model and form the bases of imagination (Barsalou 1999; Beaney 2005). As we perceive the world and have experiences, our mental model is formed by establishing and strengthening connections in our mind. These connections form concepts, which are in turn interconnected. Creative imagination is achieved by combining

these connections and experiences in different ways that produce novel results.

Thinking Beyond Natural Perception

It is possible for us to indirectly experience things outside of our perceptual abilities by translating other modalities into our range of senses. For example, we visualize infrared light by shifting it into the visible spectrum. We create charts and graphs that represent data we cannot observe directly, like barometric pressure, or electromagnetic fields. In this way we can vicariously think in terms of other modalities and perhaps even be creative in those modalities.

This idea is applied explicitly in the case of *sensory substitution* (Bach-y-Rita and Kercel 2003), where one sense can take the place of another that has been lost. For example, devices have been made that can allow blind people to literally "see" with their tongue. They work by mounting a video camera on the blind person's forehead, which sends video data to a plate that sits on the person's tongue. This plate contains a grid of "pixels" consisting of pressure points. These pressure pixels correspond to grayscale video by pressing harder where the image is brighter and pressing lighter where the image is darker. The tongue can then "feel" the video information and, after several months of training, a blind person's brain starts to see images in their mind. It's certainly not high resolution, but it's enough to allow a blind person to read large print text and navigate new terrain.

Another way that we humans can communicate and understand things that we ourselves have not perceived directly is through language. In other words, through verbal/written communication we can experience by proxy what others have directly experienced (Zwaan and Kaschak 2008). In this case, language acts as an analogy between two people's experiences. Our interpretation of a described experience must still be grounded by our personal perceptions and experiences (Barsalou 1999). For example, it is very difficult to describe colors to a congenitally blind person because colors are inherently visual and the blind person has no visual grounding at all. This is why even creative literature and poetry also require perception—the writer must have experiences to write about and the reader must have experiences with which to interpret the writing.

Art, whether it be visual, written, musical, etc, acts as a metaphor between the experiences of the artist and the experiences of the receiver. Successful artists are creative because they have a unique perspective on the world that they are trying to communicate through their art, and people appreciate art that helps them gain new perspectives. In other words, having unique experiences and perceiving the world differently plays a role in the creative process. It has been postulated in cognitive psychology that creative people literally see the world differently (Flowers and Garbin 1989; Berns 2008), which is in turn why they tend to think differently and can produce novel things and ideas.

Quality of Perception Affecting Visual Art

There have been studies analyzing several famous artists with documented visual impairments (Marmor and Ravin 2009). For example, Claude Monet developed cataracts,

while Edgar Degas began to suffer from retinal disease. These studies point out that the earlier works of these painters (when they had good eyesight) were better formed and detailed, while later works (made with poorer eyesight) became more and more abstract. These studies generally conclude that the failing eyesight of the artists *did* have a large impact on the quality and style of their work. Although some researchers say that this was not necessarily a bad thing, and some artists would use their visual impairments to their advantage by removing their corrective lenses for certain paintings.

These artists had issues with just their eyes, but what about *cognitive* impairments involving vision? How do different cognitive disorders of the brain affect artists' work? This question was explored by Anjan Chatterjee, where he reported on multiple studies analyzing the drawing ability of several artists with various cognitive disorders, including spatial neglect, visual agnosia, epilepsy, TBI, etc (Chatterjee 2004). The results for many of the disorders, like epilepsy were mixed, but artists with disorders more specific to vision, like agnosia, had some notable peculiarities to their drawing ability.

For example, one artist with a type of visual agnosia could create beautiful drawings as long as the item he was drawing was continuously right in front of him. However, if he was asked to draw from memory (e.g., draw a 'bus'), then his drawings were simplistic and often unrecognizable. Another artist, with a traumatic brain injury, produced drawings that were more abstract and "expressive" than drawings produced before the accident. Although these studies appear anecdotal due to the rarity of many of these disorders, it is apparent that how the brain sees and understands the world does affect the ability to draw.

There are several cases of successful artists who are blind, and one artist in particular has received a lot of attention because he is *congenitally* blind (Amedi et al. 2008). This blind man has a remarkable ability to paint and draw pictures that are consistently meaningful to sighted people. He uses special paper and pencils that form ridges that he can feel as he draws. He first explores an object with his hands and then, remarkably, can draw it from different perspectives. He's never been able to see, yet can understand perspective. His case provides insight into how the brain perceives and builds invariant mental representations of the world.

The blind artist's case is related to sensory substitution, where a blind person can "see" through touch, and further supports the idea that the brain is what processes and makes sense of perceptual input. Researchers who study blindness and visual art have indicated that vision and touch are linked and make use of similar processes and similar features in the brain (Kennedy 1993; Kamel and Landay 2000). The brain can do remarkable things even when the quality of the input signal is disrupted or re-routed. Perception is really about being able to build these mental models and using them to interact with the world. It's not that *visual* art requires *vision*, but that creating visual art requires *some* form of perception that establishes and continually informs the artist's mental model.

Perception and Computational Creativity

We've discussed the role of perception in *human* creativity, but what about computational creativity? Certainly, there's no requirement that computers can only be creative in the same way as humans. However, we are positing that perception is *fundamentally* a necessary component of the creative process. So, just as perception is important for human creativity, perceptual ability is also important for computational creativity. The exact methods of perceiving and creating may be different than those of humans, but some form of perceptual grounding is requisite for a truly creative system.

Colton proposed the creative tripod as necessary criteria for a creative system (Colton 2008). A creative system must have imagination, which is analogous to producing novel artifacts; it must have skill, which corresponds to generating quality artifacts; and it must have appreciation, which is the ability to recognize the novelty and quality of its own artifacts (i.e., self-evaluation). In other words, there must be a perceptual component that directs the creative process by helping the system explore new ideas (imagination), and understanding which ideas are worth pursuing (appreciation).

Many creative systems exist across several domains that can generate novel artifacts. Most of these systems, however, are merely mimicking example human-created artifacts without understanding or appreciating what they are producing, like a parrot mimicking human speech. For example, the PIERRE system generates new crockpot recipes according to a model trained on user ratings of existing recipes, but it has no sense of what the recipes actually taste like, only that humans have liked similar recipes (Morris et al. 2012). In music, there are several systems that analyze patterns and *n*-grams from existing melodies, then probabilistically draw from those distributions or construct grammars when producing music (Cope 1996; Pachet and Roy 2011). Likewise, poetry systems are also often based on corpora and *n*-gram distributions, without much understanding of what the words actually represent (Colton, Goodwin, and Veale 2012; Netzer et al. 2009).

Other existing creative systems produce artifacts according to hand engineered metrics and databases, where the ability to appreciate and perceive what is produced is limited to those explicit metrics. For example, some musical systems rely on rules and metrics based on musical theory in order to produce and evaluate melodies (Ebcioglu 1988; Melo and Wiggins 2003). Visual art systems often use some form of evolutionary algorithm for producing art, which involves a fitness function by which the art is evaluated at each iteration. The fitness functions in these systems are usually based on models trained using extracted image features in order to evaluate aesthetic quality or novelty (Machado, Romero, and Manaris 2007; DiPaola and Gabora 2009). In these cases the perceptual ability is ultimately limited to those specific features.

There are some creative systems that do attempt to incorporate a sophisticated model of perceptual ability. For example, there is a system that invents recipes based on actual chemical properties of the individual ingredients (Varshney et al. 2013). It at least has some understanding of what would actually taste good in a recipe and isn't limited to

just producing something that is mimicking human examples. The DARCI system extracts various image features and trains neural networks to evaluate how well the images convey the meaning of particular adjectives (Heath and Ventura 2016). Although DARCI still relies on extracting specific low level features, it at least attempts to learn the semantic qualities related to those features (in the form of adjectives). In this way DARCI, more than other visual art systems, is able to at least partially perceive *meaning* in the art that is produced.

The example systems just described can produce interesting and novel artifacts. However, without advanced perceptual abilities, the systems lack any notion of understanding and intentionality. The systems can produce something, but can't necessarily tell us why, or what it means. They are instances of Searle's Chinese room (Searle 1980), that simply follow rules and algorithms, with no comprehension of what is taking place. Just as humans cannot think beyond our perceptions, computational systems cannot think beyond theirs. Some have argued that even human thought and creativity is subject to the Chinese room analogy at the biological (cellular) level. This may be true, but if we aim to build systems that can be creative at a human level, then they must at least have human-level perception.

Somewhat surprisingly, in the case of visual art, current creative systems rarely use state-of-the-art computer vision techniques, like deep neural networks. Certainly having more advanced perceptual abilities would improve the quality of their art by enabling these systems to understand more concrete things. For example, a system could conceivably create an original image of a dog, if it knew how to see and recognize dogs. It seems, then, that incorporating advanced computer vision techniques, especially ones tied to semantic understanding, should be a high priority in the field of computational creativity.

Visual Art and Deep Learning

The last few years have shown a resurgence of *deep neural networks* (DNNs), especially for computer vision tasks, where they hold current records for several vision benchmarks (Farabet et al. 2013; Szegedy et al. 2015). Deep learning has the potential to significantly improve visually creative systems as well. A key advantage of DNNs is that they are capable of learning their own image features, while the visual art systems described above all rely on manually engineered features. Thus, deep learning models can provide more advanced perceptual abilities by building better "mental" models of the world.

Some of these deep learning models can already be used directly to improve current artistic systems. In recent work on DARCI, we built a sophisticated semantic model that uses a shallow neural network to associate image features with a vector space model (Heath and Ventura 2016). Here we can show significant improvement by replacing the shallow neural network and extracted features, with a DNN (and the raw image pixels as input). Specifically, we used a deep learning framework, called Caffe (Jia et al. 2014), and started with the CaffeNet model, which was first trained to recognize 1000 different items using the ImageNet 2012

	<i>Random</i>	<i>DARCI</i>	<i>Deep Network</i>
Coverage	0.709	0.444	0.202
Ranking Loss	0.502	0.199	0.102

Table 1: Zero-shot image ranking results comparing the DARCI system with our modification of DARCI that uses a deep neural network (lower scores are better). We used the same test set from the original DARCI paper (Heath and Ventura 2016). The use of a DNN improves the system's ability to perceive and understand adjectives in images.

competition data (Russakovsky et al. 2015). We then further trained and fine-tuned the model on DARCI's image-adjective dataset (with a vector space model).

The DARCI system is capable of zero-shot prediction (using the vector space model), meaning it can successfully evaluate images for adjectives that it was not explicitly trained on. We compare DARCI's original results (Heath and Ventura 2016) with our deep neural network version in Table 1. The results show significant improvement using the DNN to evaluate images, and fully incorporating a deep model into the DARCI system will likely help it to produce more semantically relevant images.

In fact, DNNs have already been used to generate images directly (Denton et al. 2015; Gregor et al. 2015; Leon A. Gatys and Bethge 2015). One particular method, called *gradient ascent* (Simonyan, Vedaldi, and Zisserman 2013), works by essentially using the DNN in reverse. The trained network starts with a random noise image and tries to maximize the activation of the output node corresponding to the desired class to generate. The network then backpropagates the error into the image itself (keeping the network weights unchanged) and the image is slightly modified at each iteration to look more and more like the desired class.

We demonstrate gradient ascent using the same deep model that we trained with the DARCI image-adjective data set, and the resulting images can be seen in Figure 1. These images can be thought of as visualizations of the features learned by the model for each adjective. Each adjective's features seem fairly general, except in the case of 'peaceful', where the visualized features are consistent with the fact that most of the training images depict calm beaches. It is theorized that imagination in humans can be partially thought of as running our vision processing systems in reverse (Barsalou 1999), in which case our deep neural network is analogously demonstrating its own kind of imagination.

The generated images seem fairly abstract, which is expected for adjectives, especially since the DARCI data set contains a wide variety of scenes, objects, genres, and styles for each adjective label. Deep neural networks are becoming powerful enough to render actual recognizable objects using the gradient ascent method. The ImageNet 2012 competition consists of classifying 1000 different categories of objects ranging from various animals, to clothing, to household items. We took the CaffeNet model (used as the base for the DARCI model), as well as another successful model called GoogleNet (Szegedy et al. 2015), and generated several images depicting objects from the 1000 possible cate-



Figure 1: Four images generated using gradient ascent from the deep neural network trained on the DARCI dataset. From left to right the images were generated for the adjectives ‘vibrant’, ‘cold’, ‘fiery’, and ‘peaceful’. These images are essentially visualizations of the features that the model has learned and demonstrate a form of imagination.

gories. The resulting images for several objects can be seen in Figure 2.

While the images are not photo-realistic, they are original and do resemble the intended item. Notice how the two models generated images with different styles as each model learned different features. The generation of images using DNNs is currently an active area of computer vision and machine learning research, and several researchers have produced impressive results (Denton et al. 2015; Leon A. Gatys and Bethge 2015). The field of computational creativity has yet to significantly leverage the potential of deep learning, although some have alluded to it (Heath, Dennis, and Ventura 2015). However, some researchers have already begun incorporating deep learning into evolutionary art systems that are capable of rendering images that resemble concrete objects, with interesting results (Nguyen, Yosinski, and Clune 2015).

To See Is To Create

We have argued that perception is an important aspect of creativity and that more advanced perceptual abilities can lead to more sophisticated creative systems. We also argue that perception is a creative act in its own right. When light hits a person’s eyes, it is converted into signals, which travel to the visual cortex via the optic nerve. The brain itself does not receive any light, only information about the light. The brain must then learn to make sense of that information, and an image in the mind is fabricated, and that is what a person “sees”. Our brain over our lifetime has built a mental model of the world through the various signals it has received from our senses. This mental model is what determines our personal reality, and it is an impressively creative act (Hoffman 2000; Peterson 2006).

We do not think of perception itself as a creative act because it happens instantly, constantly, and seemingly without effort. We take for granted how difficult perception is because it is an ordinary part of life, and we have become desensitized to it. However, even the most advanced state-of-the-art computer intelligence cannot process visual information as well as a child can almost instantaneously. The case of Shirley Jennings (Sacks 1995), in which he spent months learning how to see for the first time at age 50, and other cognitive visual disorders, shed light on the tremendous amount of work that goes into vision.



Figure 2: Images generated using gradient ascent from the CaffeNet model and the GoogleNet model, both trained on the 2012 ImageNet challenge data. The first two rows of images are from CaffeNet and, from left to right, were generated for ‘pool table’, ‘broccoli’, ‘flamingo’, ‘goldfish’, ‘bald eagle’, ‘lampshade’, ‘starfish’, and ‘volcano’. The last row of images are from GoogleNet and were generated for ‘bald eagle’, ‘tarantula’, ‘starfish’, and ‘ski mask’. These original images are certainly not photo realistic, but it is still fairly easy to identify each image’s subject. Notice that the two models have different styles because they have learned different features.

Optical illusions also provide insights on how the brain understands visual input and constructs images in the mind (Hoffman 2000). Different people given the same input, experience it differently. A person’s subjective experience is unique to them, an act of novelty by their creative brain. This idea became even more evident when a particular image of a dress sparked huge debate on social media over the color of the dress (Lafer-Sousa, Hermann, and Conway 2015). Some saw white/gold, others saw blue/black, because our brains construct differing realities based on our mental models.

If we accept the idea that our brains are doing the actual creating of the images we see, then what is the artist doing when she paints a picture? The artist is providing a set of constraints, in the form of a painting, that viewers use to create an image in their minds. The more realistic a painting is, the more it constrains the viewer to mentally create it a certain way. The more abstract or ambiguous the painting is, the less it constrains the viewer, and the more variety and novelty in the individual aesthetic experiences.

Pareidolia

Attributing creativity to a system just because it has some perceptual abilities does not appear very compelling. However, there are some perceptual tasks that seem more creative than others. Pareidolia is the phenomenon of perceiving a familiar pattern where none actually exists. For example, seeing constellations in the stars, faces in ordinary things,

objects in blotches of ink, or shapes in the clouds. Sometimes these are considered mistakes or optical illusions, but they can actually be a deliberate act of creativity. When a child says a cloud looks like a particular animal, we admire her imagination, especially when we can then see the shape too. We obviously know it's a cloud, but we have chosen to see it as something else.

Pareidolia is a creative act because it is not about seeing things for what they are but seeing things for what they could be. Creative systems capable of pareidolia may have applications in visual communication, advertising, story telling, illustration, and non-photo-realistic art. As it stands, there are few computational systems developed for automatically performing pareidolia. One group of researchers developed a system for recognizing "faces" in ordinary pictures and then automatically determining the emotion expressed by the "face" (Hong et al. 2013). Here we demonstrate how deep learning can be used for pareidolia and argue that it is a form of creativity because the model is interpreting images in novel ways.

Finding Faces Seeing faces in objects is by far the most common type of pareidolia and provides a simplified version of the problem to begin with. The initial task is to use a deep neural network (DNN) to identify what aspects of an image could make up a face. We then have the DNN iteratively emphasize those features, using gradient ascent, until the "face" that the network sees emerges in the image. We use two different DNN models trained on faces. The first is called VGG-Face and was trained to recognize the faces of over 2500 different celebrities (Parkhi, Vedaldi, and Zisserman 2015). The second model, which we'll call AGE-Face, was trained to determine the age of a person (one of eight age ranges) in a provided image (Levi and Hassner 2015).

We perform pareidolia by having each network, when given an image, determine the output node (corresponding to a class) with the highest activation. The model then performs the gradient ascent algorithm in an attempt to increase that node's activation further, thus emphasizing the strongest features it found initially. Figure 3 shows example pareidolia images generated with both the VGG-Face and AGE-Face networks. The networks generally do a decent job of drawing (cartoony) faces on the source images in ways that make sense, although some are harder to appreciate. The VGG-Face model tends to draw more realistic facial features (i.e., eyes, nose, etc) than the AGE-Face model. However, the VGG-Face model will often highlight isolated facial features (especially when a face in the source image is not apparent to humans), while the AGE-Face model tends to keep the facial features together for a full face.

Finding Objects We now move on to a harder version of pareidolia in which we ask the model to find and highlight any kind of object in an image. We again use the CaffeNet model that was trained on the 1000 category 2012 ImageNet data; thus the model could potentially see any of those 1000 items in an image. We use the same method as just described in the faces version. The model is given a source image, then performs gradient ascent on the source image in order to further maximize the highest activated output node. We



Figure 3: Images created for face pareidolia using deep neural networks. The top row are the source images, the second row are faces highlighted by the VGG-Face model, and the third row are faces highlighted by the AGE-Face model.



Figure 4: Images for object pareidolia using CaffeNet, trained on the 2012 ImageNet data for 1000 object categories. From left to right, the items highlighted in the images (bottom row) from each source image (top row) are 'mask', 'arctic fox', 'scorpion', and 'ringworm'.

applied this method to several source images, and the results can be seen in Figure 4.

For some of the examples it is easy to see why the model did what it did. For instance, it is understandable how the [Figure 4, 1st] source image looks like a mask, and we can see how the modified image came from it. However, it is more difficult to appreciate how the model saw an 'arctic fox' in the [Figure 4, 2nd] source image. Other examples are hard to relate to initially, but on inspection, we can start to see the connection. For example, the [Figure 4, 4th] source image looks, to most humans, like a spider, but the model saw it as a ringworm. After considering the resulting image, we can at least appreciate why the model thought ringworm.

This leads to an interesting discussion about perception and creativity. If a person says that a particular cloud looks like a horse, then *if we can also see it*, we think the person has imagination. However, if we can not see it ourselves, then we do not necessarily praise the person's imagination.

Conversely, if a person says that a photo of a horse looks like a horse, we also do not admire the imagination, and we end up wondering why they bothered to say something so obvious. We appreciate creativity when it is different from the norm, but not so different that we cannot connect.

When it comes to visual art, how a person sees will influence their art; thus, people that see things differently (but not too differently) can potentially be more creative with their art. Using deep learning models for pareidolia helps us to understand how these models are actually seeing, and it helps us to visualize what features are being learned. The features learned by each model are likely different than the features that human brains use when processing visual input. This is why the CaffeNet model sees an arctic fox in the [Figure 4, 2nd] source image, but most humans would say it looks like an elephant.

If a computational system perceives things differently than a human, and accordingly produces different kinds of art, then is the art only viable if we humans can relate to it? It has been suggested that the most creative and influential people are ones that see (and therefore think) differently (Flowers and Garbin 1989; Berns 2008), and Colton argues that computational systems that see differently than humans have enhanced creative potential (Colton et al. 2015), but is that true only to an extent? Could a computational system that perceives differently (even radically differently) than humans actually help us to extend our notions of what constitutes good art?

To go even further, could we build a system capable of understanding and creating art beyond the capabilities of current human perception? For example, could we build a system that creates infrared art? Or electromagnetic field art? Or gravity art? Or some other kind of art? Would there be any purpose in doing so? Or perhaps augmenting computational systems with other forms of perception could help them gain a richer, deeper understanding of the world, and allow them to create visual art that can be even more meaningful to humans.

Conclusion

We have argued that perceptual abilities are fundamental to the creative process. We have discussed the relationship between perception and creativity from a cognitive psychology perspective and also in terms of computational systems. We have even asserted that perception itself is a creative act and that perceiving things differently can facilitate creative thinking. We have demonstrated how state-of-the-art deep neural networks can be used to create images and perform certain types of imagination, and we have also demonstrated how they can see creatively through pareidolia.

As with humans, advanced perceptual abilities can provide a foundation on which computational systems can think, imagine, and create. In the field of artificial general intelligence, current trends and ideas are also advocating the need for perception, and recent general AI systems are learning to perform intelligent tasks exclusively from raw inputs (Hawkins and Blakeslee 2007; Mnih et al. 2015). They argue that having a system learn from the ground up,

with raw inputs (e.g., raw pixel values), is essential for general/adaptable intelligence. Perceiving and understanding various raw inputs can act as a basis for a large variety of intelligence tasks, and learning how to perceive and perform for one task should transfer to additional tasks. Furthermore, advanced cognitive ability, such as language and reasoning, could emerge naturally from these perceptual primitives as they form connections and hierarchies of understanding.

The idea of perceptual primitives can also be applied to a general notion of computational creativity. Ideally, we would like to develop a universal creative process, which allows for connections to form across multiple domains, experiences, and knowledge. Perceptual abilities for multiple modalities establish an internal mental model of the world, which can provide a system with freedom and adaptability to be creative in any of its modalities or combination of modalities. For example, a system trying to invent recipes could benefit from visually recognizing ingredients (in addition to understanding how they taste) and could invent new recipes by substituting similar looking ingredients. It is possible that developing and incorporating advanced perceptual abilities in computational systems will not only increase the creative potential of those systems but may also facilitate the abstraction of a domain-independent, general creativity “algorithm”.

References

- Allen, C. B.; Celikel, T.; and Feldman, D. E. 2003. Long-term depression induced by sensory deprivation during cortical map plasticity in vivo. *Nature Neuroscience* 6(3):291–299.
- Amedi, A.; Merabet, L. B.; Camprodon, J.; Bermpohl, F.; Fox, S.; Ronen, I.; Kim, D.-S.; and Pascual-Leone, A. 2008. Neural and behavioral correlates of drawing in an early blind painter: a case study. *Brain Research* 1242:252–262.
- Bach-y-Rita, P., and Kercel, S. W. 2003. Sensory substitution and the human-machine interface. *Trends in Cognitive Sciences* 7(12):541–546.
- Barsalou, L. W. 1999. Perceptual symbol systems. *Behavioral and Brain Sciences* 22(04):637–660.
- Beaney, M. 2005. *Imagination and Creativity*. Open University Milton Keynes, UK.
- Berns, G. 2008. *Iconoclast: A neuroscientist reveals how to think differently*. Harvard Business Press.
- Chatterjee, A. 2004. The neuropsychology of visual artistic production. *Neuropsychologia* 42(11):1568–1583.
- Colton, S.; Halskov, J.; Ventura, D.; Gouldstone, I.; Cook, M.; Blanca, P.; et al. 2015. The Painting Fool sees! new projects with the automated painter. In *Proceedings of the 6th International Conference on Computational Creativity*, 189–196.
- Colton, S.; Goodwin, J.; and Veale, T. 2012. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.
- Colton, S. 2008. Creativity versus the perception of creativity in computational systems. *Creative Intelligent Systems: Papers from the AAAI Spring Symposium* 14–20.
- Cope, D. 1996. *Experiments in musical intelligence*, volume 12. AR editions Madison, WI.
- Csíkzentmihályi, M., and Robinson, R. E. 1990. *The Art of Seeing*. The J. Paul Getty Trust Office of Publications.

- Denton, E. L.; Chintala, S.; Fergus, R.; et al. 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, 1486–1494.
- DiPaola, S., and Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines* 10(2):97–110.
- Ebcioğlu, K. 1988. An expert system for harmonizing four-part chorales. *Computer Music Journal* 12(3):43–51.
- Edwards, B. 1989. *Drawing on the Right Side of the Brain*. New York: Tarcher.
- Ellis, H. D., and Lewis, M. B. 2001. Capgras delusion: a window on face recognition. *Trends in Cognitive Sciences* 5(4):149–156.
- Farabet, C.; Couprie, C.; Najman, L.; and LeCun, Y. 2013. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8):1915–1929.
- Farah, M. J. 2004. *Visual Agnosia*. MIT press.
- Flowers, J. H., and Garbin, C. P. 1989. Creativity and perception. In *Handbook of Creativity*. Springer. 147–162.
- Gaut, B. 2003. Creativity and imagination. *The Creation of Art* 148–173.
- Grassian, S., and Friedman, N. 1986. Effects of sensory deprivation in psychiatric seclusion and solitary confinement. *International Journal of Law and Psychiatry* 8(1):49–65.
- Gregor, K.; Danihelka, I.; Graves, A.; and Wierstra, D. 2015. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*, 1462–1471.
- Hamming, R. W. 1980. The unreasonable effectiveness of mathematics. *American Mathematical Monthly* 87:81–90.
- Hawkins, J., and Blakeslee, S. 2007. *On intelligence*. Macmillan.
- Heath, D., and Ventura, D. 2016. Creating images by learning image semantics using vector space models. In *Proceedings of The Thirtieth AAAI Conference on Artificial Intelligence*.
- Heath, D.; Dennis, A.; and Ventura, D. 2015. Imagining imagination: A computational framework using associative memory models and vector space models. In *Proceedings of the 6th International Conference on Computational Creativity*, 244–251.
- Hoffman, D. D. 2000. *Visual Intelligence: How We Create What We See*. W.W. Norton.
- Hong, K.; Chalup, S. K.; King, R.; Ostwald, M. J.; et al. 2013. Scene perception using pareidolia of faces and expressions of emotion. In *IEEE Symposium on Computational Intelligence for Creativity and Affective Computing*, 79–86.
- Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, 675–678.
- Kamel, H. M., and Landay, J. A. 2000. A study of blind drawing practice: creating graphical information without the visual channel. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, 34–41.
- Kennedy, J. M. 1993. *Drawing & the Blind: Pictures to Touch*. Yale University Press.
- Lafer-Sousa, R.; Hermann, K. L.; and Conway, B. R. 2015. Striking individual differences in color perception uncovered by ‘the dress’ photograph. *Current Biology* 25(13):R545–R546.
- Leon A. Gatys, A. S. E., and Bethge, M. 2015. A neural algorithm of artistic style. *Computing Research Repository*.
- Levi, G., and Hassner, T. 2015. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 34–42.
- Machado, P.; Romero, J.; and Manaris, B. 2007. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Berlin: Springer. 381–415.
- Marmor, M. F., and Ravin, J. 2009. *The Artist’s Eyes*. Harry N Abrams Incorporated.
- Melo, A. F., and Wiggins, G. 2003. A connectionist approach to driving chord progressions using tension. In *Proceedings of the AISB Symposium on Creativity in Arts and Science*.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.
- Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity*, 119–125.
- Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku: Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 32–39.
- Nguyen, A. M.; Yosinski, J.; and Clune, J. 2015. Innovation engines: Automated creativity and improved stochastic optimization via deep learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 959–966.
- Pachet, F., and Roy, P. 2011. Markov constraints: Steerable generation of Markov sequences. *Constraints* 16(2):148–172.
- Parkhi, O. M.; Vedaldi, A.; and Zisserman, A. 2015. Deep face recognition. *British Machine Vision* 1(3):6.
- Peterson, E. M. 2006. Creativity in music listening. *Arts Education Policy Review* 107(3):15–21.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.
- Sacks, O. 1995. *An Anthropologist on Mars*. New York: Knopf.
- Searle, J. R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(03):417–424.
- Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.
- Varshney, L. R.; Pinel, F.; Varshney, K. R.; Schörgendorfer, A.; and Chee, Y.-M. 2013. Cognition as a part of computational creativity. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, 36–43.
- Weiskrantz, L. 1996. Blindsight revisited. *Current Opinion In Neurobiology* 6(2):215–220.
- Zwaan, R. A., and Kaschak, M. P. 2008. Language in the brain, body, and world. *The Cambridge Handbook of Situated Cognition* 368–381.

Creative Generation of 3D Objects with Deep Learning and Innovation Engines

Joel Lehman

Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
lehman.154@gmail.com

Sebastian Risi

Center for Computer Games Research
IT University of Copenhagen
Copenhagen, Denmark
sebr@itu.dk

Jeff Clune

Department of Computer Science
University of Wyoming
Laramie, Wyoming, USA
jeffclune@uwyo.edu

Abstract

Advances in supervised learning with deep neural networks have enabled robust *classification* in many real world domains. An interesting question is if such advances can also be leveraged effectively for computational *creativity*. One insight is that because evolutionary algorithms are free from strict requirements of mathematical smoothness, they can exploit powerful deep learning representations through arbitrary computational pipelines. In this way, deep networks trained on typical supervised tasks can be used as an ingredient in an evolutionary algorithm driven towards creativity. To highlight such potential, this paper creates novel 3D objects by leveraging feedback from a deep network trained only to recognize 2D images. This idea is tested by extending previous work with *Innovation Engines*, i.e. a principled combination of deep learning and evolutionary algorithms for computational creativity. The results of this automated process are interesting and recognizable 3D-printable objects, demonstrating the creative potential for combining evolutionary computation and deep learning in this way.

Introduction

There have recently been impressive advances in training deep neural networks (DNNs; Goodfellow, Bengio, and Courville 2016) through stochastic gradient descent (SGD). For example, such methods have led to significant advances on benchmark tasks such as automatic recognition of images and speech, sometimes matching human performance. (He et al. 2015). While impressive, such advances have generally been limited to *supervised* classification tasks in which a large number of labeled examples is available. Such a process cannot readily create interesting, unexpected outputs.

As a result, DNNs have not precipitated similar advances in *creative* domains. Creating new artifacts does not fit naturally into the paradigm of SGD, because (1) creativity often lacks a clear error signal and (2) creative systems are often non-differentiable as a whole, i.e. they may encompass arbitrary computation that lacks the mathematical smoothness necessary for SGD. Combining evolutionary algorithms (EAs) with DNNs can remedy both issues. One powerful such combination is explored in this paper: The latent knowledge of the DNN can be leveraged as a reward signal for an EA; and evaluation in an EA can freely incorporate arbitrary computation.

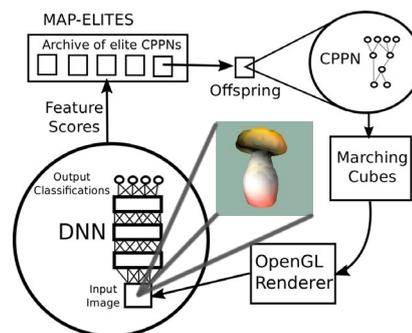


Figure 1: **Approach.** Each iteration a new offspring CPPN is generated, which is used to generate a 3D vector field. The marching cubes algorithm extracts a 3D model from this field, which is rendered from several perspectives. The rendered images are input into an image-recognition DNN, and its output confidences supply selection signals for MAP-Elites, thereby driving evolution to find 3D objects that the DNN increasingly cannot distinguish from real-world ones.

Building upon Nguyen, Yosinski, and Clune (2015b), the main idea in this paper is that classification DNNs can be coupled with creative EAs to enable *cross-modal* content creation, where a DNN's knowledge of one modality (e.g. 2D images) is exploited to create content in another modality (e.g. 3D objects). While the EA in Nguyen, Yosinski, and Clune (2015b) created images from an image-based DNN, SGD can also do so (Yosinski et al. 2015). In contrast, the system described here creates *3D models* from a *2D image* recognition DNN, making use of a non-differentiable rendering engine and an expressive evolutionary computation (EC) representation called a compositional pattern-producing network (CPPN; Stanley 2007). In this way, the unique advantages of both EAs and DNNs are combined: EAs can leverage compressed genetic representations and evaluate individuals in flexible ways (e.g. bridging information modalities), while DNNs can create high-level object representations as a byproduct of supervised training.

The paper's approach (Figure 1) combines an image-recognition DNN with a diversity-generating EA (MAP-Elites; Mouret and Clune 2015), to realize an *Innovation Engine* (Nguyen, Yosinski, and Clune 2015b). The extension here is that the genetic encoding of EndlessForms.com (Clune and Lipson 2011) enables automatic sculpting of 3D

models. In particular, evolved objects are rendered with a 3D engine from multiple perspectives, resulting in 2D images that can be evaluated by the DNN. The classification outputs of the DNN can then serve as selection pressure for MAP-Elites to guide search. In this way, it is possible to create novel 3D models from a DNN, enabling creative synthesis of new models from only labeled images.

The product of running this system is a collection of 3D objects from 1,000 categories (e.g. banana, basketball, bubble), many of which are indistinguishable to the DNN from real-world objects. A chosen set of objects is then 3D printed, showing the possibility of automatic production of novel real-world objects. Further, a user study of evolved artifacts demonstrates that there is a link between DNN confidence and human recognizability. In this way, the results reveal that Innovation Engines can automate the creation of interesting and often recognizable 3D objects.

Background

The next section first reviews previous approaches to creative generation of objects. After that, deep learning and MAP-Elites are described; together they form a concrete implementation of the Innovation Engine approach, which is applied in this paper's experiments and is reviewed last.

Creative Object Generation

EndlessForms.com (Clune and Lipson 2011) is a collaborative interactive evolution website similar to Picbreeder.org (Secretan et al. 2008), but where users collaborate to evolve diverse 3D objects instead of 2D images. Using the same genetic encoding, this paper attempts to automate the creativity of EndlessForms.com users, similarly to how Innovation Engines were originally applied (Nguyen, Yosinski, and Clune 2015b) to automate the human-powered collaborative evolution in Picbreeder (Secretan et al. 2008). Importantly, this work builds upon previous approaches that exploit combinations of ANNs and EC (Baluja, Pomerleau, and Jochem 1994) or classifiers and EC (Correia et al. 2013; Machado, Correia, and Romero 2012) for automatic pattern-generation. Other similar approaches have applied EAs in directed ways to evolve objects with particular functionalities, like tables, heat-sinks, or boat hulls (Bentley 1996).

Shape grammars are another approach to generating models (Stiny and Gips 1971), where iteratively-applied grammatical rules enable automatic creation of models of a particular family. However, such grammars are often specific to particular domains, and require human knowledge to create and apply. The procedural modeling community also explores methods to automatically generate interesting geometries (Yumer et al. 2015), although such approaches are also subject to similar constraints as shape grammars.

Perhaps the most similar approach is that of Horn et al. (2015), where a user-supplied image is analyzed through four metrics, and a vase is shaped through an EA to match such characteristics. Interestingly, the approach here could be adapted in a similar direction to create sculptures inspired by user-provided images (by matching DNN-identified features instead of hand-designed ones), or even to create novel sculptures in the style of famous artists or sculptors (Gatys,

Ecker, and Bethge 2015); such possibilities are described in greater detail in the discussion section.

Deep Learning

Although the idea of training multi-layer neural networks through back-propagation of error is not new, advances in computational power, in the availability of data, and in the understanding of many-layered ANNs, have culminated in a high-interest field called *deep learning* (Goodfellow, Bengio, and Courville 2016). The basic idea is to train many-layered (deep) neural networks on big data through SGD.

Deep learning approaches now achieve cutting-edge performance in diverse benchmarks, including image, speech, and video recognition; natural language processing; and machine translation (Goodfellow, Bengio, and Courville 2016). Such techniques are generally most effective when the task is *supervised*, i.e. the objective is to learn a mapping between given inputs and outputs, and when training data is ample. Importantly, the output of the DNN (and the error signal) must be composed only from differentiable operations.

One focus of deep learning is object recognition, for which the main benchmark is the ImageNet dataset (Deng et al. 2009). ImageNet is composed of millions of images, labeled from 1,000 categories spanning diverse real-world objects, structures, and animals. DNNs trained on ImageNet are beginning to exceed human levels of performance (He et al. 2015), and the learned feature representations of such DNNs have proved useful when applied to other image comprehension tasks (Razavian et al. 2014). In this paper, DNNs are applied to sculpt 3D objects by providing feedback to the MAP-Elites EA, which is described next.

MAP-Elites

While most EAs are applied as optimization algorithms, there are also EAs driven instead to collect a wide *diversity* of high-quality solutions (Pugh et al. 2015; Laumanns et al. 2002; Saunders and Gero 2001). Because of their drive towards diverse novelty, such algorithms better fit the goals of computational creativity than EAs with singular fixed goals.

One simple and effective such algorithm is the multi-dimensional archive of phenotypic elites (MAP-Elites) algorithm (Mouret and Clune 2015), which is designed to return the highest-performing solution for each point in a space of user-defined feature dimensions (e.g. the fastest robot for each combination of different heights and weights). The idea is to instantiate a large space of inter-related problems (1,000 in this paper), and use the current-best solutions for each problem as stepping stones to reach better solutions for any of the other ones. That is, solutions to easier problems may aid in solving more complex ones. Note that only a cursory description of MAP-Elites is provided here; Mouret and Clune (2015) provides a complete introduction.

MAP-Elites requires a domain-specific measure of performance, and a mapping between solutions and the feature space. For example, if the aim is to evolve a variety of different-sized spherical objects, the performance measure could be a measure of roundness, while the feature space dimension could be object size. In this way, MAP-Elites has a mechanism to separate the quality criterion (e.g. roundness)

from dimension(s) of desired variation (e.g. size). In practice, because the feature space is often continuous, it is first discretized into a finite set of *niches*.

A map of elite solutions is then constructed, that maintains the current elite solution and its corresponding performance score for each niche. When a new solution is evaluated, it is mapped to its niche, and its performance is compared to that of its niche's current elite. If the newly-evaluated solution scores higher than the old elite individual, it replaces the old elite in the niche, and the niche's score is updated accordingly.

Evolution is initialized with an empty map, which is seeded by evaluating a fixed number of random solutions. A fixed budget of evaluations is then expended by repeatedly choosing a solution at random from the map, mutating it, and then evaluating it. After all evaluations have been exhausted, the final map is returned, which is the collection of the best solution found in each niche.

Innovation Engines

The MAP-Elites algorithm described above can be used to realize an Innovation Engine (Nguyen, Yosinski, and Clune 2015b). Like the DeLeNoX approach (Liapis et al. 2013), Innovation Engines combine (1) EAs that can generate and collect diverse novelty with (2) DNNs that are trained to distinguish novelty and evaluate its quality. The hope is that such an architecture can produce a stream of interesting creative artifacts in whatever domain it is applied to.

This paper builds on the initial implementation in Nguyen, Yosinski, and Clune (2015b), where a pretrained image recognition DNN is combined with MAP-Elites to automatically evolve human-recognizable images. In that work, the space of MAP-Elites niches was defined as the 1,000 object categories within the ImageNet dataset (Deng et al. 2009), which is a common deep learning benchmark task (note that the same space of niches is applied here). CPPNs that represent images (as in Picbreeder; Secretan et al. 2007) were evolved, and the performance measure for MAP-Elites was to maximize the DNN's confidence that an evolved image is of a specific object category. An evolutionary run thus produced a collection of novel images, many of which resembled creative depictions of real-world objects. The work was not only accepted into a competitive university art show, but won an award (Nguyen, Yosinski, and Clune 2015b). The work here expands upon such image evolution, applying a similar technique to evolve 3D objects.

Note that the current version of Innovation Engines can be seen in Boden's terminology as realizing *exploratory creativity* but not *transformational creativity* (Boden 1996). That is, while the algorithm has a broadly expressive space of images or objects to search through, its conception of what objects are interesting and why they are interesting is fixed. In the future, unsupervised deep learning may provide a mechanism to extend innovation engines with aspects of transformational creativity (Nguyen, Yosinski, and Clune 2015b); for example, the DeLeNoX system uses unsupervised autoencoder neural networks to iteratively transform its creative space (Liapis et al. 2013).

Approach

While ideally advances in deep learning would also benefit computational creativity, creative domains often encompass arbitrary computation and reward signals that are not easily combined with the gradient descent algorithm. The approach here is thus motivated by the insight that EAs, unlike DNNs, are not limited to pipelines of computation in which each stage is differentiable. In particular, one interesting possibility enabled by EAs is to exploit the latent knowledge of the DNN to create structures with entirely different modality than with which the DNN's was trained.

For example, it is not clear how SGD can extract 3D objects from an image-recognition network, because there is no natural differentiable mechanism to translate from a 3D representation of an object to the 2D pixel representation used by image-recognition DNNs. In contrast, EC representations of 3D objects can be rendered to 2D pixel images through non-differentiable rendering engines; and the resulting images can interface with trained image-recognition DNNs. While it might be possible to train a 3D object recognition DNN (e.g. with necessary technical advances and an appropriate dataset), there are diverse cross-modal possibilities that EAs enable (particular examples can be found in the discussion section). In other words, this idea provides a general mechanism for creative cross-modal linkage between EAs and DNNs, which respects the advantages of both methods: EAs do not require differentiability, while DNNs better leverage big data and computational efficiency to learn powerful hierarchies of abstract features.

This paper realizes a proof-of-concept of cross-modal linkage, shown in Figure 1, wherein 3D objects are represented with the EndlessForms.com encoding (Clune and Lipson 2011). This encoding represents a mapping from 3D coordinate space to material density, by using a CPPN (which is similar to a neural network function approximator). Inspired by regularities of biological organisms, activation functions in such CPPNs are drawn from a set chosen to reflect such regularities, thereby enabling representing complex patterns compactly.

In more detail, the CPPN takes as input Cartesian coordinates and generates as its output the density of material to be placed in that coordinate. The CPPN is then queried systematically across the 3D coordinate space, resulting in a 3D scalar field. Next, the marching cubes algorithm (Lorensen and Cline 1987) constructs a mesh that wraps the scalar field, by defining the object's boundary as a threshold of material density. Note that the EndlessForms.com encoding is extended here to enable more detailed models that vary in color across their surface. To accomplish this effect, outputs are added to the CPPN that specify the HSV color of each voxel, enabling the creation of objects with detailed colors.

To evaluate an individual, this encoding is combined with a rendering engine that produces several rendered images of the encoded object from different perspectives. Then these rendered images are input into a pretrained DNN to produce performance signals for the MAP-Elites EA. The chosen DNN is the BLVC reference GoogleNet from the Caffe model zoo (Jia et al. 2014), a freely-available DNN similar in architecture to GoogLeNet (Szegedy et al. 2015).

As in Nguyen, Yosinski, and Clune (2015b), the underlying MAP-Elites algorithm’s space of niches is defined by the 1,000 discrete object classes that the DNN can recognize (which span household objects, animals, and vehicles). Performance for each niche is defined as the DNN’s confidence that the generated artifact is an example of the class the niche represents. In particular, the confidences of the six renders for each class are multiplied together; this was shown in preliminary experiments to improve performance.

Rendering Improvements

To improve the render quality of the 3D objects, two additions to the algorithm are considered: (1) enabling lighting and material properties of the object to evolve, and (2) enabling the background color to evolve. Overall, rendering quality is important because the DNN is trained on real-world photographs, and therefore may rely on features such as lighting or background context to discriminate between objects. As a result, the success of the approach may depend on the kinds of images that are possible or easy to represent.

For this reason, in addition to the CPPN, the genome has four evolvable floating-point numbers that encode parameters of lighting (the diffuse and ambient intensities) and the object’s material (its shininess and its specular component); and three evolvable parameters that encode the HSV of the background color. All such parameters have a fixed chance of mutating when a new offspring is produced.

These extensions enable evolution to control aspects of a rendered image unrelated to evolving a 3D object. For example, adjusting the background color can help control for a superficial discriminative feature that may always correlate with the presence of certain objects. For example, fish may nearly always be found in the context of water, and so a DNN may only recognize an evolved object as a fish if it is rendered against a blue background.

Search Improvements

To improve the effectiveness of the underlying MAP-Elites search process, two additions are considered: (1) adding niches that represent more general classes of objects to enable incremental learning; and (2) biasing search away from exploring niches that produce fewer innovations.

Previous work found it was difficult to evolve images for very specific classes (e.g. nuanced breeds of dogs; Nguyen, Yosinski, and Clune 2015b), which preliminary experiments confirmed was also problematic for evolving 3D objects. Because of how supervised training of DNNs generally works, the 1,000 target categories in ImageNet only represent the finest level of granularity. That is, SGD works most easily when an image is associated with only one label, even when broader categorical information is available. Because the niches for MAP-Elites are directly imported from the DNN’s target categories, the EA must directly evolve towards specific nuanced categories. However, learning general concepts, e.g. distinguishing a dog from other animals, often provides scaffolding for learning more nuanced ones, e.g. distinguishing a pug from a french bulldog.

Thus, the idea is to artificially create more general niches by aggregating the specific categories together. Because the

WordNet database (Miller 1995) underlies the categories of images in ImageNet, hierarchical relations in WordNet can be leveraged to cluster semantically similar categories. In particular, a tree is constructed consisting of all the ImageNet categories, with directed edges added for each hypernym relationship, from the more specific class to the more general one. Non-leaf nodes in this graph thus represent increasingly general concepts, which can be added to MAP-Elites to augment the more specific ones. Given the classification outputs of the DNN for a particular rendered image, the score for any of the added niches is calculated for by summing the confidences of all the leaf nodes beneath it, i.e. all the hyponym nodes. Because individuals can be maintained that maximize general concepts, additional pathways for incremental learning are enabled for evolution.

The second addition biases MAP-Elites away from expending resources on niches that have proved unproductive. In particular, each MAP-Elites niche is augmented with a decrementing counter. Each counter is initialized to a fixed value (10 in the experiments here) that determines the relative probability of choosing that niche for reproduction.

A niche’s counter is decremented when an offspring generated from the niche’s current champion does not replace any existing individuals in the map of elites (i.e. the niche is penalized because it did not lead to an innovation). If instead the offspring displaces other champions, then the counters for the initial niche and the niches of all displaced champions are reset to their initial maximum value.

Experimental Setup

The basic setup is replicated from Nguyen, Yosinski, and Clune (2015b), wherein the MAP-Elites algorithm is driven by the classification outputs of a DNN. However, the DNN here processes several renderings of a 3D object instead of a single image. In particular, the object is rendered six times, successively rotated by 45 degrees increments around its y-axis (yaw). The motivation is to encourage the evolution of objects that resemble the desired class when viewed head-on from a variety of perspectives. Every alternating rendering is also rotated 5 degrees around its x-axis (pitch), to encourage further robustness. The voxel field for the EndlessForms.com encoding is given a resolution of 20 x 20 x 20 units, striking a balance between possible model detail and the computational cost of querying the CPPN for each voxel. Full source code, experimental results (including downloadable model files, and renders from all six evaluated perspectives), and user study data are freely available from the project website: <http://jal278.github.io/iccc2016/>.

Ablation Experiments

One practical concern is that evaluation of an individual is expensive computationally, as it requires (1) querying a CPPN 8,000 times to generate the $20 \times 20 \times 20$ scalar field from which marching cubes produces a model, (2) rendering an image of that resulting model multiple times (here, 6), and (3) evaluating the rendered images with a large DNN. The DNN evaluation in particular is the computational bottleneck and depends upon capable GPUs to be time-efficient.

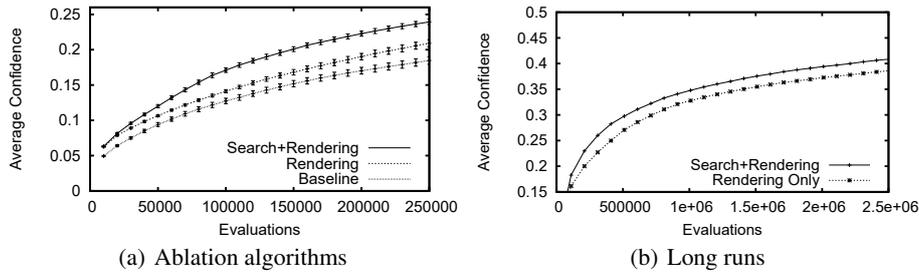


Figure 2: **Performance of evolutionary runs.** (a) The performance of incremental ablations of the proposed algorithm is shown averaged over ten independent runs. Performance is measured by averaging the confidence of the DNN first over all renderings of a particular object, and then over all object classes. All distinctions between methods are significant ($p < 0.05$), highlighting that the improvements both to lighting and to the search algorithm facilitate evolving objects that the DNN more confidently classifies. Note that error bars reflect standard error. (b) The performance over evaluations of the two long runs is shown, anecdotally highlighting how including both search and rendering additions appears also to result in long-term improvements. Many niches evolve towards high-confidence (18.2% of niches in $S+R$ and 15.1% of niches in R only achieved confidence scores > 0.85 when averaged over object renderings). However, evolution in niches representing highly-specific objects (e.g. one breed of dog) or geometrically-complex ones (e.g. a grocery store) often stagnates.

As a result, performing a single run to highlight the full potential of the system (2.5 million evaluations) took around three weeks on the hardware available to the experimenters (a modern gaming laptop with a high-quality GPU).

To inform the longer runs, a series of shorter ablation runs (250,000 evaluations each) were first performed to validate algorithmic features. In particular, three algorithms were compared: a baseline (control) algorithm, the same algorithm augmented with the rendering improvements, and an algorithm with both the rendering and search improvements. The idea is to examine whether the added features result in better performance, thereby providing guidance for what algorithm should be applied when generating the main results.

Ablation Results

The results of the comparison are shown in Figure 2(a). The order of final performance levels achieved by the algorithmic variants reflects that adding the tested components significantly improves performance ($p < 0.05$; all tests are Mann-Whitney U-tests unless otherwise specified).

Both the rendering and search improvements comprise multiple sub-improvements; to better understand each component’s relative contribution, shorter ablation experiments were conducted (10 independent runs of 70,000 evaluations each). For search improvements, pruning unproductive niches provided greater benefit than did only including more general niches ($p < 0.05$), but both improved performance over the control algorithm ($p < 0.05$). For rendering improvements, allowing the background color to evolve significantly increased performance, while allowing lighting to evolve did not ($p < 0.05$). However, because they do not decrease performance, and because preliminary experiments revealed that lighting enabled more interesting aesthetic effects, lighting changes are included in the full experiments.

Main Experimental Results

Informed by the ablation experiments, two long runs were conducted (2.5 million evaluations). To verify anecdotally that the conclusions from the ablation experiments are likely to generalize to such longer runs, one run, called $S+R$, in-

cluded the full suite of improvements (i.e. both search and rendering), while the other run, R only, included only the rendering improvements. The gross performance characteristics of the long runs are shown in Figure 2(b), and suggest that the algorithmic additions result in performance gains that persist even over runs with many more evaluations.

A curated gallery of high-confidence evolved objects is shown in Figure 3. To highlight the quality of learned object representations, mutations of selected objects are shown in Figure 4. Overall, the objects exhibit an interesting diversity and in most cases the connection between the object and the real world object class is evident.

3D Printing the Automatically Generated Objects

Because the output of the creative process are textured 3D models, it is possible to bring them into reality through 3D printing. A small selection of evolved objects was chosen from the results of both runs. In particular, objects were chosen that (1) were possible to print, (2) were colorful, and (3) highlighted interesting features used by DNNs to classify images. Model files were uploaded to the Shapeways commercial 3D printing service to be printed in their color sandstone material. The results of this process are shown in Figure 5. Note that many of the objects were too fragile to directly be printed (because their structures were too thin in particular areas). However, optimization criteria could be injected into the search process to mitigate such fragility.

To test the fidelity of the 3D printing process, the above photographs (without inlays) were also input to the training DNN, and the resulting highest-scored categories were recorded. The evolved Starfish was correctly classified by the DNN’s first choice, while the Mushroom was classified first as a bolete (a type of wild mushroom), and the sixth ranked choice was the broader (correct) mushroom class. The Jellyfish was classified by first choice as a conch, and as jellyfish by the network’s fifth choice. The Hammerhead was interestingly classified first as a *hammer*, and as eighth choice by the true hammerhead shark label. Finally, the Goldfinch was classified by fifth choice as a lorikeet, and as ninth choice a hummingbird; both also are colorful birds.

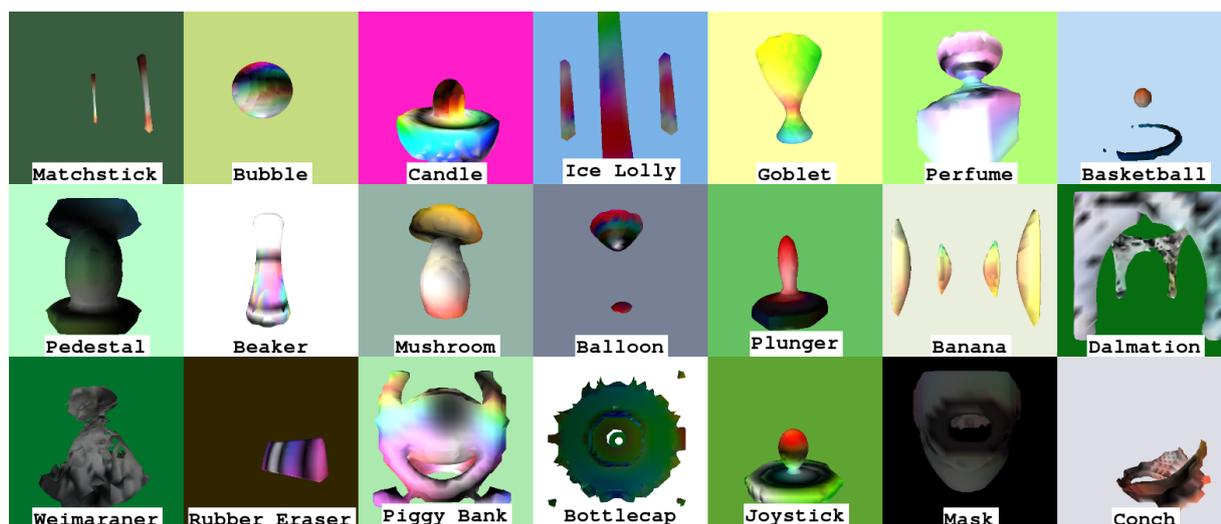


Figure 3: **Gallery of automatically generated high-confidence objects.** A curated selection of high-confidence champions from both the *S+R* and *R only* runs. Representing multiple copies of the same object (e.g. Banana, Ice Lolly, and Matchstick) helps maximize DNN confidence. The system often evolves roughly rotationally-symmetric objects (e.g. Goblet, Joystick, Bubble), both because many classes of real-world objects are symmetric in such a way and because it is the easiest way to maximize DNN confidence from all rendered perspectives. However, objects such as Conch, Mask, and Dalmatian show that asymmetric and more complex geometries can also evolve when necessary to maximize DNN confidence. Overall, the results show the promise of Innovation Engines for cross-modal creativity. Best viewed in color.

The conclusion is that even after crossing the reality gap, key features of objects are still be recognized by the DNN.

User Study

A user study was conducted to explore whether humans saw the resemblance of the evolved objects to their respective categories. In particular, 20 fixed survey questions were created by sampling from niches within which evolution successfully evolved high-confidence objects.

Each question asked the user to rank three images by their resemblance to the sampled category. One image was a rendering of an evolved object classified by the DNN with *high confidence* (i.e. the highest-confidence rendering for the intended category; always > 0.95). A second image was a rendering classified with *moderate confidence* (i.e. the rendering with score closest to 0.2, which is still qualitatively distinguished from the base expectation of 0.001). The third image was of an evolved object classified with high confidence as belonging to an arbitrarily-chosen *distractor* category. The idea is to see whether user rankings of the objects' resemblance to the true class agree with the DNN's ranking (i.e. high confidence, moderate confidence, distractor).

Twenty-three subjects were recruited using a social media post to fill out an online survey; the order of questions was fixed, but the order of images within each question was randomized. Users generally ranked images in an order similar to that of the DNN (Figure 6); the conclusion is that high-confidence objects generally bear semantic resemblance to the category they are optimized to imitate.

Discussion

The basic framework presented here could be used with DNNs trained on other image datasets to generate distinct

types of 3D objects and scenes. For example, combining a DNN trained on the Places dataset (Zhou et al. 2014) with Google's deep dream visualization (Mordvintsev, Olah, and Tyka 2015) resulted in images of fantastical architectures, highlighting the potential for architectural creativity embedded in such a DNN. Thus substituting this paper's approach for deep dream may likewise yield interesting 3D architectural creations. Similarly, DNNs trained to recognize other things could be leveraged to create diverse artifact types, e.g. 3D flowers, cars, or faces (pretrained DNNs for each such type of data are available from the Caffe model zoo).

The approach in this paper could also generalize to other kinds of cross-modal creation through non-differentiable computation. For example, a DNN trained to distinguish different speakers (Lee et al. 2009) could be leveraged to evolve parameters for speech synthesis engines, potentially resulting in diverse but realistic settings for speech synthesizers without human tuning. Another possibility is automatic creation of music; just as optimizing CPPN-based images led to more qualitatively interesting results than did optimization of a naive pixel-based representation (Nguyen, Yosinski, and Clune 2015b), optimizing CPPN-based representations of music (Hoover and Stanley 2009) fed through a music-recognition DNN (Lee et al. 2009) might similarly enable automatic generation of compositions with more coherent or interesting structure.

One possibility to enable more open-ended creativity would be to leverage high-level features encoded by the DNN to guide search, instead of only the classification labels. The novelty search algorithm (Lehman and Stanley 2011) could be applied to create objects that span the space of high-level representations. Because the features compos-

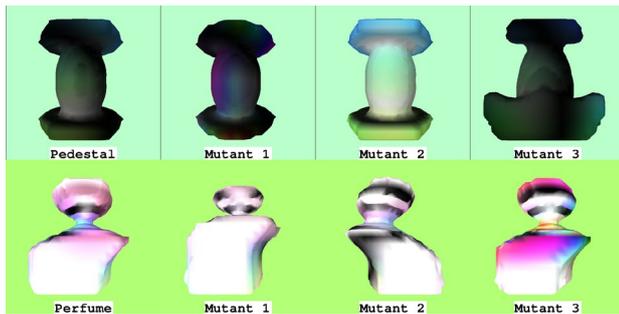


Figure 4: **Accessible variation from evolved objects.** First, twenty mutants of the Pedestal (top) and Perfume (bottom) champions were generated. For both rows, the original model is shown on the far left, followed to its right by three examples of interesting mutants. The conclusion is that in addition to the final objects, evolution also produces interesting object *representations* that can be leveraged for further creative purposes, e.g. interactive evolution.

ing the representation are constrained by their relation to classifying objects, exploration of such a space may yield a diversity of interesting objects. Conversely, the system could also be made more *directed* in interesting or interactive ways. For example, a novel 3D object might be optimized to mimic the high-level DNN features (Razavian et al. 2014) of a user-provided image, creating possibilities for human-machine artistic collaboration. Interestingly, such an approach could additionally be combined with the StyleNet objective function (Gatys, Ecker, and Bethge 2015) to sculpt objects inspired by a photograph, and cast in the *style* of a separate artwork or sculpture.

Finally, while created for computational creativity, the approach may also have implications for deployed deep learning systems. Nguyen, Yosinski, and Clune (2015a) suggested that DNNs may easily be fooled, given complete control of how an image is presented to the DNN. However, real world recognition systems may employ many (potentially unknown/unseen) cameras, which may preclude directly fooling such a system with a generated image. However, because evolved objects can be 3D printed, and because evolved objects are often recognized by diverse DNNs (data not shown), it may be possible to confound real-world deep learning recognition systems with such printed artifacts, even those based on multiple unseen cameras.

Conclusion

This paper introduced a framework for exploiting deep neural networks to enable creativity across modalities and input representations. Results from evolving 3D objects through feedback from an image recognition DNN demonstrate the viability of the approach: A wide variety of stylized, novel 3D models were generated that humans could recognize. The conclusion is that combining EC and deep learning in this way provides new possibilities for creative generation of meaningful and novel content from large labeled datasets.

Acknowledgements

Jeff Clune was supported by an NSF CAREER award (CAREER: 1453549).

References

- Baluja, S.; Pomerleau, D.; and Jochem, T. 1994. Towards automated artificial evolution for computer-generated images. *Connection Science* 6(2-3):325–354.
- Bentley, P. J. 1996. *Generic evolutionary design of solid objects using a genetic algorithm*. Ph.D. Dissertation, The University of Huddersfield.
- Boden, M. A. 1996. *Dimensions of creativity*. MIT Press.
- Clune, J., and Lipson, H. 2011. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. *Proceedings of the European Conference on Artificial Life*, See <http://EndlessForms.com> 144–148.
- Correia, J.; Machado, P.; Romero, J.; and Carballal, A. 2013. Evolving figurative images using expression-based evolutionary art. In *Proceedings of the fourth International Conference on Computational Creativity (ICCC)*, 24–31.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.
- Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.
- Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. Deep learning. Book in preparation for MIT Press.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.
- Hoover, A. K., and Stanley, K. O. 2009. Exploiting functional relationships in musical composition. *Connection Science* 21(2-3):227–251.
- Horn, B.; Smith, G.; Masri, R.; and Stone, J. 2015. Visual information vases: Towards a framework for transmedia creative inspiration. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 182.
- Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R. B.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, volume 2, 4.
- Laumanns, M.; Thiele, L.; Deb, K.; and Zitzler, E. 2002. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation* 10(3):263–282.
- Lee, H.; Pham, P.; Largman, Y.; and Ng, A. Y. 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, 1096–1104.
- Lehman, J., and Stanley, K. O. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation* 19(2):189–223.

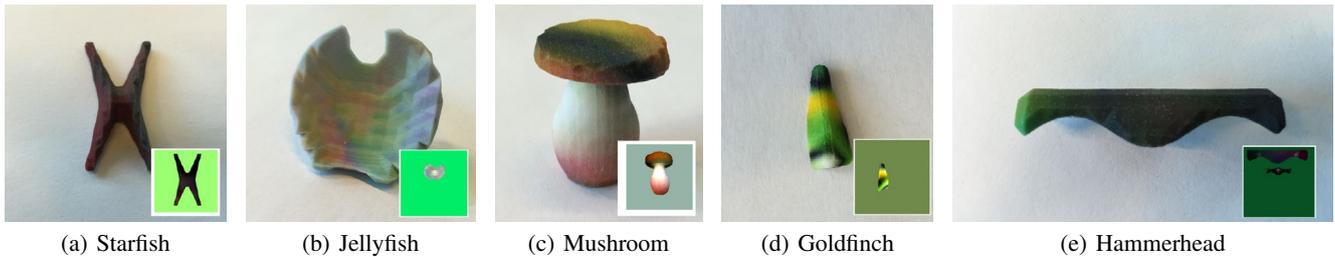


Figure 5: **Gallery of 3D printed objects.** Shown are photographs of 3D-printed objects oriented similarly to one of their simulated renders (which is inlaid). Note that the evolved Hammerhead consisted of two similar objects; one was chosen arbitrarily for 3D printing. In all cases, a clear resemblance is seen between each 3D-printed object and its render, demonstrating the feasibility of automatically generating real-world objects using the approach.

Liapis, A.; Martinez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the Fourth International Conference on Computational Creativity*, 56–63. AAAI Press.

Lorensen, W. E., and Cline, H. E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, 163–169. ACM.

Machado, P.; Correia, J.; and Romero, J. 2012. Expression-based evolution of faces. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer. 187–198.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going deeper into neural networks. *Google Research Blog*. Retrieved June 20.

Mouret, J.-B., and Clune, J. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015a. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 427–436. IEEE.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015b. Innovation engines: Automated creativity and improved stochastic optimization via deep learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*.

Pugh, J. K.; Soros, L.; Szerlip, P. A.; and Stanley, K. O. 2015. Confronting the challenge of quality diversity. In *Proc. of the Genetic and Evol. Comp. Conference*.

Razavian, A. S.; Azizpour, H.; Sullivan, J.; and Carlsson, S. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, 512–519. IEEE.

Saunders, R., and Gero, J. S. 2001. The digital clockwork muse: A computational model of aesthetic evolution. In *Proceedings of the AISB*, volume 1, 12–21.

Secretan, J.; Beato, N.; D’Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; and Stanley, K. O. 2008. Picbreeder: Collaborative interactive evolution of images. *Leonardo* 41(1):98–99.

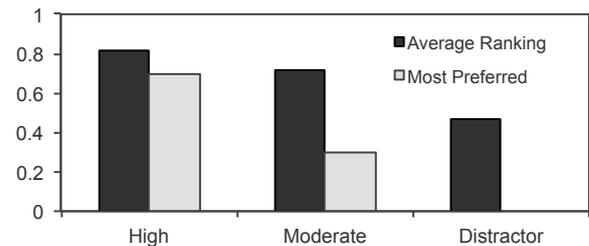


Figure 6: **User Study Results.** Black: The average rankings (normalized between 0 and 1) of each type of image (see text), aggregated across users and then questions. High and moderate confidence images are ranked significantly higher than distractor images ($p < 0.01$; Mann-Whitney U test). White: the percentage of questions in which users collectively ranked the image as *most* similar to the prompted category. Across all questions, the distractor is never the one that is most preferred across users. The conclusion is that user rankings often agree with the DNN.

Stanley, K. O. 2007. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines* 8(2):131–162.

Stiny, G., and Gips, J. 1971. Shape grammars and the generative specification of painting and sculpture. In *IFIP Congress (2)*, volume 2.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; and Lipson, H. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*.

Yumer, M. E.; Asente, P.; Mech, R.; and Kara, L. B. 2015. Procedural modeling using autoencoder networks. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 109–118. ACM.

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, 487–495.

Digits that are not: Generating new types through deep neural nets

Akın Kazakçı

MINES ParisTech,

PSL Research University, CGS-I3 UMR 9217

akin.kazakci@mines-paristech.fr

Mehdi Cherti and Balázs Kégl

LAL/LRI

CNRS/Université Paris-Saclay

{mehdi.cherti, balazs.kegl}@gmail.com

Abstract

For an artificial creative agent, an essential driver of the search for novelty is a value function which is often provided by the system designer or users. We argue that an important barrier for progress in creativity research is the inability of these systems to develop their own notion of value for novelty. We propose a notion of knowledge-driven creativity that circumvent the need for an externally imposed value function, allowing the system to explore based on what it has learned from a set of referential objects. The concept is illustrated by a specific knowledge model provided by a deep generative auto-encoder. Using the described system, we train a knowledge model on a set of digit images and we use the same model to build coherent sets of new digits that do not belong to known digit types.

Introduction

It is a widely accepted view in creativity research that creativity is a process by which novel *and* valuable combinations of ideas are produced (Runco and Jaeger 2012). This view bears a tension, the essence of which can be expressed by the following question: how to determine the value of novelty? If a new object is substantially different of the previous objects in its category, it might be hard to determine its value. On the contrary, if the value of an object can be readily determined, it might be the case that the object is not genuinely new. Indeed, there exist experimental results positing that novelty is a better predictor of creativity than the value (Diedrich et al. 2015) and that the brain processes novelty in a particular way (Beaucousin et al. 2011), suggesting that the relationship is far from trivial.

In art, the difficulty in determining the value of an object is omnipresent. An emblematic example is *Le Grand Verre* by Marcel Duchamp. The artist worked on this singular project from 1915 to 1923 and produced a groundbreaking yet enigmatic piece of art, which the critiques still continue to interpret in various ways. In 1934, Duchamp built *La boîte verte*, a green box containing preparatory material (notes, drawings, photographs) he produced for *Le Grand Verre*. Considered as a piece of art in its own right, the box was intended to assist and to explain *Le Grand Verre*, as would an exhibition catalog (Breton 1932).

In product design, there exist less enigmatic but still emblematic cases, where the value of an innovation could not

be easily determined. For instance, the first smartphone received significant criticism regarding its usability (e.g., no stylus was provided), and it was deemed to be less *evolved* than its counterparts. Beyond such problems related to the reception of novelty, the sheer difficulty in discovering new value has led companies to seek alternative approaches, such as input from lead users (Von Hippel 1986).

The difficulty in determining the value of novelty has particular implications from a computational perspective. How would a creative agent drive his search process towards novelty if its evaluation function has been predetermined? In practical implementations, we can find various manifestations of such fixed evaluation functions such as fitness functions or quantitative aesthetics criteria. These implementations fixate the kind of value the system can seek, once and for all in the beginning of the process. The creative outcome, if any, comes from an output whose perception was unexpected or unpredictable.

Theoretically, it may be argued that this can be solved by allowing the creative agent to change its own evaluation rules (Wiggins 2006; Jennings 2010). This implies that the system would be able to develop a preference for unknown and novel types of objects (Kazakçı 2014). In practice, this is implemented by interactive systems that use external feedback (e.g., the preferences of an expert) to guide the search process. Such systems explore user preferences about novelty rather than building their own value system. This is a shortcoming from the point of view of creativity (Kazakçı 2014).

An alternative approach might be to force the system to systematically explore *unknown* objects (Hatchuel and Weil 2009). This requires the system to function in a differential mode, where there is a need to define a reference of *known* objects. In other words, new kinds of values might be searched by *going-out-of-the-box* mechanisms which require the system to develop knowledge about a referential set of objects. In the absence of knowledge about such a set, creativity is reduced either to a combinatorial search or to a rule-based generative inference, both of which explore boundaries confined by the creator of the system and not the system itself. When such knowledge exists, the system can explore new types of objects by tapping into the *blind spots* of the knowledge model (Kazakci et al. 2010).

In this paper, we use a deep generative neural network



Figure 1: *Digits that are not*. Symbols generated using a deep neural net trained on a sample of hand-written digits from 0 to 9.

to demonstrate knowledge-driven creativity. Deep nets are powerful tools that have been praised for their capacity of producing useful and hierarchically organized representations from data. While the utility of such representations have been extensively demonstrated in the context of recognition (i.e., classification) far less work exists on exploring the generative capacity of such tools.

In addition, the goal of the little work on generative deep nets is to generate objects of *known types*, and the quality of the generator is judged by the visual or quantified similarity with existing objects (e.g., an approximate likelihood) (Theis, Oord, and Bethge 2015). In contrast, we use deep nets to explore their generative capacity beyond known types by generating unseen combinations of extracted features, the results of which are symbols that are mostly unrecognizable but seemingly respecting some implicit semantic rules of compositionality (Figure 1). What we mean by *features* is a key concept of the paper: they are not decided by the (human) designer, rather learned by an *autoassociative* coding-decoding process.

The novelty of our approach is two-fold. With respect to computational creativity models, our model aims at explicitly generating new types. We provide an experimental framework for studying how a machine can develop its own value system for new types of objects. With respect to statistical sample-based generative models, rather than a technical contribution, we are introducing a new *objective*: generate objects that are, in a deep sense, similar to objects in of the domain, but which use learned *features* of these objects to generate new objects which do not have the same *type*. In our case, we attempt to generate images that *could be* digits (e.g., in another imaginary culture), but which are *not*.

The first section, *Generative models for computational creativity*, describes our positioning with respect to some of the fundamental notions in creativity research in previous works. The section *Learning to generate* presents details about data-driven generative models and deep neural nets relevant to our implementation. The section *Generating from the learned model* describes our approach for exploring novelty through generation of new types, presents examples and comments. The section *Discussion and perspectives* discusses links with related research and points to further

research avenues. Finally, section *Summary* concludes.

Generative models for computational creativity

The purpose of a generative model

In the computational creativity literature, exploration of novelty has often been considered in connection with art (Boden and Edmonds 2009; McCormack et al. 2014). Despite various debates and nuances on terminology, such work has generally been categorized under the term *generative art* (or generative models). As defined by (Boden and Edmonds 2009), a generative model is essentially a rule-based system, albeit one whose output is not known in advance, for instance, due to non-determinism or to many degrees of freedom in the parameters of the systems (see also (Galanter 2012)). A large variety of such systems has been built, starting as early as the 90s (Todd and Latham 1991; Sims 1991), based on even earlier foundations (Nees 1969; Edmonds 1969). The definition, the complexity and the capabilities offered by such models evolved consistently. To date, several such models, including L-systems, cellular automata, or artificial life simulations, have been used in various contexts for the generation of new objects (i.e., drawings, sounds, or 3D printings) by machine. Such systems achieve an output perceived as creative by their users by opportunistically exploiting existing formal approaches that have been invented in other disciplines and for other purposes. Within this spirit, computational creativity research has produced a myriad of successful applications on highly complex objects, involving visual and acoustic information content.

In contrast, this work considers much simpler objects since we are interested, above all, in the clarification of notions such as novelty, value, or type, and in linking such notions with the solid foundations of statistics and machine learning. These notions underlie foundational debates on creativity research. Thus, rather than producing objects that might be considered as artistic by a given audience, our purpose is to better define and explicate a minimalist set of notions and principles that would hopefully lead to a better understanding of creativity and enable further experimental studies.

The knowledge of a generative system

The definition of a generative model as a rule-based system (Boden and Edmonds 2009) induces a particular relationship with knowledge. It is fair to state that such formalized rules are archetypes of consolidated knowledge. If such rules are hard-coded into the creative agent by the system designer, the system becomes an inference engine rather than a creativity engine. By their very nature, rules embed knowledge about a domain and its associated value system that comes from the system designer instead of being discovered by the system itself.

Allowing the system to learn its own rule system by examining a set of objects in a given domain resolves part of this problem: the value system becomes dependent on the learning algorithm (instead of the system designer). In our system, we use a learning mechanism where the creative agent is forced to *learn to disassemble and reconstruct* the examples it has seen. This ensures that the utility of the features and the transformations embedded within the rules learned by the system are directly linked to a capacity to construct objects. As we shall see, the particular deep neural net architecture we are using is not only able to *reconstruct known* objects: it can also *build new* and *valuable* objects using their hierarchically organized set of induced transformations.

Knowledge-driven exploration of value

Today, more often than not, generative models of computational creativity involve some form of a biological metaphor, the quintessence of which is evolutionary computation (McCormack 2013). Contrary to human artists who are capable of exploring both novelty and the value of novelty, such computational models often consider the generation of novelty for a value function that is independent of the search process. Either they operate based on a fixed set of evaluation criteria or they defer evaluation to outside feedback. For the former case, a typical example would be a traditional fitness function. For the later case, a typical example would be an interactive genetic algorithm (Takagi 2001) where the information about value is provided by an oracle (e.g., a human expert). In both cases, the system becomes a construction machine where the generation of value is handled by some external mechanism and not by the system itself. This can be considered as a fundamental barrier for computational creativity research (Kazakçı 2014) that we shall call *fitness function barrier*.

(Parikka 2008) summarizes the stagnation that this approach causes for the study of art through computers: “... if one looks at several of the art pieces made with genetic algorithms, one gets quickly a feeling of not ‘nature at work’ but a Designer that after a while starts to repeat himself. There seems to be a teleology anyhow incorporated into the supposed forces of nature expressed in genetic algorithms practice ‘a vague feeling of disappointment surrounds evolutionary art’”.

The teleology in question is a direct consequence of fitness function barrier and the hard-coded rules. In our system, we avoid both issues by using a simple mechanism that enables the system to explore novel objects with novel values. Given a set of *referential objects* $\mathcal{D} = \{x_1, \dots, x_n\}$

whose *types* $\mathcal{T} = \{t_1, \dots, t_k\}$ are *known* (or can be determined by a statistical procedure such as clustering), the system is built in such a way that it generates objects $\mathcal{D}' = \{x'_1, \dots, x'_m\}$ with types $\mathcal{T}' = \{t'_1, \dots, t'_\ell\}$ such that $\mathcal{D}' \not\subseteq \mathcal{D}$ and $\mathcal{T}' \not\subseteq \mathcal{T}$. In other words, the system builds a set of new objects, some of which have new types. While the current system does not develop a preference function over the novelty it generates, the current setup provides the necessary elements to develop and experiment with what might be a value function for the unknown types. At any rate, the generation of unknown types of objects is an essential first step for a creative system to develop its own evaluation function for novelty and to become a designer itself.

Learning to generate

Data-driven generative models

In contrast to computational creativity research that aims to generate new object descriptions, disciplines such as statistics and machine learning strive to build solid foundations and formal methods for modeling a given set of object descriptions (i.e., *data*). These disciplines do not consider the generation of data as a scientific question: the data generating process is considered fixed (given) but unknown. Nevertheless, these fields have developed powerful theoretical and practical formal tools that are useful to scientifically and systematically study what it means to generate novelty.

In fact, generative models have a long and rich history in these fields. The goal of generative models in statistics and machine learning is to sample from a fixed but unknown *probability distribution* $p(x)$. It is usually assumed that the algorithm is given a *sample* $\mathcal{D} = \{x_1, \dots, x_n\}$, generated independently (by nature or by a simulator) from $p(x)$. There may be two goals. In classical *density estimation* the goal is to estimate p in order to evaluate it later on any new object x . Typical uses of the learned density are *classification* (where we learn the densities \hat{p}_1 and \hat{p}_2 from samples \mathcal{D}_1 and \mathcal{D}_2 of two *types* of objects, then compare $\hat{p}_1(x)$ and $\hat{p}_2(x)$ to decide the type of x), or *novelty* (or *outlier detection*) (where the goal is to detect objects from a stream which do not look like objects in \mathcal{D} by thresholding $\hat{p}(x)$).

The second goal of statistical generative models is to *sample* objects from the generative distribution p . If p is known, this is just random number generation. If p is unknown, one can go through a first density estimation step to estimate \hat{p} , then sample from \hat{p} . The problem is that when x is high-dimensional (e.g., text, images, music, video), density estimation is a hard problem (much harder than, e.g., classification). A recent line of research (Hinton, Osindero, and Teh 2006; Salakhutdinov and Hinton 2009) attempts to generate from p without estimating it, going directly from \mathcal{D} to novel examples. In this setup, a formal generative model g is a function that takes, as input, a random seed r , and generates an object $x = g(r)$. The learning (a.k.a, training or building) process is a (computational) function \mathcal{A} that takes, as input, a data set \mathcal{D} , and outputs the generative model $g = \mathcal{A}(\mathcal{D})$.

The fundamental problem of this latter approach is very similar to the main question we raised about computational creativity: what is the value function? When the goal is den-

sity estimation, the value of \hat{p} is formally $\sum_{x \in \mathcal{D}'} \log \hat{p}(x)$, the so-called *log-likelihood*, where \mathcal{D}' is a second data set, independent from \mathcal{D} which we used to build (or, in machine learning terminology, to *train*) \hat{p} . When p is unknown, evaluating the quality of a generated object $x = g(r)$ or the quality of a sample $\hat{\mathcal{D}} = \{g(r_1), \dots, g(r_n)\}$ is an unsolved research question in machine learning as well.

There are a few attempts to formalize a quantitative goal (Goodfellow et al. 2014), but most of the time the sample $\hat{\mathcal{D}}$ is evaluated visually (when x is an image) or by listening to the generated piece of music. And this is tricky: it is trivial to generate exact objects from the training set \mathcal{D} (by random sampling), so the goal is to generate samples that are *not* in \mathcal{D} , but which *look like* coming from the *type* of objects in \mathcal{D} . By contrast, our goal is to generate images that look like *digits* but which do not come from digit *types* present in \mathcal{D} .

Deep neural networks

In the machine learning literature, the introduction of deep neural networks (DNNs) is considered a major breakthrough (LeCun, Bengio, and Hinton 2015). The fundamental idea of a DNN is to use of several hidden layers. Subsequent layers process the output of previous layers to sequentially transform the initial representation of objects. The goal is to build a specific representation useful for some given task (i.e., classification). Multi-layered learning has dramatically improved the state of the art in many high-impact application domains, such as speech recognition, visual object recognition, and natural language processing.

Another useful attribute of deep neural nets is that they can learn a *hierarchy* of representations, associated to layers of the net. Indeed, a neural net with L layers can be formalized as a sequence of coders (c^1, \dots, c^L). The representation in the first layer is $y^1 = c^1(x)$, and for subsequent layers $1 < \ell \leq L$ it is $y^\ell = c^\ell(y^{\ell-1})$. The role of the output layer is then to map the top representation y^L onto a final target $\hat{y} = d(y^L)$, for example, in the case of classification, onto a finite set of object types. In what follows, we will denote the function that the full net implements by f . With this notation, $\hat{y} = d(y^L) = d(c^L(y^{L-1})) = \dots = f(x)$.

The formal training setup is the following. We are given a training set $\mathcal{D} = \{x_1, \dots, x_n\}$, a set of learning targets (e.g., object types) $\{y_1, \dots, y_n\}$, and a score function $s(y, \hat{y})$ representing the error (negative value) of the prediction \hat{y} with respect to the real target y . The setup is called *supervised* because both the targets of the network y_i and the value of its output s is given by the designer. We train the network f_w , where w is the vector of all the parameters of the net, by classical stochastic gradient descent (modulo technical details): we cycle through the training set, reconstruct $\hat{y}_i = f_w(x_i)$, compute the gradient $\delta_i = \partial s(y_i, \hat{y}_i) / \partial w$, and move the weights w by a small step in the direction of $-\delta_i$.

Autoassociative neural nets (autoencoders)

Formally, an autoencoder is a supervised neural network whose goal is to predict the input x itself. Such neural networks are composed of an encoder part and a decoder part. In a sense, an autoencoder learns to disassemble then

to reassemble the object x . Our approach is based on a particular the technique described in (Bengio et al. 2013). We first learn about the input space by training an *autoassociative* neural net (a.k.a. *autoencoder*) f using objects $\mathcal{D} = \{x_1, \dots, x_n\}$, then apply a technique that designs a generative function (simulator) g based on the trained net f .

Autoencoders are convenient because they are designed to learn a representation $y = c(x)$ of the object x and a decoder $x' = d(y)$ such that x is close to x' in some formal sense, and y is concise or simple. In the classical *information theoretical* paradigm, both criteria can be formalized: we want the code length of y (the number of bits needed to store y) to be small while keeping the distortion (e.g., the Euclidean distance) between x and x' also small. In (neural) *representation learning*, the goals are somewhat softer. The distortion measure is usually the same as in information theory, but simplicity of y is often formalized implicitly by using various *regularization* operators. The double goal of these operators is to prevent the algorithm to learn the identity function for the coder c , and to learn a y that uses elements (“code snippets”) that agree with our intuition of what object components are.

The decoder d takes the top representation y^L and reconstructs $x' = d(y^L)$. The goal is to minimize a score $s(x, x')$, also called *distortion*, that measures how close the input image x is to the reconstructed image x' . Throughout this paper, we will use the Euclidean squared distance in the pixel space $s(x, x') = \|x - x'\|_2^2$.

We are using a particular variant of autoencoders, called sparse convolutional autoencoders (Makhzani and Frey 2015) with $L = 3$ coding layers and a single decoding layer. Convolutional layers are neural net building blocks designed specifically for images: they are essentially small (e.g., 5×5) filters that are repeated on the full image (in other words, they share the same set of weights, representing the filter). The sparse regularizer penalizes dense activations, which results in a sparse representation: at any given layer, for any given image x , only a small number of units (“object parts”, elements of y^ℓ) are turned on. This results in an interesting structure: lower layer representations are composed of small edgelets (detected by Gabor-filter like coders), followed by small object parts “assembled” from the low-level features. The convolutional filters themselves are object parts that were extracted from the objects of the training set. The sparsity penalty and the relatively small number of filters force the net to extract features that are general across the population of training objects.

Generating from the learned model

In this section we present and comment some experimental results. First, we provide some illustrations providing an insight regarding the usefulness of the representations extracted by a deep net for searching for novelty. Then, we present the method we use to generate novel image objects, based on the formal approach described in the section *Learning to generate*.

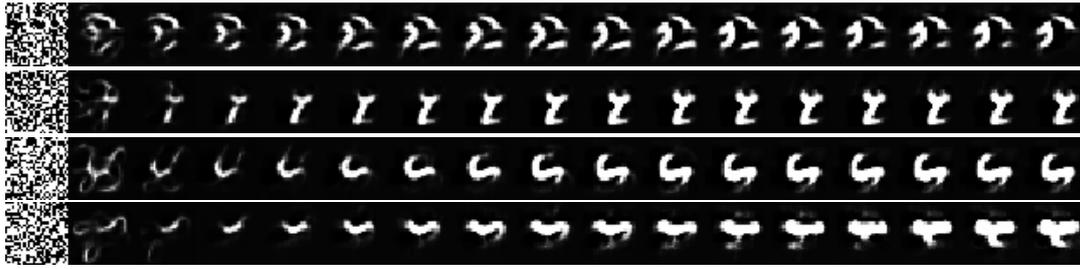


Figure 2: Four examples illustrating the iterative generative process. At each iteration, the net pushes the input image closer to what it can “understand” (reconstruct easily), converging to a fixed point (an image that can be reconstructed without an error).

Searching for new types: with and without knowledge

We argued in previous sections that combinatorial search over the objects has disadvantages over a search process driven by a knowledge over the same set of objects obtained by the system itself. When the learning is implemented through a deep neural net, this knowledge is encoded in the form of multiple levels of representations and transformations from layer to layer. To demonstrate the effect of knowledge over these search procedures, instead of searching in the original object space of x , we have applied simple perturbation operations on the representation space y .

Figure 3 illustrates the results of these perturbations. In the original representation space, crossover and mutation operators create noisy artifacts, and the population quickly becomes unrecognizable, which, unless the sought effect is precisely the noise, is not likely to produce novel objects (let alone types) unless a fitness function that drives the search is given (which is what we are trying to avoid). In comparison, the same operators applied to the code y produced by the deep nets produce less noisy and seemingly more coherent forms. In fact, some novel symbols that go beyond the known digits seem to have already emerged and can be consolidated by further iteration through the model. Overall, combinatorial search in the representation space provided by the deep net seems more likely to generate meaningful combinations in the absence of a given evaluation function, thus, making it more suitable for knowledge-driven creativity.

Method for generating new objects from a learned model

To generate new objects in a knowledge-driven fashion, we first train a generative autoencoder to extract features that are useful for constructing such objects. To train the autoencoder f , we use the MNIST (Lecun and Cortes 2012) data set (Figure 4) containing gray-scale hand-written digits. It contains 70 000 images of size 28×28 . Once the model learned to construct objects it has seen, it has also learned useful transformations that can be queried to generate new objects.

Autoassociative networks exist since the 80s (Rumelhart, Hinton, and Williams 1986; Baldi and Hornik 1989; Kramer 1991), nevertheless, it was discovered only recently that they can be used to generate new objects (Bengio et al. 2013; Kamyshanska and Memisevic 2013). The procedure is the

following. We start from a random image $x_0 = r$, and reconstruct it $x_1 = f(x)$ using the trained network f . Then we plug the reconstructed image back to the net and repeat $x_k = f(x_{k-1})$ until convergence. Figure 2 illustrates the process. At each step, the net is forced to generate an image which is easier to reconstruct than its input. The random seed r initializes the process. From the first iteration on, we can see familiar object parts and compositions rules, but the actual object is new. The net converges to a fixed point (an image that can be reconstructed without an error).

It can be observed that, although this kind of generative procedure generates new objects, the first generation of images obtained by random input (second column of Figure 2) look noisy. This can be interpreted as the model has created a novelty, but has not excelled yet at constructing it adequately. However, feeding this representation back to the model and generating a new *version* improves the quality. Repeating this step multiple times enables the model to converge effectively towards fixed points of the model, that are more precise (i.e., visually). Their novelty, in terms of typicality, can be checked using clustering methods and visualised as in Figure 5.

Generating new types

When the generative approach is repeated starting from multiple random images $\{r_1, \dots, r_n\}$, the network generates different objects $\{x_1, \dots, x_n\}$. When projecting these objects (with the original MNIST images) into a two-dimensional space using stochastic neighbor embedding (van der Maaten and Hinton 2008), the space is not filled uniformly: it has dense clusters, meaning that structurally similar objects tend to regroup; see Figure 5. We recover these clusters quantitatively using k-means clustering in the feature space $\{y_1, \dots, y_n\}$. Figure 6 contains excerpts from these clusters. They are composed of similar symbols that form a coherent set of objects, which can be perceived as new *types*.

Discussion and perspectives

It is possible to compare our work with several other published results. To start with, the generation of novelty through the use of neural nets is an old idea (Todd 1992; Todd 1989; Thaler 1998). There are two main differences between our approach and theirs. First, our emphasis is on studying how an artificial agent can generate novelty that

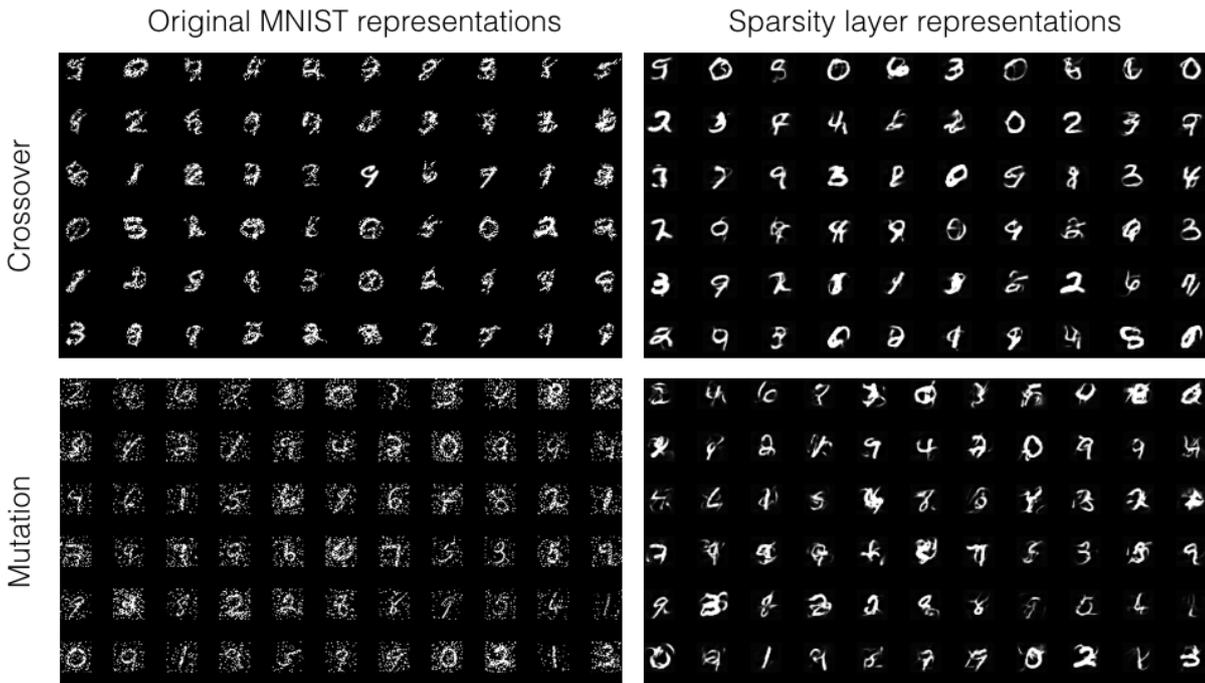


Figure 3: The effect of perturbations applied to object representations. On the left, the effect of crossover and mutation on the original representations of MNIST. On the right, the same operators applied to the representations learned by the deep generative net. Visually, this latter category seem less affected by perturbations, and thus is likely to provide a better search space for novelty.

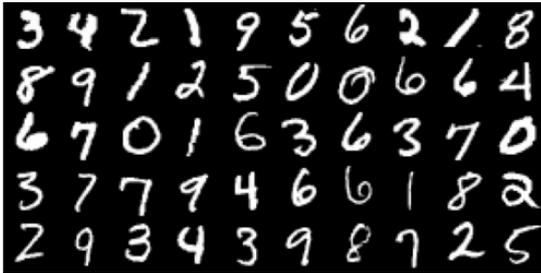


Figure 4: A subsample of MNIST, the data set we use to train the autoencoder f .

does not fit into learned categories, rather than creating objects with artistic value. This experimental setup is intended to provide means for studying how a creative agent can build an evaluation function for new types of objects. Second, we explicitly aim at establishing a bidirectional link between generative models for computational creativity and generative models within statistics and machine learning. Beyond the use of techniques and tools developed in these disciplines, we wish to raise research questions about creative reasoning that would also be interesting in statistics and machine learning.

In fact, some recent work has already started exploring the creative potential of deep neural networks. For instance, (Mordvintsev, Olah, and Tyka 2015) uses a deep net to project the input that would correspond to a maximal activation of a layer back onto an image in an iterative fashion.

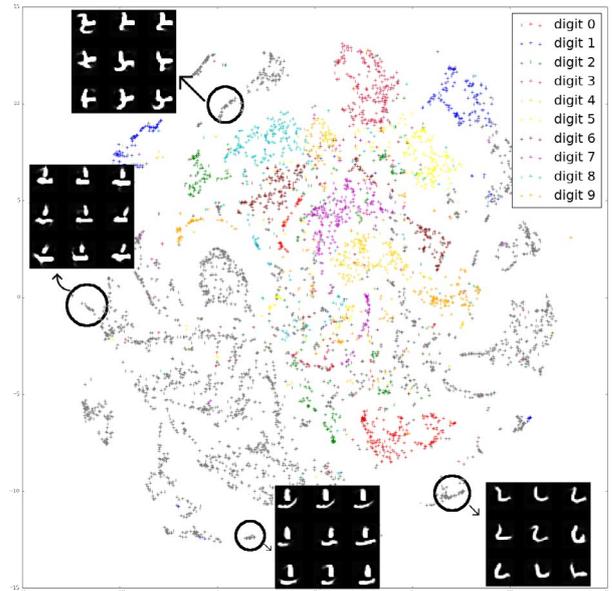


Figure 5: A distance-preserving projection of digits to a two-dimensional space. Colored clusters are original MNIST types (digit classes from 0 to 9). The gray dots are newly generated objects. Objects from four of the clusters are displayed.

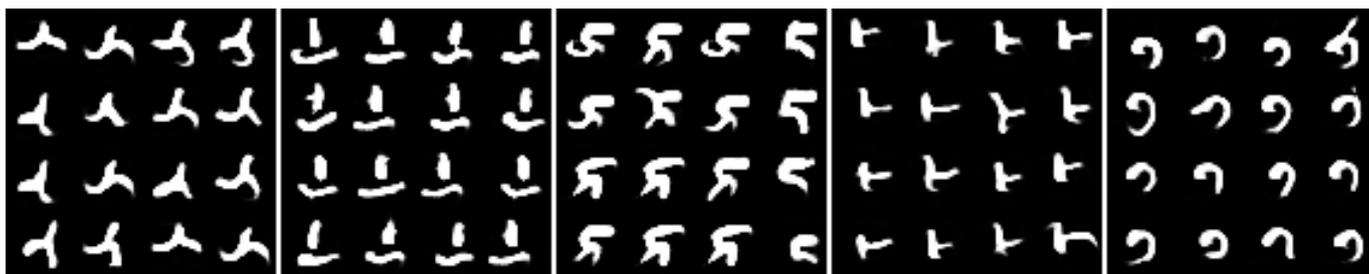


Figure 6: A sample of new types discovered by the model

The images are perceived as dreamy objects that are both visually confusing and appealing. Another work (Gatys, Ecker, and Bethge 2015) uses correlations of activations in multiple layers of a deep net to extract style information from one picture and to transpose it to another. Finally, (Nguyen, Yosinski, and Clune 2015) uses a trained net as a fitness function for an evolutionary approach (see also (Machado, Romero, and Manaris 2008) for a similar application with shallow nets). These successful approaches demonstrate the potential of deep nets as an instrument for creativity research and for generating effects that can be deemed as surprising, even creative. The present approach and the points the paper puts forward are significantly different. Compared to the architectures used in these studies, ours is the only one that uses a generative deep autoassociative net. The reason for this choice is twofold. First, we aim at using and understanding the generative capacity of deep nets. Second, we are interested in the deconstruction and reconstruction our architecture provides since our aim is to build objects through the net (not to create an effect that modifies existing objects). Once again, thinking about and experimenting with these foundational aspects of generative deep nets provide a medium through which notions of creativity research can be clarified through statistical notions. This is not among the declared objectives of previous works.

The novelty-seeking behavior of our system can also be compared to the recent novelty-driven search approaches in the evolutionary computing literature (Mouret and Doncieux 2012; Lehman and Stanley 2011). These approaches, like ours, seek to avoid objective functions and push the system to systematically generate novelty in terms of system behavior (e.g., novelty in the output). Our system is akin to such methods in spirit with one main difference: we believe that knowledge plays a fundamental role in creative endeavor and the decision of the system regarding the search for novelty should come from its own knowledge model. Note that this does not exclude a more general system where several systems such as ours can compete to differentiate themselves from the observed behavior of others, effectively creating a community of designers.

Our system provides a step towards an experimental study of how an artificial agent can drive its search based on knowledge. Furthermore, it can effectively create new types of objects preserving abstract and semantic properties of a domain. However, we have not fully addressed the question

of how such an agent can build its own value function about novelty. Nevertheless, the system enables numerous ways to experiment with various possibilities. An obvious next step would be to hook our system to an external environment, where the system can receive feedback about value (Clune and Lipson 2011; Secretan et al. 2008). To avoid the fitness function barrier, this should be done in such a way that the system can build its own value system rather than only learning the ones in its environment.

Summary

We provided an experimental setup based on a set of principles that we have described. The pinnacle of these principles is that artificial creativity can be driven by knowledge that a machine extracts itself from a set of objects defining a domain. Given such knowledge, a creative agent can explore new *types* of objects and build its own value function about novelty. This principle is in contrast with existing systems where the system designer or audience imposes a value function to the system, for example, by some fitness function.

We argued that when an artificial creative agent extracts its own domain knowledge in the form of features that are useful to reconstruct the objects of the domain, it becomes able to explore novelties beyond the scope of what it has seen by exploring systematically unknown types. We have demonstrated the idea by using a deep generative network trained on a set of digits. We proposed a compositional sampling approach that yielded a number of new types of digits.

While our setup provides a basis for further exploring how an agent can develop its own value function, it is also a bridge with the powerful theories and techniques developed within the statistics and machine learning communities. A colossal amount of work has already been published on deep neural networks with significant breakthroughs in many domains. Deep learning will be all the more valuable if it offers an evolution of the machine learning paradigm towards machine creativity.

Acknowledgments

We thank our anonymous referees for helpful comments. This work was partially supported by the HPC Center of Champagne-Ardenne ROMEO. This work has been funded by the P2IO LabEx (ANR-10-LABX-0038) in the framework Investissements d'Avenir (ANR-11-IDEX-0003-01) managed by the French National Research Agency (ANR).

References

- [Baldi and Hornik 1989] Baldi, P., and Hornik, K. 1989. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks* 2(1):53–58.
- [Beaucousin et al. 2011] Beaucousin, V.; Cassotti, M.; Simon, G.; Pineau, A.; Kostova, M.; Houdé, O.; and Poirel, N. 2011. Erp evidence of a meaningfulness impact on visual global/local processing: when meaning captures attention. *Neuropsychologia* 49(5):1258–1266.
- [Bengio et al. 2013] Bengio, Y.; Yao, L.; Alain, G.; and Vincent, P. 2013. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, 899–907.
- [Boden and Edmonds 2009] Boden, M. A., and Edmonds, E. A. 2009. What is generative art? *Digital Creativity* 20(1-2):21–46.
- [Breton 1932] Breton, A. 1932. Marcel Duchamp : The bride stripped bare by her own bachelors. *This Quarter Surrealist Number* 5(1).
- [Clune and Lipson 2011] Clune, J., and Lipson, H. 2011. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *Proceedings of the European Conference on Artificial Life*, 144–148.
- [Diedrich et al. 2015] Diedrich, J.; Benedek, M.; Jauk, E.; and Neubauer, A. C. 2015. Are creative ideas novel and useful? *Psychology of Aesthetics, Creativity, and the Arts* 9(1):35.
- [Edmonds 1969] Edmonds, E. 1969. Independence of rose’s axioms for m-valued implication. *The Journal of Symbolic Logic* 34(02):283–284.
- [Galanter 2012] Galanter, P. 2012. Generative art after computers, [in:] generative art—proceedings ga2012 xv generative art conference, red. soddu c.
- [Gatys, Ecker, and Bethge 2015] Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style.
- [Goodfellow et al. 2014] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2672–2680.
- [Hatchuel and Weil 2009] Hatchuel, A., and Weil, B. 2009. Ck design theory: an advanced formulation. *Research in engineering design* 19(4):181–192.
- [Hinton, Osindero, and Teh 2006] Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18(7):1527–1554.
- [Jennings 2010] Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.
- [Kamyshanska and Memisevic 2013] Kamyshanska, H., and Memisevic, R. 2013. On autoencoder scoring. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 720–728.
- [Kazakci et al. 2010] Kazakci, A.; Hatchuel, A.; Le Masson, P.; Weil, B.; et al. 2010. Simulation of design reasoning based on ck theory: a model and an example application. In *DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia*.
- [Kazakçı 2014] Kazakçı, A. 2014. Conceptive artificial intelligence: Insights from design theory. In *International Design Conference DESIGN2014*, 1–16.
- [Kramer 1991] Kramer, M. A. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AICHE journal* 37(2):233–243.
- [Lecun and Cortes 2012] Lecun, Y., and Cortes, C. 2012.
- [LeCun, Bengio, and Hinton 2015] LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436–444.
- [Lehman and Stanley 2011] Lehman, J., and Stanley, K. O. 2011. Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX*, 37–56. Springer.
- [Machado, Romero, and Manaris 2008] Machado, P.; Romero, J.; and Manaris, B. 2008. Experiments in computational aesthetics. In *The art of artificial evolution*. Springer. 381–415.
- [Makhzani and Frey 2015] Makhzani, A., and Frey, B. J. 2015. Winner-take-all autoencoders. In *Advances in Neural Information Processing Systems*, 2773–2781.
- [McCormack et al. 2014] McCormack, J.; Bown, O.; Dorin, A.; McCabe, J.; Monro, G.; and Whitelaw, M. 2014. Ten questions concerning generative computer art. *Leonardo* 47(2):135–141.
- [McCormack 2013] McCormack, J. 2013. *Aesthetics, art, evolution*. Springer.
- [Mordvintsev, Olah, and Tyka 2015] Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going deeper into neural networks. *Google Research Blog*. Retrieved June 20.
- [Mouret and Doncieux 2012] Mouret, J.-B., and Doncieux, S. 2012. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary computation* 20(1):91–133.
- [Nees 1969] Nees, G. 1969. *Generative Computergraphik*. Siemens AG.
- [Nguyen, Yosinski, and Clune 2015] Nguyen, A. M.; Yosinski, J.; and Clune, J. 2015. Innovation engines: Automated creativity and improved stochastic optimization via deep learning. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, 959–966. ACM.
- [Parikka 2008] Parikka, J. 2008. Leonardo book review: The art of artificial evolution: A handbook on evolutionary art and music.
- [Rumelhart, Hinton, and Williams 1986] Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *NATURE* 323:9.
- [Runco and Jaeger 2012] Runco, M. A., and Jaeger, G. J. 2012. The standard definition of creativity. *Creativity Research Journal* 24(1):92–96.
- [Salakhutdinov and Hinton 2009] Salakhutdinov, R., and Hinton, G. E. 2009. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, 448–455.
- [Secretan et al. 2008] Secretan, J.; Beato, N.; D Ambrosio, D. B.; Rodriguez, A.; Campbell, A.; and Stanley, K. O. 2008. Picbreeder: evolving pictures collaboratively online. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1759–1768. ACM.
- [Sims 1991] Sims, K. 1991. *Artificial evolution for computer graphics*, volume 25. ACM.
- [Takagi 2001] Takagi, H. 2001. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE* 89(9):1275–1296.
- [Thaler 1998] Thaler, S. L. 1998. The emerging intelligence and its critical look at us. *Journal of Near-Death Studies* 17(1):21–29.
- [Theis, Oord, and Bethge 2015] Theis, L.; Oord, A. v. d.; and Bethge, M. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.
- [Todd and Latham 1991] Todd, S., and Latham, W. 1991. *Mutator: a subjective human interface for evolution of computer sculptures*. IBM United Kingdom Scientific Centre.
- [Todd 1989] Todd, P. M. 1989. A connectionist approach to algorithmic composition. *Computer Music Journal* 13(4):27–43.
- [Todd 1992] Todd, P. M. 1992. A connectionist system for exploring melody space. In *Proceedings of the International Computer Music Conference*, 65–65. International Computer Association.
- [van der Maaten and Hinton 2008] van der Maaten, L., and Hinton, G. 2008. Visualizing data using t-SNE. *The Journal of Machine Learning Research* 9(2579-2605):85.
- [Von Hippel 1986] Von Hippel, E. 1986. Lead users: a source of novel product concepts. *Management science* 32(7):791–805.
- [Wiggins 2006] Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

NARRATIVES



Murder Mystery Generation from Open Data

Gabriella A. B. Barros¹, Antonios Liapis² and Julian Togelius¹

1: Tandon School of Engineering, New York University, New York, USA

2: Institute of Digital Games, University of Malta, Msida, Malta

gabriella.barros@nyu.edu, antonios.liapis@um.edu.mt, julian@togelius.com

Abstract

This paper describes a system for generating murder mysteries for adventure games, using associations between real-world people mined from Wikipedia articles. A game is seeded with a real-world person, and the game discovers suitable suspects for the murder of a game character instantiated from that person. Moreover, the game discovers characteristics of the suspects which can act as clues for the player to narrow down her search for the killer. The possible suspects and their characteristics are collected from Wikipedia articles and their linked data, while the best combination of suspects and characteristics for a murder mystery is found via evolutionary search. The paper includes an example murder mystery generated by the system revolving around the (hypothetical) death of a contemporary celebrity.

Introduction

Computational creativity focuses on discovering artifacts in creative domains such as art, music and digital games, or even mathematics and engineering. Sometimes, the results of creative processes are created *in vitro*, without a basis in the real world. This is possible, for instance, in automated theorem discovery (Colton 2002) where a model of finite algebra suffices for proving a generated theorem. However, the real world often influences the creative process in some way, acting as a training set (Eigenfeldt and Pasquier 2012), as a seed (Krzeczkowska et al. 2010; Hoover et al. 2012), as an evaluation (Correia et al. 2013; Martins et al. 2015) or as a mapping between dissimilar modalities (Johnson and Ventura 2014; Veale 2014). There is interesting research in transforming textual data into images (Krzeczkowska et al. 2010), poems (Colton, Jacob, and Veale 2012) or even games (Cook and Colton 2014).

This paper describes a system for creating digital adventure games using open data (primarily Wikipedia articles). Starting from a designer- or player-specified real-world person, the game produces a murder mystery where said person has been killed and the player must find the killer among several suspects. The player can pinpoint the killer by eliminating suspects based on certain characteristics they do not share with the killer. Suspects and their characteristics are collected from Wikipedia articles; the suspects are associated with the victim, while the characteristics may be shared

among some but not all of the suspects. From this broad range of suspects and types of characteristics (e.g. date of birth, affiliation, awards), the best combination for use in a murder mystery is discovered via evolutionary search, which ensures that the mystery is solvable while the characteristics are diverse within the chosen set of suspects.

The WikiMystery system described here is the next step from our previous work on “Data Adventures”, i.e. generating adventure games from open data (Barros, Liapis, and Togelius 2015; 2016). The system presented hereby enriches the experience by adding multiple paths between a victim and possible culprits (thus allowing the player to explore the game in a non-linear fashion). Moreover, the addition of clues (which help the player eliminate suspects) and puzzles (which block progression on specific plot lines) increases the richness of interaction. More importantly, WikiMystery is unique among attempts to generate games from data in that the storyline and affordances (possibly even the difficulty in terms of solvability) are affected directly by the data.

Related Work

As a highly creative domain, storytelling has received considerable attention in computational creativity research. Such research has often focused on generating complete stories in textual form: examples include stories written around a specific theme such as betrayal in BRUTUS (Bringsjord and Ferrucci 2000), a specific caste of heroes such as the Knights of the Round Table in MINSTREL (Turner 1993), or whether the story achieves an intended goal state (Riedl and Young 2010). On the other hand, games as interactive media (Aarseth 1997) can do away with several of the requirements of full story generation and the challenges posed e.g. by the natural language processing needed for narration (Montfort and Pérez y Pérez 2008). Examples of a narrative structure transformed (computationally) to and from interactive experiences include the work of Laclaustra et al. (2014) which uses a game map and game characters to create a story from their interactions, and the work of Robertson and Young (2015) which transforms a story plan into a level structure, instantiates game characters and creates actions for the player to issue to their avatar.

Automatically transforming stories into games and vice versa is an example of computational creativity used for *transformation* or *reinterpretation* of data from one medium

to another. Other examples include the transformation of images (Johnson and Ventura 2014), game levels (Lopes, Liapis, and Yannakakis 2015), or text (Thorogood and Pasquier 2012) into soundscapes, news articles into games (Cook and Colton 2014) or collages (Krzeczkowska et al. 2010), etc. For the purpose of conciseness, we will focus on how data has been transformed into playable experiences.

The automated game creator Angelina is a prime example of a system transforming data into games: Angelina (Cook, Colton, and Pease 2012) generates platformer levels and decorates them in an audiovisual theme. The theme is derived from the text body of news articles, while the sounds, images, backgrounds are derived from parsing the text and searching online databases using these keywords. Game-o-matic (Treanor et al. 2012) uses human-authored concepts and their associations to generate consistent games and their rules, while relying on web-based images of the authored concepts for their in-game visual representation. Open data, on the other hand, have also been transformed into playable experiences: examples include the simple physics game BarChartBall (Togelius and Friberger 2013) which uses UK census information to form the terrain of a ball-rolling game, or OpenTrumps (Cardona et al. 2014) which instantiates *Top Trumps* (Winning Moves 1999) card decks with countries' statistics (from United Nations and World Development Indicator databases). Games that feature content generated from open data are often referred to as *Data Games* (Friberger et al. 2013).

Similarly to the aforementioned projects, WikiMystery and its overarching project of "Data Adventures" (Barros, Liapis, and Togelius 2015) use online data to produce playable games featuring real-world people (both alive and dead) and locations. Similar to Angelina and Game-O-Matic, they decorate the game's visuals with images collected from Wikimedia Commons. Unlike some other data games, however, the data form a core part of the gameplay and deeply affect the experience. Compared to Game-O-Matic or Angelina where the online data is used to theme an already playable game, WikiMystery relies on the data to form the plot of the game, the locations the player can visit, the non-player characters (NPCs) and their relationships (which act as clues in the mystery). This reliance on data (which, due to the crowd-sourced nature of Wikipedia and human fallibility can be erratic or incomplete) poses a grand challenge to Data Adventures. To guarantee playable adventure games relying exclusively on such data requires an almost human-like intelligence and creativity. The current steps taken by the authors with this paper and its predecessors (Barros, Liapis, and Togelius 2015; 2016) can — and often do — create absurd storylines and unintuitive associations between non-player characters. It should be noted, however, that absurdity is inherent to the grounding of data (and a desirable artifact of freedom of online speech) and hiding or curating it would remove the core strength (and evidence) of the open data-driven nature of the game. The world is absurd, and this is reflected in some of the results of a data-based game generator.

Overview of WikiMystery

WikiMystery draws inspiration from several adventure games, a broad and rich genre that has gained a resurgence of popularity over the past years. Due to the diversity of games within this genre, there is a large variety of mechanics and gameplay styles in adventure games that can also be found in other genres. In this work, we refer to adventure games as games with the following characteristics: they are story-driven, their core mechanics revolve around puzzle-solving, interaction with the game world is mostly done through object manipulation, the player controls a character in the world and is motivated to explore the interactions that the space around her provides (Fernández-Vara and Osterweil 2010).

The plot of the game revolves around a crime: someone was killed and it is up to the player, a detective, to find out who did it by gathering clues to arrest the culprit. The plot is constructed from Wikipedia articles and their links. The player begins in the house of the deceased, knowing who died and finding a list of possible suspects. With this list, the detective can travel between locations, talk to NPCs and interact with items. Her goal is to pinpoint who, within that list, is the culprit and prove so by selecting the right options in an arrest warrant template. This template has a series of possible characteristics, such as "residence", "nationality" or "awards received", and each characteristic type has a series of possible values. By selecting the right combination of values, the player can differentiate between suspects.

Initially, the user inputs a person's name to the system, ("Justin Bieber" in this paper's example) who will be, for plot purposes, killed. The system queries DBpedia (Auer et al. 2007), a structured version of Wikipedia, to find possible suspects: people linked to the victim. A genetic algorithm evaluates possible suspects and the relations between them, to optimize the suspect pool and guarantee playability. In other words, to guarantee that it is possible to pinpoint the culprit among the suspects, by assigning characteristics such as "genre: Trip_hop" or "homeTown: Stratford".

Once suspects are selected, the system finds a path between the victim and each suspect. These paths are used to populate the system with game objects: cities, buildings, NPCs, items and dialogue. The process of selecting suspects and paths is exemplified in Figures 1 and 2. Finally, for each object in the game, images are obtained from Wikimedia Commons using Spritely (Cook and Colton 2014).

Choosing Suspects and their Characteristics

In a crime-based story, such as the Sherlock Holmes series (BBC 1965) or the game *"Where in the world is Carmen Sandiego?"* (Brøderbund 1985), the hero/player is often asked to identify the culprit of a crime by solving puzzles and finding clues. In a game where characters and puzzles are created from open data, the challenge of selecting data suitable for suspect and clue creation arise. In a design sense, we want suspects that are related to the victim so that we can establish a motive. Additionally, we want the game to be solved by eliminating suspects when the player discovers that the killer does not have characteristics possessed

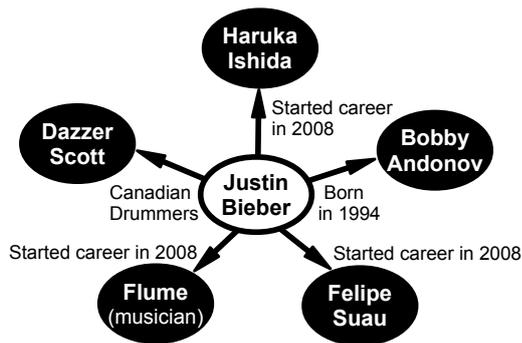


Figure 1: Simple representation of the process of selecting suspects and finding relations between the victim and the suspects. Initially, the system only has a single node: the victim (white node). Suspects related to the victim are selected with a genetic algorithm (black nodes), and paths between the victim and the suspects are created from DBpedia (see Fig. 2). All suspects share common characteristics with the victim, shown on the arrows: in this example three of the suspects started their careers on the same year as the victim.

by other suspects. The killer must therefore have enough unique characteristics not shared by at least one other suspect. In the current design paradigm, each clue should suffice to eliminate one innocent suspect: thus for X suspects (among which one is the culprit), $X - 1$ clues (and therefore types of characteristics) are needed.

Creating a pool of possible suspects: It is necessary to create a pool of possible suspects for the genetic algorithm to choose from throughout evolution. To do this, we use DBpedia, which structures information from Wikipedia in the form of tuples¹. All suspects must share a direct link to the victim based on DBpedia: i.e. suspects must have one or more common characteristic types (predicates) and characteristic value (object) with the victim. For instance, both the victim and a suspect may have lived in the same town, or have the same age (see Fig. 1 for sample shared characteristics). Given these suspects, we generate a list of all possible characteristic types (e.g. “homeTown”, “associatedBand”, etc.) that one or more suspects have. We omit the value “Living.People”, which simply indicates that the person is alive, as well as website-related types (e.g. thumbnails, links for redirected pages, etc) in order to avoid highly abstract relationships as well as a vast, unmanageable search space for the genetic algorithm.

Evolving combinations of suspects: Once we collect all possible suspects, we need to select a number of X suspects among those and $X - 1$ types of characteristics (e.g. “place-Of-Birth” or “yearsActive”), used to pinpoint the culprit. The suspects’ types of characteristics must be varied but common among most suspects; each suspect however must have

¹An article in Wikipedia may be represented in DBpedia by a collection of $\langle \text{Subject}, \text{Predicate}, \text{Object} \rangle$ tuples (e.g. $\langle \text{Nikola Tesla}, \text{Birth Date}, 1856-07-10 \rangle$).

a unique combination of characteristic values so that we can identify the culprit. A genetic algorithm (GA) is used to find this combination of suspects and types of characteristics. Mutation alone is used as a genetic operator, and in our experiments evolution runs for 100 generations on a population of 100 individuals. The genotype is a vector of size $2X - 1$ (X suspects and $X - 1$ characteristics). The initial population is generated by selecting random suspects, without repetition, and selecting characteristics within the sub-pool of these suspects’ characteristics. The mutation operator changes a few of these elements (i.e. suspects and characteristics). Since we do not have a crossover operation, mutation is mandatory: it will always change at least a few elements. Additionally, if the element mutated is a person, then all types of characteristics undergo a validation check: any characteristic type not possessed by at least one suspect is replaced by a new one.

Fitness function: Our fitness function takes two major concerns into account: diversity and solvability. Diversity measures the distribution of characteristics’ types and values for all suspects, and favors characteristics for which values differ more between suspects. For example, a set of suspects living in the same city and owning the same kind of car is less diverse than a set where most suspects live in different cities and drive different cars (but still have the “city” and “car” characteristics). The diversity fitness (f_D) is calculated as the total entropy of each type, multiplied by the number of suspects that have that characteristic type:

$$f_D = \sum_{i=0}^P \left(Q_i \times \left(- \sum_{j=0}^{V_i} p_{ij} (\log_2 p_{ij}) \right) \right) \quad (1)$$

where P is the number of characteristic types, V_i is the number of values for type i , and Q_i is the number of people that have type i . We multiply Q_i to encourage that more suspects have that type. p_{ij} is calculated as the number of people that have a certain value j in characteristic i , divided by Q_i .

Solvability, on the other hand, guarantees that it is possible to pinpoint the killer among the suspects. As discussed above, WikiMystery operates under the assumption that each discovered clue should eliminate one suspect: the clue in this case specifies which value of a characteristic does **not** belong to the culprit, but belongs to an innocent individual instead. To do so, we need to pair the suspects and types uniquely, such that each suspect can be identified from the killer. This fitness is calculated via a form of Depth-First Search (DFS): for each person in a genome, we choose a potential culprit and let the remaining people be suspects. For each one of the $X - 1$ characteristic types in the gene, add that type to a ‘clue’ list, if and only if: 1) the killer has a value for that type of characteristic; 2) at least one of the suspects has that type of characteristic; 3) the value of the killer for that characteristic is different than that of the suspect. This is done to avoid characteristic types that can, single-handedly, allow the player to pinpoint the culprit.

Having chosen a potential culprit, the algorithm creates a ‘clue’ list for each innocent suspect (s). For each characteristic type in the pool, if s has a different value from that of

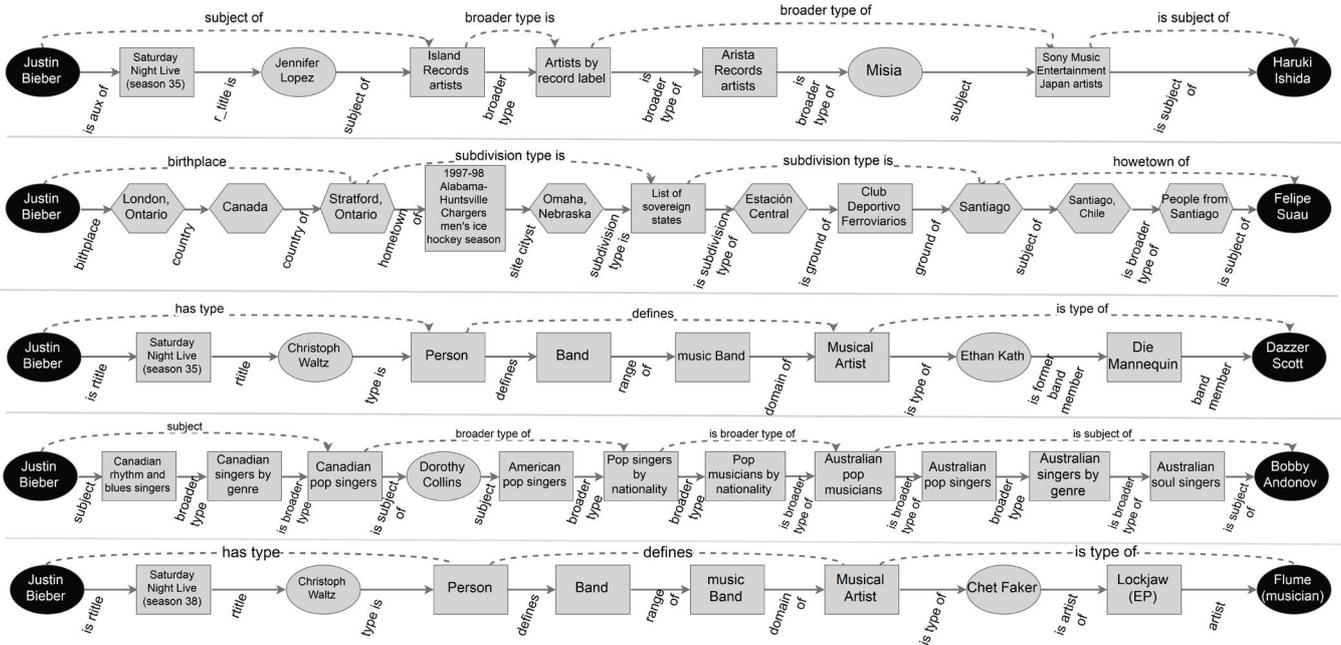


Figure 2: All major and minor paths between the victim and each suspect. Major paths are denoted with dotted arrows, minor paths have black arrows. Locations are represented as hexagons, NPCs as circles and items (books) as squares.

the culprit, then the characteristic is added to the list. The algorithm also inserts an empty symbol in each list, allowing a suspect to remain without any clue. At this point, the algorithm recursively pairs suspects and characteristic types (or empty values), so that each type cannot be used more than once (i.e. if a suspect is paired to a type, this type cannot be assigned to any other suspect). The algorithm backtracks if it does not find the optimal solution, similarly to a DFS, and stops only if it finds the maximum possible fitness or exhausts the search space. If the algorithm did not find the maximum fitness, it will select the next suspect as the potential culprit and restart the process, until it iterates through all X suspects or finds the optimal solution. The fitness is the number of successfully paired types, and the maximum fitness is $X - 1$ (when each type is paired to one suspect).

The GA uses cascading elitism (Togelius, De Nardi, and Lucas 2007) to ensure both fitnesses are optimized: first, the population is sorted using the solvability fitness, and half of the population is eliminated. The remaining population is sorted by diversity, and half of those individuals are removed. The remaining $\frac{1}{4}$ of the original population is cloned and mutated until the population reaches its original size.

Finding relations between Wikipedia articles

An adventure game can be viewed as a series of linked, concurrently or sequentially, challenges and events. As such, they may be represented as a directed graph, starting at the game's initial state/event, and leading to possible endings. If we identify the victim as the initial node in our graph, and each suspect as a potential ending, we can generate a path between the victim and each suspect, which would essen-

tially amount to the game itself. To generate this graph, our system calculates paths from the victim to each suspect using DBpedia. In this context, a path represents the relations between a sequence of Wikipedia articles, and consists of nodes (articles) and edges (the links between them).

The system queries DBpedia multiple times for possible paths between two individuals. At first, we select a "major path", a longer path between any given two articles that models the general relations between these two. Secondly, for each pair of articles in the major path, we select another "minor path", a shorter, more refined path that expands on the general idea. It provides a longer gameplay and a different, more indirect relation between the victim and the suspect. Figure 2 shows all five paths obtained for a set of suspects: major paths are represented with dotted arrows, and minor paths with black arrows. Paths are evaluated with a weighted sum based on their uniqueness and length. To evaluate uniqueness, we calculate the entropy of the nodes and of the edges of a path in relation to all nodes/paths found in that query: the more uncommon the nodes/edges, the better the path is evaluated. More details on the generation of major and minor paths are provided by Barros, Liapis, and Togelius (2016).

Data transformation

The previous step has secured a set of suspects and a set of characteristic types. This section describes how these DBpedia entries are transformed into playable, interactable game objects, and describes how these objects are placed in the gameworld based on their relationship with the victim. Game objects are divided into three groups: locations, items

```

Initialize empty stack elements;
for i ← 1 to depth-1 do
  if elements not empty then
    Choose an empty node n at depth i;
    Pop top of elements stack and assign to n;
  else if rolled the chance for adding a puzzle then
    Create a random SOLUTION/BLOCK pair;
    Choose an empty node n at depth i;
    Assign SOLUTION to n;
    Push BLOCK to elements stack;
while elements not empty do
  Choose an empty node n at depth depth;
  Pop top of elements stack and assign to n;

```

Algorithm 1: Pseudocode for puzzle placement, for a set of paths of maximum depth *depth*. Empty nodes are nodes with no blocks or solutions assigned to them.

and NPCs. To create them, we use the following guidelines:

- Any article tagged with type “Person” (real or fictional) is instantiated as an NPC. An image of the person is obtained from Wikimedia Commons². If no image is found, a random image of a person of the same sex is used.
- Any article that contains a geographic coordinate and is tagged with type “Place” in DBpedia is transformed into a city or a state. Information about this location is added to the game object created. Locations can be accessed in-game through the world-map. A map of this place is obtained through JMapView³ and OpenStreetMaps (Haklay and Weber 2008).
- If an article has a geographic position, but is not tagged as a “Place” in the DBpedia ontology, it becomes a building: a location within a city/state where the player can interact with NPCs or items. It can be accessed in-game by clicking on the building icon while the player is viewing the map of a city/state.
- If the article does not follow any of the rules above, a game item is created with information on this article. At this point, the game items include only books.

Once all game objects within a path are created, we have something resembling a tree, with the victim as the root, and the suspects at the leaves. However, it is still necessary to add a set of clues and conditions between them, so that each object appears in the path in the correct order. Clues for items include text (e.g. if the next node in the path is a location, the book item will have some text indicating that this location is interesting), and for NPCs they will include dialogue sub-trees. For locations and buildings, however, it is necessary to create either a random NPC or a random item, and set the clue as above. Once all the clues are ready, the algorithm creates conditions between each object, so the player cannot interact with a game object or visit a location before they have seen the previous clue in that path.

²Wikimedia Commons: <http://commons.wikimedia.org/>

³JMapView: <http://wiki.openstreetmap.org/wiki/JMapView>

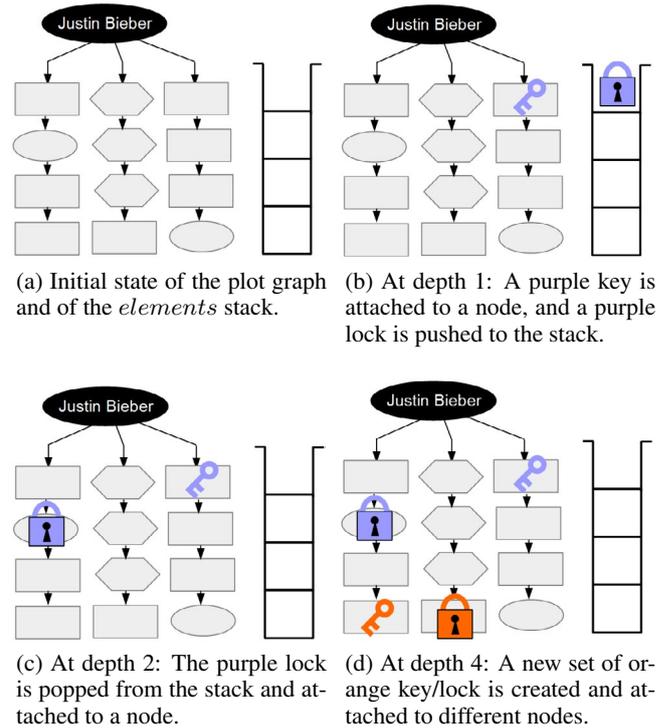


Figure 3: Example of position of pairs of solutions (keys) and blocks (locks). In Fig. 3d both the key and the lock are placed at the same depth (as it is the maximum depth).

Finally, additional puzzle elements (keys for locks, flashlights for dark rooms, tickets for private events) are included in the path. The path can be represented as a tree, therefore it is possible to group nodes by their depth, and add such puzzle elements while traversing the tree from the root to all leaf nodes. By always generating pairs of elements in order (SOLUTION: e.g. keys, tickets; BLOCK: locks, private event), we guarantee that the node will be reachable by placing the solution (on any path) at a lower depth than the block. For this, our algorithm works as shown in Algorithm 1. Figure 3a shows the initial state of a tree with three branches and maximum depth 4. At depth 1, the algorithm adds a solution (key) to a random plot node and a block object (lock) to the stack, as seen in Figure 3b. At the next depth, the lock is popped from the stack and added to a random node. At the maximum depth the chance for adding a new puzzle is rolled, so the key is placed in a random node at depth 4 and consecutively, since we are at the maximum depth, the lock is placed at the same depth in a random empty node.

Results

While our system can, theoretically, create mysteries for any given victim, our tests focused on a single subject. The reasoning behind this mainly involved query limitations imposed by DBpedia. To test our suspect and clue selection algorithm, it was necessary to run an enormous amount of queries, which could quickly become a problem for a non-

	Average (sd)
Number of cities	19.33 (5.07)
Number of buildings	50.60 (3.04)
Number of NPCs	31.00 (8.57)
Number of items	29.60 (5.21)
Ratio of real NPCs (over all NPCs)	82% (6%)
Average path length per suspect	11.93 (1.03)

Table 1: Average and standard deviation (in parentheses) of game objects generated and average length of paths selected by the crawler for each suspect. Results are collected from 50 independent runs of the generator with Justin Bieber as a hypothetical victim.

dedicated server. To avoid this, we limited our search to Justin Bieber (according to Wikipedia, “a Canadian singer and songwriter”) as the hypothetical victim of the murder mystery; as a contemporary celebrity figure, it was expected that he would have many connections in Wikipedia. The system queried DBpedia for every possible article about a person that had some non-trivial characteristic type and value in common with Justin Bieber. As described above, characteristics that were too broad were excluded (e.g. “category: Living people”, which includes every person currently alive, and hardly an indicative motive for a crime).

In order to assess, quantitatively, the types of games generated by WikiMystery, the full generative process was executed 50 times, with Justin Bieber as the hypothetical victim and five suspects required ($X = 5$). Table 1 shows the average and standard deviation of the number of game objects created, as well as the average path length per suspect. All of these metrics are indicative of the game’s playtime: there are apparently many cities and buildings for the player to visit; however, most locations contain items (books) which provide information and far fewer contain NPCs which the player can talk to. This indicates that discovered paths rely more on categories (e.g. Saturday Night Live, see Fig. 2) rather than people. It should be noted, however, that most NPCs represent real-world people (since less than a fifth of the NPCs are random, and therefore less interesting).

Example Game

An indicative run is included in Fig. 1 which shows the real people chosen as suspects, Fig. 2 which shows the paths between victim and suspects, and Table 2 which describes the characteristic types and values of the suspects. Table 2 shows, in bold, the clue that will be used to prove the innocence of each of the suspects. For instance, Haruka Ishida is eliminated as a possible culprit when the player discovers a clue that the culprit was **not** born in 1993: since Haruka Ishida is born in 1993, he can not be the culprit. Note that some values are shared between more than one suspects (e.g. both the culprit and Flume were born in 1991) and thus can not be used to rule out that particular innocent suspect.

When the game described in the above Figures and Tables is played, the player starts at the city “London, Ontario”. In the city map, she can find the house of a dead fictional Justin Bieber and come upon objects that indicate the existence of

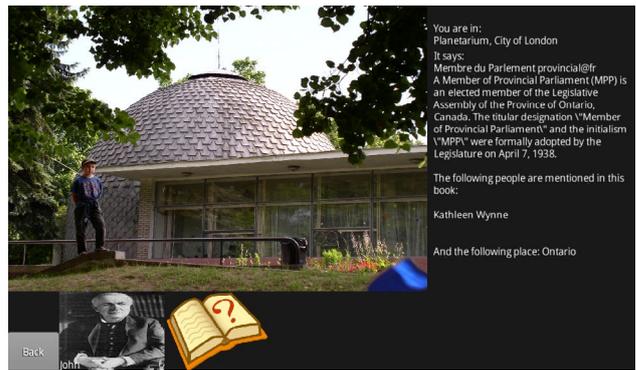


Figure 4: Screenshot of a ‘planetarium’ building location with a random NPC and a book containing clues.

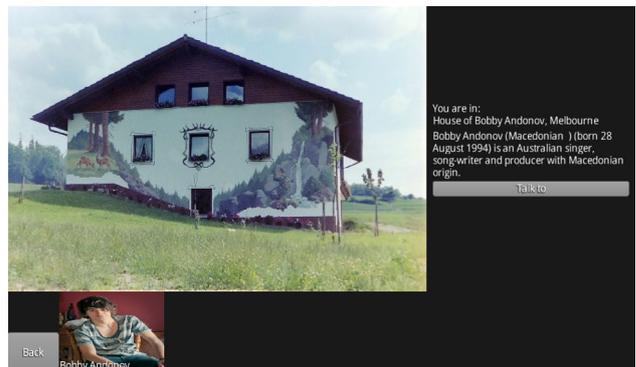


Figure 5: Screenshot of the ‘house of Bobby Andonov’ location with the NPC Bobby Andonov.

the suspects and the next clue in each path: e.g. for Haruki Ishida, a book on “Saturday Night Live (season 35)”. The player can then travel to places (including Omaha and Santiago), talk to various NPCs (some based on real people, such as Jennifer Lopez, others randomly generated) and search for clues (pieces of text in books, letters, etc). At one point, the player stumbles upon a locked house in Toronto, where a fictional Ethan Kath resides. To enter this house, she needs to go to a theatre in Concepción, Chile, where she will find a key. Eventually, the player can find the location of each suspect and, based on the clues gathered along her journey, identify the culprit: (fictional) Felipe Suau. Figure 4 shows a screenshot of a moment in the game, where the player is in a planetarium in the city of London, and can talk to a random NPC named John, or inspect a book she finds nearby. The book is titled “Membre du Parlement provincial” (author’s note: Member of Provincial Parliament) and mentions the NPC of Kathleen Wynne and the place of Ontario (which can now be interacted with and visited respectively). Figure 5 shows a building quoted as being the ‘House of Bobby Andonov’ (the background image is one of a random house), where the suspect NPC instantiated from Bobby Andonov can be found (his image from the Wikipedia article is used, shown at the bottom of the screen).

Innocent Suspects	Characteristics			
	Genre	Occupation	Birth year	Background
Flume (musician)	Electronica	Producer	1991	Non vocal instrumentalist
Haruka Ishida	J-pop	Singing	1993	Solo singer
Bobby Andonov	Pop music	Singing	1994	Solo singer
Dazzer Scott	Electronica	Singing	0025	Solo singer
Culprit				
Felipe Suau	Electronic music	PUNCHI PUNCHI Director	1991	Non vocal instrumentalist

Table 2: Solution found by the solver for five individuals. Each suspect is paired to one attribute (marked in bold) that will differentiate him from the killer. Note that Dazzer Scott’s birth year is not a typo; 0025 is the value returned by DBpedia.

Discussion

The main purpose of the overarching Data Adventures project is to create a generator capable of producing complete adventure games from open data, such as Wikipedia and OpenStreetMaps. The version of the system presented in this paper is able to generate complete adventures with multiple paths. It differs from other approaches which use data to create gameplay because, in this case, the data influences the gameplay directly: if we select a different starting point (i.e. the victim), we obtain a different story, characters, dialogues and general playable paths. Examples of different playthroughs when the starting point is different have been showcased by Barros, Liapis, and Togelius (2016), for a simpler adventure game generator using the same techniques.

The system is capable of evolving an interesting set of suspects, one of which is the killer. It can also successfully map suspects to predicates in such a way that the culprit can be pinpointed among innocents, thus guaranteeing that the game can be won. An interesting note is that the algorithm seems able to cluster suspects within certain domains (e.g. choosing only artists, or only sport-related people), which emerge from the fitness function’s attempts at maximizing characteristic types shared among all suspects (musicians will have the same characteristic types such as “associated-Band” or “producer”).

At the moment, our system still has limitations. The dialogues, items and puzzles are created from a limited set of templates, which we intend to expand in the future. The latest results also show flaws in the calculation of uniqueness while searching for paths. Our evaluation is based on comparing nodes and edges, which can be represented with words (e.g. “residence”, “New_York” or “20th_Century_Mathematicians”). Our current method compares two words with a hard comparison, i.e. if they have one character different, they are different nodes/edges. This implies that nodes like “Pop singers by nationality” and “Pop musicians by nationality” are considered as different as “Canada” and “Misia”. Further work should improve this by using a word comparison algorithm, such as Jaro-Winkler (Winkler 1999). We also evaluate the paths separately, not accounting for similarities between paths (Fig. 2 includes paths with many similar or identical nodes among them such as “Band” or “Saturday Night Live”). Measuring how different the paths are to one another is an important step to avoid repetitiveness. Another issue with the paths is the occurrence of very general categories, such as “person”

and “musical artist” (see Fig. 2). We are working on heuristics for excluding such categories.

Finally, an important limitation of the current system is that data collection for generating games when seeded with an individual (e.g. Justin Bieber) takes considerable time. This is due to the very large number of database requests needed and the nature of network communications. A priority for further development is therefore to create a version working from a local database.

Conclusion

This paper described a system for generating murder mysteries from open data, which can be used as the basis for generating adventure games. The methods presented here primarily revolve around the collection of appropriate data from Wikipedia articles, and the selection of the best suspects and their characteristics based on criteria of diversity (via data uniqueness) and value (via solvability) for a playable adventure game. Moreover, the paper outlined the necessary first steps for a fully generated adventure game which features the exploration of open data and the transformation of real-world frames and associations into playable experiences.

Acknowledgments

The NPCs discussed in the generated adventures are instantiated from real people, but it should be obvious that the similarities end there. The actions of NPCs in the adventure (as victims or culprits) in no way reflect the real-world people they are based on. The output of the generator in no way accuses or misrepresents these real-world individuals. WikiMystery creates fictional counterparts of public figures who have a presence in Wikipedia: any similarity between the (fictional) NPCs in the game and real-world people is therefore due to the data available in these open, freely accessible, online repositories.

We thank Ahmed Khalifa and Scott Lee for all fruitful and helpful discussions. Gabriella Barros acknowledges financial support from CAPES and Science Without Borders program, BEX 1372713-3.

References

Aarseth, E. 1997. *Cybertext: Perspectives on Ergodic Literature*. Johns Hopkins University Press.

- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. *Dbpedia: A nucleus for a web of open data*. Springer.
- Barros, G. A. B.; Liapis, A.; and Togelius, J. 2015. Data adventures. In *Proceedings of the FDG workshop on Procedural Content Generation in Games*.
- Barros, G. A. B.; Liapis, A.; and Togelius, J. 2016. Playing with data: Procedural generation of adventures from open data. In *Proceedings of the International Joint Conference of DiGRA and FDG*.
- Bringsjord, S., and Ferrucci, D. A. 2000. Inside the mind of brutus, a storytelling machine. In *Artificial Intelligence and Literary Creativity*. Lawrence Erlbaum Associates.
- Cardona, A. B.; Hansen, A. W.; Togelius, J.; and Friberger, M. G. 2014. Open trumps, a data game. In *Proceedings of the International Conference on the Foundations of Digital Games*.
- Colton, S.; Jacob, G.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the International Conference on Computational Creativity*.
- Colton, S. 2002. Automated theorem discovery: A future direction for automated reasoning. In *Proceedings of the IJCAR Workshop on Future Directions for Automated Reasoning*.
- Cook, M., and Colton, S. 2014. A rogue dream: Automatically generating meaningful content for games. In *Proceedings of the AIIDE workshop on Experimental AI & Games*.
- Cook, M.; Colton, S.; and Pease, A. 2012. Aesthetic considerations for automated platformer design. In *Proceedings of the Artificial Intelligence for Interactive Digital Entertainment Conference*.
- Correia, J.; Machado, P.; Romero, J.; and Carballal, A. 2013. Evolving figurative images using expression-based evolutionary art. In *Proceedings of the International Conference on Computational Creativity*.
- Eigenfeldt, A., and Pasquier, P. 2012. Considering vertical and horizontal context in corpus-based generative electronic dance music. In *Proceedings of the International Conference on Computational Creativity*.
- Fernández-Vara, C., and Osterweil, S. 2010. The key to adventure games design: Insight and sense-making. In *Proceedings of the Meaningful Play Conference*.
- Friberger, M. G.; Togelius, J.; Cardona, A. B.; Ermacora, M.; Moustén, A.; Jensen, M. M.; Tanase, V.; and Brøndsted, U. 2013. Data games. In *Proceedings of the FDG Workshop on Procedural Content Generation*.
- Haklay, M., and Weber, P. 2008. Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE* 7(4):12–18.
- Hoover, A. K.; Szerlip, P. A.; Norton, M. E.; Brindle, T. A.; Merritt, Z.; and Stanley, K. O. 2012. Generating a complete multipart musical composition from a single monophonic melody with functional scaffolding. In *Proceedings of the International Conference on Computational Creativity*.
- Johnson, D., and Ventura, D. 2014. Musical motif discovery in non-musical media. In *Proceedings of the International Conference on Computational Creativity*.
- Krzeczowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation – with intent. In *Proceedings of the International Conference on Computational Creativity*.
- Laclaustra, I. M.; Ledesma, J. L.; Mendez, G.; and Gervas, P. 2014. Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. In *Proceedings of the International Conference on Computational Creativity*.
- Lopes, P.; Liapis, A.; and Yannakakis, G. N. 2015. Targeting horror via level and soundscape generation. In *Proceedings of the Artificial Intelligence for Interactive Digital Entertainment Conference*.
- Martins, T.; Correia, J.; Costa, E.; and Machado, P. 2015. Evotype: Evolutionary type design. In *Proceedings of the International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design*.
- Montfort, N., and Pérez y Pérez, R. 2008. Integrating a plot generator and an automatic narrator to create and tell stories. In *Proceedings of the International Joint Workshop on Computational Creativity*.
- Riedl, M. O., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):76–99.
- Robertson, J., and Young, M. 2015. Automated gameplay generation from declarative world representations. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Thorogood, M., and Pasquier, P. 2012. Computationally created soundscapes with audio metaphor. In *Proceedings of the International Conference on Computational Creativity*.
- Togelius, J., and Friberger, M. G. 2013. Bar chart ball, a data game. In *Proceedings of Foundations of Digital Games*.
- Togelius, J.; De Nardi, R.; and Lucas, S. M. 2007. Towards automatic personalised content creation for racing games. In *Proceedings of the Symposium on Computational Intelligence and Games*, 252–259. IEEE.
- Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the FDG Workshop on Procedural Content Generation*.
- Turner, S. R. 1993. *MINSTREL: A computer model of creativity and storytelling*. Ph.D. Dissertation, University of California Los Angeles.
- Veale, T. 2014. Coming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Proceedings of the International Conference on Computational Creativity*.
- Winkler, W. E. 1999. The state of record linkage and current research problems. In *Statistical Research Report Series RR99/04*. U.S. Bureau of the Census.

Framing Tension for Game Generation

Phil Lopes, Antonios Liapis, Georgios N. Yannakakis

Institute of Digital Games, University of Malta, Msida, Malta
{louis.p.lopes; antonios.liapis; georgios.yannakakis}@um.edu.mt

Abstract

Emotional progression in narratives is carefully structured by human authors to create unexpected and exciting situations, often culminating in a climactic moment. This paper explores how an autonomous computational designer can create frames of tension which guide the procedural creation of levels and their soundscapes in a digital horror game. Using narrative concepts, the autonomous designer can describe an intended experience that the automated level generator must adhere to. The level generator interprets this intent, bound by the possibilities and constraints of the game. The tension of the generated level guides the allocation of sounds in the level, using a crowdsourced model of tension.

Introduction

Several computationally creative systems have stood at the interplay of different multidisciplinary creative domains. It should not come as a surprise, therefore, that several projects in computational creativity tackle the transformation of data from one domain to another, e.g. images to soundscapes (Johnson and Ventura 2014), news articles to collages (Krzeczkowska et al. 2010), academic papers to songs and their lyrics (Scirea et al. 2015), text descriptions to player abilities (Cook and Colton 2014), to name a few. Due to the dissimilarities between source and target creative domains, such computational systems must learn to creatively interpret the patterns of the input, and work towards making them apparent in the output while still obeying the constraints and the expressivity of the target creative domain (e.g. a limited color palette).

In this context, digital games are particularly relevant as a multi-faceted medium where visuals, audio, narrative and rule- and level-design come together in an interactive experience (Liapis, Yannakakis, and Togelius 2014). Not only must these creative domains go well together, but they must provide players with an enjoyable experience: depending on the genre, this experience can be, for instance, frantic in “bullet hell” action games, relaxing in exploration games, or tense in horror games (Ekman and Lankoski 2009).

When drawing inspiration from dissimilar creative domains, it is important to find the right patterns to replicate (or re-interpret) in the creative output of the system. While systems can look at structural similarities and associations (Grace, Gero, and Saunders 2012), a promising approach is

to identify the intentions of the creator of one artefact and attempt to match those intentions in the artefact of the other domain. Towards that outcome, having access to a *frame of reference* for the intentions going into the creative act is ideal. Framing information, as suggested in the FACE model of Colton, Charnley, and Pease (2011), can be provided by the creative system itself as “a piece of natural language text that is comprehensible by people”. Such framing information can clarify the intentions of the system in its design choices and can make its creativity more easily perceptible (Colton 2008). Moreover, the framing information can act as a guide when transforming media generated by such a creative system into different media.

In the context of digital games, a human game designer’s primary concern and frame of reference is the intended player experience. In most games, the intended player experience affects all design decisions: from the color palette to the responsiveness of the controls and from the sound effects for rewards to the back-story presented in an introductory cut-scene. Taking a successful horror game such as *Amnesia: The Dark Descent* (Frictional Games 2010) as an example, the intended player experience is one of dread, of imminent tragedy, of confusion and constant second-guessing of players’ perception and actions. Towards this experience, the visuals include dark colors and dim lights, the audio focuses on ambient noises which foreshadow monsters, the level design has narrow corridors and low visibility while the game rules preclude any way to combat monsters.

This paper extends the *Sonancia* creative system, (Lopes, Liapis, and Yannakakis 2015a; 2015b) by providing the software with the capacity to choose and describe the intended player experience, which is then used to generate game levels and their soundscapes for a horror game. The ability of the computational designer to describe its intentions in clear text to a human audience is paramount in the perception of creativity. Moreover, the system can then create the frame (as the progression of tension) via evolutionary search driven by several fitnesses targeting specific narrative structures. The paper includes several examples of generated frames and their corresponding levels and soundscapes. As an additional contribution to earlier work, the current version of *Sonancia* uses a crowdsourced model of tension to allocate sounds to the level in a way that more closely matches the human perception (or ground truth) of tension.

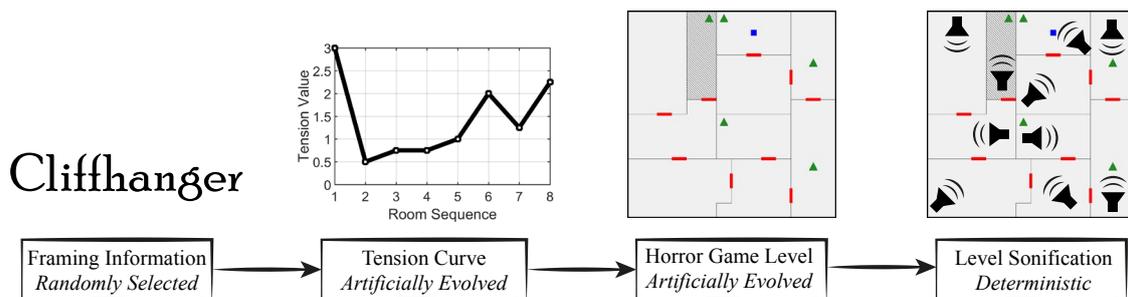


Figure 1: The creative process of the Sonancia system described here: a randomly selected tension frame is used to evaluate an evolving tension progression (the intended tension curve). Once complete, the final intended tension curve guides the evolution of a level generator which attempts to place monsters and items to match the tension curve. Finally, the evolved level and its derived tension curve are used to deterministically allocate sounds based on a crowdsourced model of aural tension. The core innovations of this paper are the framing information and the evolved tension curve (the first two modules); the level generator was first described in (Lopes, Liapis, and Yannakakis 2015a), while sonification has been improved via crowdsourced models.

Background

Sonancia attempts to blend different game facets (in this paper level design, audio and narrative): this section covers related work on blending, focusing on the audio facet.

Blending Game Facets

Digital games are a medium combining different creative facets: visuals, audio, narrative, ludus, level architecture and game-play; these facets complement each other to create specific kinds of interactive experiences (Liapis, Yannakakis, and Togelius 2014). While designing content for each facet is a creative task, blending the different facets is of utmost challenge and promise within computational creativity (Lopes and Yannakakis 2014). Game generation systems like *Angelina* (Cook, Colton, and Pease 2012) and *Game-o-matic* (Treanor et al. 2012) extensively explore how different facets of games can be combined to create interesting and thought-provoking experiences. Commercial games (designed and fine-tuned by humans) tend to blend either their rules (ludus) or level design (architecture), in the case of e.g. action-RPGs. However, suggestions for automating such blends creatively have been put forth (Gow and Corneli 2015). Blends between audio and gameplay have been explored in *AudioInSpace*, where the shooting mechanics change according to the background music, which can be hand-authored (loaded from a music library) or artificially evolved (Hoover et al. 2015). Similar studies have focused on blending audio and narrative in order to foreshadow upcoming story events via sound (Scirea et al. 2014).

The current paper builds upon and extends earlier work on *Sonancia* (Lopes, Liapis, and Yannakakis 2015a; 2015b), by allowing it to autonomously decide on an emotional progression through framing inspired by narrative structures, and by applying a crowdsourcing methodology for the emotional evaluation of sounds in the sonification audio library.

Sound and User Experience

When effectively used, audio has the potential of enhancing the player experience by fully immersing the player within

a virtual world (Collins 2013). This property is especially important within the genre of horror in which particular audio patterns such as musical foreshadowing, the absence of noise, or even a rise of tempo, volume and pitch can elicit stressful experiences for players (Garner, Grimshaw, and Nabi 2010; Ekman and Lankoski 2009). These audio patterns are successful in eliciting intense affective responses if they are well interwoven with the design of the game levels. Earlier work of the authors explored how this could be achieved by sonifying levels based on a common progression of tension (Lopes, Liapis, and Yannakakis 2015a). In those studies each sound asset was given an empirical measure of how tense that particular sound was perceived, allowing the *Sonancia* system to effectively place sounds that accommodate the rise and fall of tension during play (Lopes, Liapis, and Yannakakis 2015b).

Inspired by earlier success of crowdsourcing for annotating highly subjective notions such as game aesthetics (Shaker, Yannakakis, and Togelius 2013), the previous *Sonancia* system (Lopes, Liapis, and Yannakakis 2015b) is extended via crowdsourcing of annotations on tension for sound samples. Such annotations can be used to derive more accurate data-driven computational models of tension in horror games, and offer *Sonancia* a human-verified, objective and more reliable way to select and place sounds to create spooky, tense soundscapes.

Methodology

Sonancia consists of several generative modules working as a pipeline (see Fig. 1): each generator restricts and guides the type of content which can be created in the next generative step, and with each step the content becomes more refined. The final result is a complete horror game, where players must reach a specific room within a haunted mansion while avoiding terrifying monsters along the way (see Fig. 2a). Players do not have weapons and must avoid direct confrontation with monsters; monsters thus act as an instigator of tension and fear, regardless of the player's skill.

Levels in *Sonancia* are generated via evolutionary com-

putation guided by intended tension frames, evolved previously. Every *Sonancia* level consists of rooms connected by doors; rooms can have monsters to be avoided and the objective which must be reached to complete the level (see Fig. 2a). The level is characterized by its *critical path*, which is the shortest sequence of rooms (i.e. shortest path) between the player’s starting room and the room with the objective.

The version of *Sonancia* presented in this paper consists of three different generative modules (see Fig. 1): the framing of tension to a randomly chosen narrative property, the game level generation, and the level sonification module. The details of each module are presented below.

Framing Tension

To clearly explain how levels are created in *Sonancia*, it is important to firstly define how the designer intention is represented within the system. The frame for the task of horror game generation is provided by an *intended tension curve* which consists of a 2D representation of how tension rises and falls as the player progresses along the critical path (see Fig. 2b). In other words, the intended tension curve portrays the ideal player experience when going through the level.

This paper specifically explores how an autonomous creative system can provide a frame to the level generation process by creating different intended tension curves. For horror games, we focus on a frame of *tension* as an amalgam of the predominant emotions within the horror genre (Ekman and Lankoski 2009): fear, anxiety and stress.

Evolving Intended Tension Curves: The intended tension curves are created via a genetic algorithm (GA), driven by one or more aesthetics of narrative progression. The GA allows for flexibility and creativity when defining the curve, but push it towards specific shapes. The tension curve is represented as an array of values between 0 and 3 (in increments of 0.25), where the array index is the room in the order of the critical path, while each value of the array is the specific tension value. Evolution applies a roulette wheel selection mechanism with one-point crossover (Mitchell 1998). After recombination each offspring has a 20% chance of mutating, i.e. incrementing or decrementing a single value in the array by 0.25 (provided the result is within 0 and 3). The GA runs for 100 generations with a population of 100 individuals, each initialized with random tension values.

Evaluating Intended Tension Curves: Eight different fitness functions are encoded into the system, inspired by narrative structures and normalized to $[0, 1]$. The **Escalating** and **Decreasing** tension fitness rewards individuals with rooms that have a higher or lower tension value from the previous room, respectively. The **Resting Point** fitness rewards individuals with the deepest tension ‘valley’, while the **Surprising Moment** fitness rewards the height of the highest ‘peak’. The **Cliffhanger** fitness rewards tension curves with at least one peak, where the last room’s tension is higher than any of the peaks. The **Denouement** fitness gives high values to individuals if the highest peak is close to the final room (but is not the final room). **Unresolved Tension** fitness rewards consecutive rooms with the same tension. Finally the **Rising & Falling Tension** fitness is proportionate to the

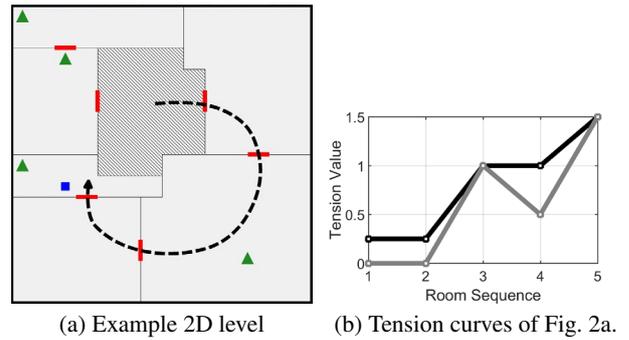


Figure 2: Example of a *Sonancia* “haunted manor” level in 2D (Fig. 2a) and 3D (Fig. 2c). In Fig. 2a, the room with the diagonal lines is the starting room, red rectangles are doors, green triangles are monsters, the blue square is the objective and the black arrow is the critical path (the shortest path between the starting room and objective). The critical path creates a level tension curve (grey) in Fig. 2b which must closely match the intended tension curve (black).

number of peaks in the tension curve. Among these eight fitnesses, one is chosen randomly to generate the appropriate tension frame. To increase the expressivity of generated frames, the system can also choose two fitnesses and apply an “Or” or “And” operator which sums or multiplies, respectively, the individual fitness scores.

The Level Tension Curve: Each level derives a tension curve from the distribution of monsters on the level’s critical path: this process generates the *level tension curve*. Going through each room on the critical path, the level tension curve increases tension by 1 if the room contains a monster; if the room has no monster the tension decreases by 0.5 (to a minimum of 0) to simulate the players relaxing after a stressful event. Figure 2b shows the level tension curve for the level of Fig. 2a.

Level Generation

To create levels that adhere to the frame of intended tension, a search-based PCG approach was chosen (Togelius et al. 2011). The level generation process has been described in (Lopes, Liapis, and Yannakakis 2015a), but a high level description is included in this paper for the sake of com-

pleteness. The level layout is represented as an array of integers (each integral value corresponding to a room’s identifier or ID), while the doors are represented by their connecting rooms’ IDs and monsters or objective by the ID of the room they are in and their type. Mutations allow a gene to change the level layout (pushing walls or splitting rooms), or to add, remove and move doors, monsters or the objective. Crossover is omitted due to its disruptive nature.

Levels construct their own version of the tension curve (i.e. the *level tension curve*), and the fitness function rewards rooms which more closely match the intended tension curve. The fitness function is the average distance between level and intended tension curve (see Fig. 2b). If the level has fewer rooms on the critical path than the intended tension curve is scaled, while if it has more rooms then it receives a minimal fitness as the intended tension curve acts as a constraint on room number. In addition, the fitness function also calculates the number of unique rooms visited between the start (room ID 0) and all “dead-end” rooms and subtracts the number of rooms with no doors (as those can not be visited). More details on the evolutionary algorithm and objectives can be found in (Lopes, Liapis, and Yannakakis 2015a).

Level Sonification

Level sonification in the *Sonancia* system consists of allocating specific audio pieces within the level, based on the level tension curve. The goal of sonification is to have sounds which match the tension of the room, i.e. rooms with monsters will have scarier associated music; this is different from (Lopes, Liapis, and Yannakakis 2015b) which used sonification for suspense as the reverse of tension. *Sonancia* includes a soundbank of human-authored recordings with an average length of 7 seconds. To accurately map sound assets to specific values of tension, a crowdsourcing experiment was conducted to obtain an approximation of how tense the different sounds are compared to each other.

The Sound Library: The *Sonancia* sound library currently contains 97 different sound assets, recorded by human authors via the *FM8* (Native Instruments 2006) tool and the *Reaper* (Cuckos 2005) digital audio workstation. To maintain a large, yet feasible number of samples for crowdsourced annotation, we undersampled sounds from the library based on their “pitch” and “loudness”.

According to Garner, Grimshaw, and Nabi (2010) loud (i.e. power) and high-pitched sounds tend to trigger fearful emotions. Based on this finding, we plotted (see Fig. 3) each audio asset according to the ΔDb value (loudness) and average power of frequencies above 5k (high-pitch). For the crowdsourcing experiment presented in this paper we selected the 40 sounds (out of the 97 available) with the highest average Euclidean distance between them along pitch and loudness (see Fig. 3).

Crowdsourcing Tension: A survey was conducted to derive an approximate value of reported tension to each sound asset in the library¹. For annotating the tension value of sounds we adopt a rank-based approach due to its evidenced

¹sonancia.institutedigitalgames.com

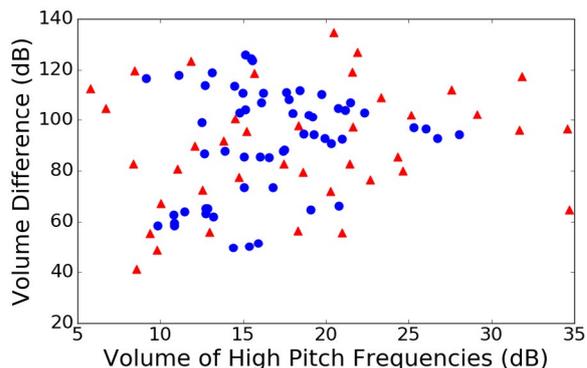


Figure 3: Scatter plot of the entire *Sonancia* sound library. High pitch frequencies are between 5 and 22 Hz, while volume difference is between the maximum and minimum dB values of the sound. Triangles and circles are, respectively, selected and unselected audio samples.

effectiveness for highly subjective notions such as affect and emotion (Yannakakis and Hallam 2011). Human annotators were presented with pairs of sounds selected randomly and were asked to report which sound in each pair is more tense via a 4-alternative forced choice questionnaire (Yannakakis and Hallam 2011). Annotators could listen to the two selected sounds as many times as they desired. At the time of writing, 452 pairs of sounds have been ranked by tension. While this is a smaller number than the 780 possible pairings, the sound pairs were randomized and thus all sounds were annotated for at least half of the possible pairings; some insight on every sound’s tension properties can be gleaned even with the limited data.

The 40 sounds are ranked based on the human-annotated tension preferences. The *global order* of sound tension is derived through the pairwise preference test statistic (Yannakakis and Hallam 2011) which is calculated as $P_i = (\sum_j z_{ij})/N$, where z_{ij} is the tension preference score of i in the pair of sounds i and j (z_{ij} is +1 if sound i is preferred, -1 if sound j is preferred, and 0 if no sound is preferred or there is no annotation); N is the total number of sounds. The obtained tension preference scores P define the global order (rank) of each sound with respect to tension.

Audio Allocation and Mixing: Audio allocation consists of placing sound assets in each room of a level, based on the level tension curve and the tension preference score of each sound in the library. The system picks sounds equidistantly from the global order (in descending tension preference score) depending on the total number of rooms (not only those in the critical path). A sound is assigned to each room so that the room’s tension value matches the global order of sound tension. The process starts with the most tense sound which is allocated to the room with the highest tension and it continues until no more rooms (or sounds) are available and each room has a unique sound. Higher ranked sounds with respect to tension are prioritized for rooms on the critical path. For rooms with equal tension values, the

first room in the critical path gets the more tense sound.

The mixing algorithm controls how the sounds are played in the game. Audio mixing uses the player’s distance from a neighbouring room to adjust the volume of the contribution from each neighbouring room’s sound. This mixing rule allows players to hear sounds from neighbouring rooms, offering a sense of foreshadowing.

Experiments

This section describes results obtained from *Sonancia*’s entire process, from creating a framing of tension to generating levels based on this frame and finally sonifying the level. The goal is to evaluate, in a qualitative way, how the different generators interpret (in a tension graph, level structure or sound sequence) a frame of increasing detail created by the previous generative step in the pipeline of Fig. 1. The discussion of results assesses how accurately, for instance, the levels match the tension curves and where the limitations of one generative domain lead to a creative transformation of the other domain’s data.

The system ran independently 40 times, where the framing fitnesses were selected (and often combined) by the system without human intervention. Once a framing fitness is selected, intended tension curves evolved for 100 generations in 20 independent runs; the fittest one among these runs is selected to guide level generation. Level generation performed 20 independent runs for 100 generations using the intended tension curve found previously. For brevity, we discuss the fittest individuals (tension curves, levels and soundscapes) for four chosen generated frames; these provide the most varied and interesting results. The highlighted system’s frames were provided in text as such:

1. “I want an experience with a denouement.”
2. “I want an experience with a cliffhanger.”
3. “I want an experience with both a surprising moment and a point of rest.”
4. “I want an experience with decreasing tension or a cliffhanger.”

The following sections describe (in the above order) the tension curves, levels and soundscapes created following this computer-generated frames of tension.

Framing Denouement

Denouement (or conclusion) is encoded aesthetically as a fitness function which rewards when the highest peak in the tension curve is near the last room (but not the last room, as that would not form a ‘peak’ per se). Observing the fittest intended tension curve in Figure 4c, the intended tension curve (in black) matches this specification as the highest peak (at 2.5) is on the 7th room out of 8 rooms on the critical path.

Level Generation: Figure 4a shows the fittest level for the intended tension curve discussed above; its level tension curve is shown in Fig. 4c, in grey. It is immediately obvious that the level tension curve does not match the intended one closely, although it does have a single peak at room 4 and a denouement of 4 rooms after that (rooms 5-8). The level

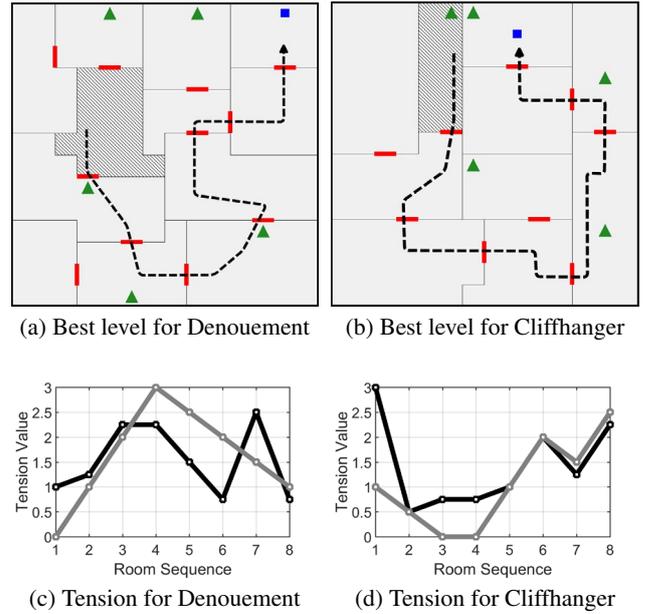


Figure 4: Haunted mansions and their intended and actual (level) tension curves for single aesthetics.

Room	1	2	3	4	5	6	7	8
Rank	22	16	7	1	4	10	13	19
P	0.04	0.13	0.44	0.79	0.67	0.24	0.19	0.05

(a) Denouement

Room	1	2	3	4	5	6	7	8
Rank	10	16	19	22	13	4	7	1
P	0.24	0.13	0.05	0.04	0.19	0.67	0.44	0.79

(b) Cliffhanger

Table 1: Sound selection for the Denouement (Table. 1a) and Cliffhanger (Table. 4b) levels. The sounds’ corresponding rank position with respect to tension (Rank) and tension preference score (P) are also presented.

cannot match the intended tension curve since e.g. monsters always add 1 to the tension and decay does not allow the ‘constant’ tension between rooms 3 and 4 or the quick drops of rooms 5 and 6. Instead, evolution attempts to balance the tradeoffs between monsters and tension decay, by adding or removing monsters in specific rooms. The result in Fig. 4a contains 3 monsters in the first three rooms after the starting one, and then no monsters until the objective room — allowing the player to relax. The biases and constraints of the level generation forced evolutionary search to interpret the intended tension curve to the best of its ability; the level tension curve does exhibit denouement, albeit lasting longer.

Level Sonification: Table 1a shows the distribution of different audio assets and their respective rank value within the level of Fig. 4a. It is important to note that sonification will always follow the level tension curve, to create a

soundscape coherent to the current level. In this instance the algorithm places the highest ranked sound at the tension peak (i.e. room 4). Room 3 has a higher ranked sound than room 6 even though they have the same tension values, as it occurs first on the critical path. A game-play video of this level is available online².

Framing the Cliffhanger

The cliffhanger is encoded aesthetically as a fitness function which rewards tension curves with at least one peak, where the last room’s tension is higher than any of the peaks (acting, thus, as the cliffhanger). The fittest intended tension curve in Figure 4d (in black) matches this specification as the tension peaks in the 6th room (with a value of 2.0) but the final room is even more tense (2.25). It should be noted that the curve starts at the highest value (3.0) in room 1; this is due to the fact that the first room does not register as a peak (peaks compare tension values with both neighbours).

Level Generation: Figure 4b shows the fittest level for the intended tension curve discussed above; its level tension curve is shown in Fig. 4d, in grey. Unlike denouement, the level tension curve closely matches the intended one for the cliffhanger aesthetic. Both curves start at the maximum possible tension (for levels, this is 1 if there is a monster in the first room) and then drop the tension in the next rooms only to increase it around rooms 5 and 6, culminating at the highest value (ignoring the first room in the intended curve) on room 8. Interestingly the level curve drops to 0 in rooms 3 and 4 as it can not maintain the near-identical tension of the intended curve (due to tension decay).

The result in Fig. 4b has 4 monsters on the critical path, distributed near the start and end of this path. This causes an initial tense moment for the players when they start the level, then lets them relax with 3 empty rooms, reach a climax after two monsters and release some tension with the next-to-last room only to find a monster in the room with the objective.

Level Sonification: Table 1b contains the sounds allocated along the critical path of the level in Fig. 4b). As the level tension curve closely matches the intended curve, sonification largely matches the original frame as well. While rooms 2 to 5 have sounds with a low global rank value, this changes swiftly with tense sounds which culminate to the most tense sound in the last room. A game-play video of this level is available online³.

Frame of Surprising Moments and Resting Points

When combining fitness functions, the “and” combination forces both fitnesses to have high scores: in this case, the surprising moment aesthetic rewards high ‘peaks’ while the resting point aesthetic rewards deep ‘valleys’. Indeed, both aesthetics are present in the intended tension curve of Fig. 5c as it exhibits the highest peak (height of 3) and the lowest possible valley (depth of 3, considering the tallest adjacent peak). The aggressive changes in tension were expected, as both fitnesses directly reward high peaks and deep valleys;

²<https://youtu.be/IJQFqxfHqY8>

³<https://youtu.be/z5R12NPVVFA>

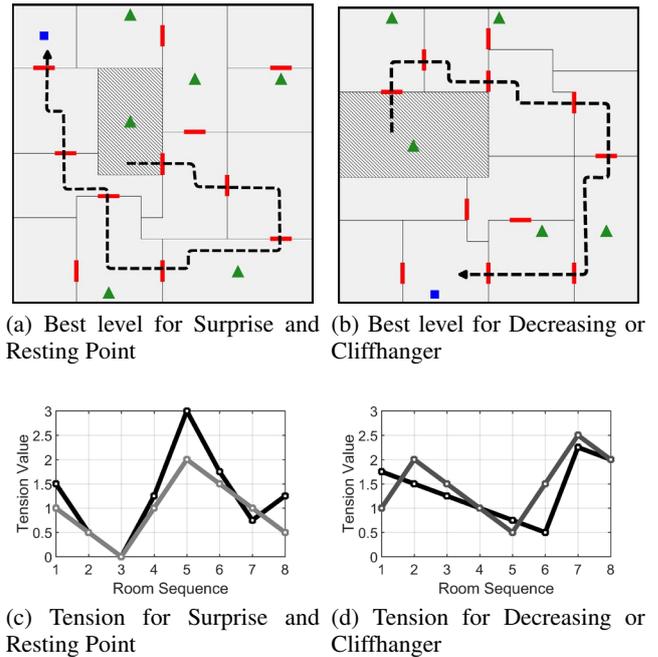


Figure 5: Haunted mansions and their intended and actual (level) tension curves for combined aesthetics.

Room	1	2	3	4	5	6	7	8
Rank	7	16	22	10	1	4	13	19
<i>P</i>	0.44	0.13	0.04	0.24	0.79	0.67	0.19	0.05

(a) Surprising Moments and Resting Points

Room	1	2	3	4	5	6	7	8
Rank	16	4	10	19	22	13	1	7
<i>P</i>	0.13	0.67	0.24	0.05	0.04	0.19	0.79	0.44

(b) Decreasing Tension or a Cliffhanger

Table 2: Sound selection for the Surprising Moments and Resting Points (Table 2a) and Decreasing Tension or a Cliffhanger (Table 2b) levels. The sounds’ corresponding rank position with respect to tension (Rank) and tension preference score (*P*) are also presented.

their combination unsurprisingly causes tension to soar from a value of 0 to 3 within the span of two rooms. In many other runs, the fittest individuals contained adjacent rooms with tension values of 0 and 3 (or vice versa).

Level Generation: Figure 5a shows the fittest level for the intended tension curve discussed above; its level tension curve is shown in Fig. 5c, in grey. The level tension curve matches the intended tension curve as closely as possible given the constraints of the way it is computed. The fact that each room can have only one monster (which increases tension by 1) causes the peak of room 5 after the resting point in room 3 to have lower tension values than the intended. The structure of the level tension curve retains both a resting

point at room 3 and the surprising moment at room 5, and thus matches the provided frame. Of interest is the observation that unlike the intended curve, the last room in the level does not contribute to an increase in tension since adding a monster there (getting the tension value to 2) would cause more deviation from the intended value (1.25).

The level of Fig. 5a has 3 monsters on the critical path, placed primarily midway to the objective. This yields a high spike (surprising moment) in room 5 after encountering two monsters. The starting room also has a monster, in order to let players relax in the next two rooms and thus reach the resting point (before more stressful events) in room 3.

Level Sonification: Table 2a contains the sounds allocated along the critical path of the level in Fig. 5a. Obviously, the most tense sound is placed on the surprising moment (room 5) which matches both the intended and the level tension curve; similarly, the resting point has the least tense sound as per the provided frame. Due to the similarity of the level curve with that of Fig. 4c, tense sounds are allocated in a somewhat similar fashion with a slight change in the first rooms. A game-play video of this level is available online⁴.

Framing Decreasing Tension or a Cliffhanger

Combining fitness functions with an “or” in this system adds the two fitness scores together. This will still reward the presence of both features but since it is less aggressive than multiplying the scores (as in “and”), it may reward either fitness equally. The fittest tension curve in Fig. 5d, for instance, does not have the cliffhanger pattern (although partially it does exhibit a peak in room 7) but has predominantly a decreasing tension. The cliffhanger and decreasing tension are conflicting objectives, as the former rewards an increase in tension both for the presence of a peak and for the final room. Therefore, the intended curve in Fig. 5d attempts to balance between the two by predominantly having a decreasing tension while also having a peak (which is rewarded, partially, by the cliffhanger fitness). Thus, the less aggressive search of the “or” operator is demonstrated.

Level Generation: Figure 5b shows the fittest level for the intended tension curve discussed above; its level tension curve is shown in Fig. 5d, in grey. The level tension curve matches the intended tension curve except that the gradient of the tension decay is different: this causes evolution to use two rooms (6 and 7) to increase the tension in order to match the tension value of room 7 (2.25 in the intended curve and 2.5 in the actual one). Interestingly, despite the expected differences when the level generator interprets the intended frame (e.g. an increase in tension at room 2), the aesthetics match between intended and level tension curve. The level tension curve predominantly has decreasing tension, with no cliffhanger but at least one peak (thus fulfilling one of the requirements for a cliffhanger).

The level of Fig. 5b has 4 monsters on the critical path, placed at the start and towards the end of the critical path. The two monsters in the first and second rooms trigger a

very tense experience to the player, but the decreasing tension aesthetic allows them to relax for the next 4 rooms before facing two more monsters in rooms 6 and 7. The room with the objective does not have a monster, affording some relaxation to the player.

Level Sonification: Table 2b shows how sounds were allocated within the level and their respective values. Compared to the other sonification results, this soundscape spreads highly tense sounds throughout the level rather than concentrating them in a specific section. Interestingly, the level tension curve is unique compared to the other cases as no room has a tension value of 0. For instance, room 2 has the second highest ranked sound, but is surrounded by less tense sounds, while the most tense sound is reserved for the climax (i.e. room 7). A game-play video of this level is available online⁵.

Discussion

The results highlighted four example tension frames which were associated with one or multiple fitness dimensions. The results showed that the intended tension curves created by the system matched the patterns in the narrative structures they were based on. An exception was when conflicting fitnesses were combined with the “or” operator, where one fitness could dominate the other (earning the operator its name). The generated levels in many cases matched the intended curve (if not value-for-value) but the limitations of the level tension curve calculation could cause deviations (e.g. in the case of denouement). At a high-level, all generated levels exhibited the intended aesthetics of each frame.

Observing results with other fitness dimensions of framing, we found that *Escalating*, *Decreasing* and *Unresolved Tension* fitnesses created the least variability in the tension curves. This was expected, as these fitnesses reward small incremental changes in the tension or no changes (for *Unresolved Tension*). Both the *Surprising Moment* and *Resting Point* fitnesses created more variations in the tension curves but both showed similar patterns: a drastic change of tension (from 0 to 3 or vice versa) between two adjacent rooms (similar to Fig. 5c). This pattern is impossible to replicate in the levels, leading to more free-form interpretation of the intended curve by the level generator. An interesting emergent solution to attain less aggressive tension changes was when fitnesses were combined: for instance, combining any fitness with the *Escalating* or the *Decreasing* fitness yielded curves with smoother changes in tension. Due to a less strict evaluation formula, the *Denouement*, *Cliffhanger* and *Rising & Falling Tension* created the most diverse curves. Peaks very often varied in tension, and in some cases the entire curve would have low values of tension, or only high values.

The additional modules of the *Sonancia* pipeline (highlighted in Fig. 1) contribute to the creativity of the system in two core ways: *framing information* and *interpretation*. Framing information (as desired narrative structures) allow the generator to describe in human language its intent; the fitness function associated with each narrative structure al-

⁴<https://youtu.be/P2HkGr719f0>

⁵https://youtu.be/JnFli_F-r38

lows the system to *appreciate* whether it has achieved this intent. The fact that the initial frame is chosen randomly is a current limitation of the system; its creativity could be strengthened if the inspiration for a narrative structure comes from elsewhere (e.g. a newspaper article). Interpretation in *Sonancia* is strengthened by extending the pipeline to include generated tension curves which guide the level generator, which in turn guides the sonification process: as the level of detail of the creative artifact increases from an abstract frame to a playable game, the generators must creatively interpret the guidelines of the previous generative step in order to satisfy them while still obeying the limitations of their own level of detail (e.g. the structural requirements of a level). This requires a degree of *imagination* from each module in transforming high-level directives into higher-detail artifacts. Finally, as the final game levels are always playable and contain the necessary components for horror gameplay, *Sonancia* has the necessary *skill* and thus completes the creative tripod exhibited by creative systems (Colton 2008).

Conclusions

This paper presented a system capable of creating and combining different structures of tension influenced by narrative concepts, then transforming them into horror levels and their accompanying soundscapes. Several dimensions of tension framing structure were developed and tested; the four example generated frames highlighted the process from frame conceptualization to level generation and finally to sonification. Demonstrations of the sonification of all the example levels in this paper can be found online⁶.

Acknowledgments

The authors would like to thank all participants of the crowdsourcing experiment. This work was supported, in part, by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665) and the Horizon 2020 project Cross-Cult (project no: 693150).

References

- Collins, K. 2013. *Playing with sound: a theory of interacting with sound and music in video games*. MIT Press.
- Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The face and idea descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*.
- Colton, S. 2008. Creativity vs. the perception of creativity in computational systems. In *Papers from the AAAI Spring Symposium on Creative Intelligent Systems*.
- Cook, M., and Colton, S. 2014. A rogue dream: Automatically generating meaningful content for games. In *Proceedings of the AIIDE workshop on Experimental AI & Games*.
- Cook, M.; Colton, S.; and Pease, A. 2012. Aesthetic considerations for automated platformer design. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment conference*.
- Ekman, I., and Lankoski, P. 2009. Hair-raising entertainment: Emotions, sound, and structure in silent hill 2 and fatal frame. *Horror video games: Essays on the fusion of fear and play* 181–199.
- Garner, T.; Grimshaw, M.; and Nabi, D. A. 2010. A preliminary experiment to assess the fear value of preselected sound parameters in a survival horror game. In *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, 10. ACM.
- Gow, J., and Corneli, J. 2015. Towards generating novel games using conceptual blending. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Grace, K.; Gero, J.; and Saunders, R. 2012. Representational affordances and creativity in association-based systems. In *Proceedings of the International Conference on Computational Creativity*.
- Hoover, A. K.; Cachia, W.; Liapis, A.; and Yannakakis, G. N. 2015. AudioInSpace: Exploring the creative fusion of generative audio, visuals and gameplay. In *Proceedings of the EvoMusArt conference*. Springer. 101–112.
- Johnson, D., and Ventura, D. 2014. Musical motif discovery in non-musical media. In *Proceedings of the International Conference on Computational Creativity*.
- Krzeczowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation – with intent. In *Proceedings of the International Conference on Computational Creativity*.
- Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *Proceedings of the International Conference on Computational Creativity*, 285–292.
- Lopes, P., and Yannakakis, G. N. 2014. Investigating collaborative creativity via machine-mediated game blending. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment conference*.
- Lopes, P.; Liapis, A.; and Yannakakis, G. N. 2015a. *Sonancia*: Sonification of procedurally generated game levels. In *Proceedings of the 1st Computational Creativity and Games Workshop*.
- Lopes, P.; Liapis, A.; and Yannakakis, G. N. 2015b. Targeting horror via level and soundscape generation. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Mitchell, M. 1998. *An introduction to genetic algorithms*. MIT press.
- Scirea, M.; Cheong, Y.-G.; Nelson, M. J.; and Bae, B.-C. 2014. Evaluating musical foreshadowing of videogame narrative experiences. In *Proceedings of the Audio Mostly: A Conference on Interaction With Sound*. ACM.
- Scirea, M.; Barros, G. A. B.; Shaker, N.; and Togelius, J. 2015. SMUG: Scientific music generator. In *Proceedings of the International Conference on Computational Creativity*.
- Shaker, N.; Yannakakis, G.; and Togelius, J. 2013. Crowd-sourcing the aesthetics of platform games. *IEEE Transactions on Computational Intelligence and AI in Games* 5(3).
- Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *Proceedings of the Computational Intelligence and Games Conference* 3(3):172–186.
- Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the FDG workshop on Procedural Content Generation in Games*, 11. ACM.
- Yannakakis, G. N., and Hallam, J. 2011. Ranking vs. preference: a comparative study of self-reporting. In *Affective computing and intelligent interaction*. Springer. 437–446.

⁶<https://goo.gl/Qshvqv>

What If A Fish Got Drunk?

Exploring the Plausibility of Machine-Generated Fictions

Maria Teresa Llano¹, Christian Guckelsberger¹, Rose Hepworth¹
Jeremy Gow¹, Joseph Corneli¹ and Simon Colton^{1,2}

¹Computational Creativity Group, Goldsmiths, University of London, UK

²The Metamakers Institute, Falmouth University, UK

m.llano@gold.ac.uk

Abstract

Within the WHIM project, we study *fictional ideation*: processes for automatically inventing, assessing and presenting fictional ideas. Here we examine the foundational notion of the *plausibility* of fictional ideas, by performing an empirical study to surface the factors that affect judgements of plausibility. Our long term aim is to formalise a computational method which captures some intuitive notions of plausibility and can predict how certain types of people will assess the plausibility of certain types of fictional ideas. This paper constitutes a first firm step towards this aim.

Introduction

In Llano et al. (2016), we define a successful fictional idea as “one that presents a character, event or scenario that transforms or distorts the ‘real’ world in the imagination of the reader without requiring him or her to leave it entirely”. In the WHIM project (an acronym for The *What-if Machine*), we are undertaking the first large-scale study of how software can invent, evaluate and express fictional ideas with real cultural value (www.whim-project.eu). We have identified *plausibility* as one of the key dimensions of fictionality, and so investigating questions of plausibility is important for the aims of the WHIM project. Unfortunately, plausibility resists a simple definition. Here, we explore the factors that support the perception of a machine-generated fictional idea as plausible or implausible.

Plausibility in fictional scenarios is different from notions of probability, which rely on modelling situations in terms of relative frequency, or the updating of prior distributions. Judgements about plausibility in fictional situations involve a process of *interpretation*, where the reader makes – perhaps implicit – subjective decisions about the underspecified fictional universe. For example, in the absurdist play *Rosencrantz and Guildenstern are Dead*, Rosencrantz bets “heads” on a coin flip 92 times in a row, and wins each time. If presented as a factual news story, this would likely both be judged as implausible and mathematically improbable (albeit no more improbable than any other session of 92 coin flips). However, within the context of the play, we’re invited to consider a fictional world in which this highly improbable chain of events is plausible, i.e., it actually happens.

Following an overview of prior research, we propose some candidate factors to capture how people assess the

plausibility of fictional statements. We then report on how these were used in an exploratory study, where 20 participants were asked to categorise a set of fictional ideas – both machine- and human-generated – into four plausibility categories and interviewed about their judgements. We then examine these judgements from two distinct perspectives: (1) a grounded theory analysis that identifies several factors they considered relevant to plausibility; (2) a multidimensional scaling analysis that suggests several factors that can explain differences between how fictional ideas were assessed. Finally – as per Rothbauer (2008) – we triangulate the outcomes of these analyses and our initial theories, leading to a final set of factors. We conclude with a discussion of the relevance of this work for further research on fictional ideation and Computational Creativity in general.

Background

Connell and Keane (2004) carried out an empirical study of plausibility. They first evaluated the *concept coherence* of a set of events written as two connected sentences, e.g.,

The bottle fell off the shelf. The bottle smashed.

These were classified according to different types of inferences; the sentence above references a *causal inference*, while the sentence pair

The bottle fell off the shelf. The bottle was pretty.

references an *attributive inference*. Their results supported the received view that concept coherence is important in plausibility judgements, and showed that different inference types differentially affect plausibility. A second experiment evaluated *word coherence* rather than concept coherence, but this experiment found no reliable effect of word coherence on plausibility.

Connell and Keane note that their studies used “the term plausible interchangeably with other descriptions such as appropriate, sensible, or makes sense.” Our approach differs considerably from theirs, as they focused on statements whose two constituent parts have a strong conceptual relation. In contrast, as the fictional statements evaluated in this paper have not been conceived with such restrictions, this has allowed for a more comprehensive analysis.

Lombardi, Nussbaum, and Sinatra (2015) have sought to outline a primarily theoretical model for plausibility judgements. In particular, they examine the role such judgements

play in *conceptual change*, and come to understand plausibility as meaning: ‘what is perceived to be potentially truthful when evaluating explanations’. They cite Nicholas Rescher’s observation that a statement deemed to be plausible, or potentially truthful, indicates that there has been a ‘highly provisional and conditional epistemic inclination towards it’ (Rescher 1976).

In sum, there seem to be myriad criteria upon which people may base judgements of plausibility. Some factors in an individual’s judgement appear to be highly subjective, often being heavily influenced by personal circumstances, religious belief, cultural background, or political sympathies. Indeed, while most people feel they have an understanding of plausibility, that understanding is almost always destabilised when an individual tries to apply consistent criteria to analyse the plausibility of various sample statements.

Plausibility is closely connected with the notion of *interpretation*: that is, a given interpretation of a given scenario is deemed “plausible” if potentially valid, under a given set of assumptions. Interpretation of conventional symbols is highly constrained. However, creative interpretations may be almost endlessly fanciful. To take one example: the science fiction author Philip K. Dick suggests in several of his written works that we continue to live in biblical times, evidence of which can only be accessed by visionary experience (Dick 1995). This interpretation of the world is grounded in the data of personal perception and reflection. Nevertheless, most people would find such an interpretation implausible if presented as anything other than fiction.

Eco argued that the world, history, and texts have constraints on their plausible interpretations, despite the wide range of *possible* interpretations. He posits that fiction and reality intersect in the following way: “We can make true statements about fictional characters because what happens to them is recorded in a text, and a text is like a musical score” (Eco 2009). Music shows us that a creative work can take on a life of its own through interpretation. Fictional characters can also “become individuals living outside their original scores,” or, to put it more formally: “a fictional character is a semiotic object.”

We can thus make chains of interpretations about fictional characters and other elements of fictional worlds. In the first instance, the validity of such interpretations is not “grounded” in real-world facts, but in the fictive notions of the fictional world – subject also to the perceptions, beliefs, and other features of the interpreting agent. When instigating behaviour (including storytelling behaviour), certain interpretations may be predicted, based in part on a preliminary interpretation of those agents who are expected to perceive the behaviour (Kockelman 2012).

We believe it is important for Computational Creativity researchers to tackle issues related to fictional interpretations, in particular to ask what kinds of interpretations are *useful* (Eco 2006) – rather than merely true. Although subjectivity plays a role, keeping in mind Eco’s remarks on limits of interpretation, we think that the reader’s perception of a text’s plausibility will often draw on the text’s objective features, and we develop this theme below.

Candidate Factors

As a first step, we conducted an introspective study to identify an initial set of factors that may be involved in human plausibility judgements. These candidate factors helped guide the design of the exploratory study, described below. Eight fictional statements were used. Four were from the What-If Machine and four were summaries of well known literary works, included to foster the generalisability of the findings. The machine-generated statements were selected for quality and diversity, to showcase a range of potentially relevant factors.

Three of the authors independently read the statements and rated each sentence 1–5 in terms of how plausible they were (1= low plausibility and 5 = high plausibility). They also wrote a commentary for each statement, describing their rationale for that score, their scoring process (including whether they had revised a score during analysis), and a set of labels that described relevant properties, dimensions or features. By comparing our individual answers, we found a common set of factors that appeared to affect our plausibility judgements, listed below.

Complexity The level of elaboration of the idea in terms of the amount of narrative detail that it is composed of. Our intuition is that a larger number of statements, or narrative details, used to compose an idea reduces its plausibility. An illustrative example is the statement:

“What if there was a poor orphan girl who was abused by her aunt, sent away to school where conditions were harsh, before becoming a governess and marrying her employer, who was already married to a mentally ill woman whom he has locked up in his house?” (1)

Each part is rather plausible, but their conjunction renders the overall idea less plausible. *We hypothesise that there is a negative correlation between complexity and plausibility.*

Universality The scope of an idea, in terms of how general people think it is intended to be. In other words, whether the scenario in the fictional idea applies to one, a few or all members of a group. The intuition behind this is that an idea that is generalised to a large number of members of a group is less plausible than an idea that only involves one member. For instance, from the statement:

“What if there was a young girl who went through a rabbit hole and found herself in a strange and mysterious land where animals could talk and everyone is mad?” (2)

The implicit universally quantified sentences “animals could talk” and “everyone is mad” decrease the plausibility of the idea. *We hypothesise that there is a negative correlation between universality and plausibility.*

Openness How open to subjective interpretation an idea is perceived to be. Our intuition is that if an idea that is composed of statements that are ambiguous or not specific, for which the reader can provide different interpretations or scenarios, is perceived as more plausible. As an example, take the statement:

“What if there was a young man who kept a painting of himself which aged while he himself stayed young?” (3)

Here, “stayed young” could be interpreted both in terms of not looking old or actually not ageing, while the painting that “aged” could have been painted in highly impermanent materials. The possible explanations (natural youthfulness and cheap paint, or a bizarre medical condition) differ strongly in their plausibility; if there is a choice, a subject might choose the more plausible explanation. *We hypothesise that there is a positive correlation between openness and plausibility.*

Causality The level of connectivity between the components that make up an idea. In other words, how naturally the statements that make up an idea lead coherently from one to the other. The intuition behind this is that plausibility increases when the statements of an idea are clearly connected so as to serve as supporting arguments themselves. To illustrate this, take the example:

“What if there was a little doctor who couldn’t take a pulse?” (4)

Without an explanation as to why the doctor is unable to perform the common task of taking a pulse, this statement will likely score low for plausibility. *We hypothesise that there is a positive correlation between causality and plausibility.*

Familiarity The level of awareness of the overall scenario relative to known ideas. Although this is a subjective factor, the intuition behind it is that our perception of plausibility is affected by common themes, scenarios and characters that figure more commonly in culture. Statement (1) illustrates this intuition. Well-known character stereotypes such as *an orphan girl, an evil aunt and a mentally ill woman* render the statement more plausible. *We hypothesise that there is a positive correlation between familiarity and plausibility.*

Feasibility How well the elements within an idea fit within the overall scenario. The intuition behind this is that plausibility increases if an element; e.g., a character, is better suited to one situation than another. To illustrate, the statement:

“What if there was a little cat who learned how to use a phone?” (5)

Would rank lower in plausibility if instead of *a cat*, the subject was an inanimate object, for instance *a cooker* due to the affordances of the subjects. *We hypothesise that there is a positive correlation between feasibility and plausibility.*

An Exploratory Categorisation Study

To further explore the factors underlying human plausibility judgements, we conducted a categorisation study where the above candidate factors guided the selection of the stimuli for, and design of, the study. In the study, participants were asked to assign machine- and human-generated fictional ideas into different categories of plausibility. We collected both quantitative and qualitative data from these sessions, which were then separately analysed and interpreted: 1) the raw categorisation results were used to calculate explorative statistics; 2) participant think aloud commentaries and post-task interviews formed the basis of a grounded theory analysis (Adams, Lunt, and Cairns 2008), identifying key factors in their categorisation process. 3) The categorisation results were transformed into similarity data for a

multidimensional scaling (MDS) analysis (Borg and Groenen 2005) of the fictional ideas, to collect more evidence on the underlying dimensions which influenced the plausibility categorisations; By studying the same plausibility judgements qualitatively and quantitatively, we hoped to triangulate the results to arrive at a final set of factors. A categorisation task with physical cards, as opposed to ordinary Likert-scale rating, was deliberately chosen to promote think-aloud comments and the comparison of stimuli. These methods are well-suited to exploring complex and poorly conceptualised domains, e.g., see Wallraven et al. (2009) or Gow et al. (2010).

Stimuli We used 28 fictional ideas in total, consisting of 18 ideas generated by The What-If Machine (three from each of the six categories the system currently supports: “Disney”, “Metaphors”, “Utopian/Dystopian”, “Alternative Scenarios”, “Kafkaesque” and “Musicals”), 7 ideas summarising well-known fictional literature works (“Literary Fiction”), and 3 ideas that used known fictional characters or worlds (“Fiction in Fiction”). We also selected a subset of six ideas (from the 28 already selected) for the participants to verbally elaborate on in more detail. A selection of stimuli from each category can be found in Table 1.

Method Participants took part in the study individually and were all read the same introductory material. Each session was audio recorded for later analysis. We first asked them to sort the 28 stimuli, provided as paper cards, into four plausibility categories:

1. **Highly implausible:** describe scenarios that have very little grounding in your experience of reality.
2. **Slightly implausible:** describe scenarios that have a low degree of grounding in your experience of reality.
3. **Slightly plausible:** describe scenarios that are somewhat grounded in your experience of reality.
4. **Highly plausible:** describe scenarios that have a high degree of grounding in your experience of reality.

An “**I don’t understand**” category was also provided. In contrast to our introspective study, we chose four categories to eliminate the neutral choice. Participants were not told that some of the statements had been written by software, nor asked if they recognised those from human-authored narratives.

We asked participants to think aloud while performing this task, i.e., to articulate their categorisation process and rationale. For some participants (see below) this was followed by open-ended questions where these issues were probed in greater depth, focusing on the six statements which we had pre-selected, or others highlighted during the categorisation study. Finally, we asked some participants explicit questions about our candidate factors (described above), to determine if they considered them relevant. For instance, regarding *complexity* we asked: “Do you believe that a complex fictional statement; that is, with a large amount of conditions, makes the plausibility higher, lower or neither?”. Each such question was accompanied by an example statement.

Id	Mean	Var.	Ag.	NAs	Stimulus	Category
5	2.00	1.44	14	0	What if we could give life to a being created by combining the body parts of dead people?	Literary Fiction
8	0.39	0.72	16	1	What if a zombie rugby-tackled a ghost and broke his leg?	Fiction in Fiction
12	0.38	0.78	14	3	What if there was a little pen who forgot how to write?	Disney
14	2.50	0.62	15	1	What if ignorant fools were to overcome mistakes, establish cults and become knowledgeable gurus?	Metaphors
19	1.69	1.03	9	3	What if the world suddenly had lots more assassins? Then there would be more antidotes, since assassins use the poisons that require antidotes.	Utopia / Dystopia
20	0.44	0.61	15	1	What if there was an old fish, who couldn't swim anymore, which he used to do for relaxation, so decided instead to get drunk?	Alternative
21	0.94	1.31	11	2	What if there was an old car that could be used as the space for holding a star?	Alternative
24	0.17	0.26	17	1	What if a bicycle appeared in a dog pound, and suddenly became a dog that was able to drive an automobile?	Kafkaesque
26	2.72	0.21	18	1	What if a wounded soldier had to learn how to understand a child in order to find true love?	Musicals
28	2.06	0.43	14	2	What if a janitor needed to suppress a rebellion in order to gain admiration?	Musicals

Table 1: A selection of stimuli, with response mean and variance. Ag.= participants agreeing with most common response. NAs = times classified as “Don’t understand”.

Participants In total, 20 participants took part in the study, although one participant’s data was excluded (see below). Of the remaining 19, 4 participants were female and 15 male. 8 participants were in the age range 18-24 years old, 9 were 25-34, and 2 were 35-44. 4 of them specified A levels as their current level of education, 7 had a first degree, 6 a higher degree, and 2 a doctorate. 7 participants were fluent in English, 11 were native speakers, and one self-rated as “intermediate”, but was considered fluent. We assumed that the lack of demographic diversity would have limited impact on our results, although future studies could make some provision for variations related to gender, age or educational background, e.g., cultural references. Participants did not have familiarity with our work on plausibility prior to taking part the study.

All participants were paid £10 and undertook the categorisation experiment. Only 12 were asked the open-ended questions and questions about the candidate factors. This allowed us to constrain the amount of data collected for the grounded theory analysis, while satisfying representativeness for the quantitative analysis. One participant was a very distinct outlier in terms of categorisation mean and variance, as they classified most statements into either “highly implausible” or “I don’t understand”. They were perfectly aware of the meaning but didn’t agree with the logic of the statement. Their think aloud data also suggested that they did not engage with the task as requested. We therefore excluded this participant from the analysis that follows.

Categorisation Results

Of the 532 judgements made, the most common were “highly implausible” (34%) and “highly plausible” (25%), followed closely by “slightly plausible” (23%). The least common responses were “slightly implausible” (11%) and

“don’t understand” (7%). In the analysis below, we sometimes interpret these ordinal responses (excluding “don’t understand”) as interval data from 0 (highly implausible) to 3 (highly plausible). Table 1 shows the response mean and variance for a selection of stimuli.

By Participant All participants used the entire range of responses. There were notable individual differences: 4 participants had median response of “highly implausible”, 8 had “slightly plausible”, with the remaining 7 medians falling in-between. The variance for each stimuli provides a measure of agreement between participants: the mean variance was 0.93 (min 0.21, max 1.49), suggesting quite a high level of disagreement. However, if we ignore the distinction between *highly* and *slightly* and merge categories to *plausible/implausible/don’t understand*, we actually see many participants agreeing with the modal (most popular) category: for 68% of stimuli, at least two-thirds agree. This shows at least a weak consensus was often present.

Participants used the “I don’t understand this statement” category a median of 1 times, indicating comprehension was not a problem for most participants. Only one participant claimed to understand all stimuli and, at the other extreme, two didn’t understand six stimuli. Using Spearman’s ρ , there is a medium negative correlation ($\rho_S(28) = -0.4, p = 0.09$) between not understanding and use of “Highly implausible” and a medium positive correlation ($\rho_S(28) = 0.35, p = 0.1$) between not understanding and that participant’s mean plausibility. This suggests there may be some confusion between “don’t understand” and “implausible”, which should be addressed in the design of future studies.

By Stimuli Almost all the stimuli provoked the full range of responses, confirming that assessing plausibility is a highly subjective task. The mean response for each stimuli

ranged from 0.17 (Stimuli 24) to 2.84. The variance ranged from 0.21 (Stimuli 26) to 1.49. We compared the plausibility ratings between the different stimulus groups described above. A Kruskal-Wallis one-way analysis of variance indicated that the plausibility ratings between the eight groups were significantly different $H(7) = 80, p < 1e - 13$. We then performed a series of Wilcoxon rank sum post-hoc tests with Bonferroni correction to determine which of the groups are significantly different. The p -values and group means are listed in Table 2. It shows, amongst others things, that statements from the categories “Musicals” and “Metaphor” were rated highest in plausibility ($\mu = 2.33, \mu = 1.76$), and differ significantly from the categories that were considered highly implausible, namely “Kafkaesque” and “Utopia/Dystopia” ($\mu = 0.56, \mu = 1.06$).

Think Aloud Results

To understand the factors which contributed to participants’ plausibility judgements, we performed a grounded theory analysis of the think aloud data. Grounded theory is a qualitative research method that is used to build, validate and expand theories from data, in order to reach “a theoretical formulation of the reality under investigation” (Corbin and Strauss 1990). Our analysis validated four of our initial hypothesised factors as influential within our participants’ judgements: *openness*, *familiarity*, *causality* and *feasibility*; the other two, *complexity* and *universality*, were concluded as non-influential. An additional factor, *perception of reality* was identified. Furthermore, for each of the supported factors, we identified a set of properties that represent the different ways the participants talked about the factors, as well as dimensions describing values these properties can hold. These results are summarised in Table 3.

Participants often based their judgement on how *Familiar* they were with the content; either from *experience* (own or by others) or *knowledge* they have acquired from different mediums. An illustrative example is Statement (1), quoted earlier to highlight the Complexity dimension. Two participants said the following:

“Doesn’t go with things in this time and day but people’s lives are complicated [...]”

“you see similar situations in the news [...] these are different personalities that actually exists [...]”

Consequently, this statement was often classified in the plausible spectrum (10 as highly plausible and 8 as slightly plausible). Familiarity at the level of *cultural recognition* also affects plausibility judgements. For instance, despite the fact that the statement: *What if a zombie rugby-tackled a ghost and broke his leg?*, contains fictional characters, as these are well-known concepts that form part of our culture, participants would hesitate about their plausibility value (even if eventually most decided the statement was not plausible).

Openness was also a recurrent factor we identified from the recorded sessions. Often, participants would try to make sense of the statements, saying things such as:

‘Maybe because my brain [is trying to give] sense to sentences.’

‘Where there is more room for interpretation, it is more easy to be black or white.’

Ambiguity and *context* played an important role for this factor. We found that key concepts appearing in a statement made a significant difference in plausibility judgements when these could be interpreted in different ways, and there was not enough context to narrow down the intended meaning. This led participants to stick with their favourite interpretation and provide their judgement accordingly. To illustrate, regarding the statement ‘*What if there was an old car that could be used as the space for holding a star?*’, participants would often ask if the concept *star* meant the astrological object or a celebrity, with most of the participants selecting the former and consequently placing this statement within the implausible spectrum. A similar reasoning was common with statement (4) above, for which participants would consider the concept of the *little doctor* as being either a child or a doctor short in height. Most participants chose the former interpretation and placed the statement in the plausible spectrum.

Feasibility was also one of the factors used by the participants when judging plausibility. In particular, we found that they would consider if the *likelihood* of the statement would form a usual or unusual scenario to decide on its plausibility. This was often seen in statements like (1), where the co-occurrence of all the elements of the statement was seen as unusual – but still plausible. One participant said:

‘it’s quite a complicated story but elements of the story makes it feel more real.’

Additionally, feasibility was also accounted for based on the use of *stereotypes* and how the individual parts of the statement fit together with a stereotypical construct. To illustrate, take the statement: *What if the world suddenly had lots more dictators? Then there would be less neediness, since dictators abuse the victims that demonstrate neediness*, for which a handful of participants focused on the contradiction between the concept of *dictators*, which has negative connotations, and the concept of *less neediness*, which has positive connotations.

Specific keywords, in particular *attributes* of the concepts in the statement, were also a decisive property when judging the statement based on its feasibility. For instance, the use of the adjective *little* in statement (4) made the plausibility higher, since participants interpreted the scenario as a *child playing doctor who is not able to actually take a pulse*, which in their view was completely feasible.

We also found that, although *causality* was not a strong factor in the decision making process, it was present on some occasions. In particular, finding *arguments* in favour or against particular elements of a statement had an influence in plausibility judgements. For instance, the statement: *What if the world suddenly had lots more assassins? Then there would be more antidotes, since assassins use the poisons that require antidotes*, links the concept of *assassins* with the concept of *poisons*, and this itself to the concept of *antidotes*. Although this statement was built through well-attested associations, specifically that assassins use poisons, and that poisons require antidotes, the intended strong link between assassins and poisons was used constantly as an argument against the plausibility of the statement. In contrast, from the statement: *What if there was a punishable man who had to learn how to eat a person in order to achieve his dream of becoming a criminal?*, the link between ‘eating

Category	Literary Fiction	Fiction in Fiction	Disney	Metaphors	Utopia / Dystopia	Alternative	Kafkaesque	Musicals
Mean Plausibility	1.71	1.11	1.37	1.76	1.06	1.12	0.56	2.33
Fiction in Fiction	0.11394	-	-	-	-	-	-	-
Disney	1.00000	1.00000	-	-	-	-	-	-
Metaphors	1.00000	0.16271	1.00000	-	-	-	-	-
Utopia / Dystopia	0.07103*	1.00000	1.00000	0.08518*	-	-	-	-
Alternative	0.14113	1.00000	1.00000	0.19369	1.00000	-	-	-
Kafkaesque	5.9e-07**	0.48768	0.00455**	2.4e-06**	0.31448	0.44275	-	-
Musicals	0.09299*	3.5e-05**	0.00039**	0.23095	5.9e-06**	4.4e-05**	1.9e-11**	-

Table 2: p -values from pairwise comparisons of stimuli groups using Wilcoxon rank sum test and Bonferroni correction. Significance: * low ($\alpha < 0.1$) and ** high ($\alpha < 0.01$). The second row comprises plausibility means for all categories.

a person' and becoming a 'criminal' was seen as logically connected:

'I can imagine eating a person as an act of initiation for a person to be part of a gang...'

Interestingly, the idea of *unknowns* was also used as an argument to decide on a plausibility category. This is when a participant considered that he/she did not have enough knowledge to argue against or in favour of a particular scenario. An example of this was the statement: *What if the ministry of magic paid JK Rowling to write her books so we muggles would think magic is fiction?*:

'I don't know if there is a minister of magic [...] who knows?'

which some participants used as an argument to assign a higher plausibility value.

Lastly, *perception of reality* played a role for some individuals when making their judgements. This factor represents how people may account for different ways of perceiving reality within certain scenarios. To illustrate, a participant categorised statement (1) as highly plausible based on the following reasoning:

"From my experience of reality, that might happen in some psychedelic state, a dream state, an imaginary state. I don't think it's right to count what happens in these states as any less real [...] the rabbit hole could be a doorway to other states. That's an idea I'm definitely open to."

another participant questioned the meaning of reality:

'[...] what is reality? Different statements push different readings of what reality is: objective reality in terms of things that are physically possible for ever and ever, things that might be possible in

Factor	Properties	Dimensions
Familiarity	Experience Knowledge Cultural recognition	Own/Others Cultural/Heard/Read/Seen Conceptual/Factual
Openness	Ambiguity Context	Most/Least plausible Lack/Presence of
Feasibility	Likelihood Stereotypes Attributes	Usual/Unusual Confirmation/Contradiction Opened/Specific
Causality	Arguments Context	In favour/Against/Unknowns Lack/Presence of
Perception of reality	Abstraction Cultural influence	Conceptual/Physical Background/Beliefs

Table 3: Influential factors when judging plausibility.

the future with technology, things that kind of work in a fictional world, and things that don't work at all.'

This is a subjective factor, but the intuition behind it is that judgements of plausibility are affected by personal views of what can be considered to be real or not.

Within this factor, *abstraction* was found to be a common property. In this case, the overall scenario was considered as having a hidden meaning. To illustrate, the statement: *What if the world suddenly had lots more angels? Then there would be more barriers, since angels serve the gods that impose the laws that create barriers*, was abstracted by some participants:

'I don't believe in angels or God, but I think the government can use it as a tool to manage people.'

leading them to assign a higher plausibility value to the statement. Similarly, *cultural influence* played a role in how participants' perception of reality would affect plausibility. Take the statement: *What if respected senators were to retire from their senates, join gangs and become shady gangsters?*, which was implausible for many participants because it did not make sense with their notion of reality:

'there is no reason why a senator with power and money would choose to be a gangster'

while for others, this was a plausible scenario due to their cultural background, where this situation was feasible:

'[...] in certain very corrupt countries it actually happens [...] when they are senators they belong to gangs, legal ones, but they do [...] and when they retire they keep being part of those clubs'

Likewise, the statement *What if a janitor needed to suppress a rebellion in order to gain admiration?* was classified as slightly plausible because:

'this is kind of a standard Hollywood plot really, I can imagine that being played by Tom Cruise [...] it has high degree of grounding in my experience of reality [...] not my experience of reality, my experience of Hollywood film making'

suggesting the participant considered the fictional world of Hollywood films as a type of reality.

Complexity, as mentioned before, did not come across as an influential factor for plausibility. For instance, regarding the complexity of statement (1), a participant highlighted:

'All happening at once is unlikely but it's possible [...] that doesn't change the plausibility.'

instead, this factor was considered to sometimes make the statements more difficult to understand. *Universality*, on the other hand, was seen as a factor that would make a statement more interesting, but would not have a significant effect on its plausibility value, specially when the statement was placed in the implausible spectrum:

‘if you are in the implausible categories, then it doesn’t matter, one, many [...] we are talking about something that is not real, so it doesn’t matter’

Multidimensional Scaling Results

We performed an multidimensional scaling (MDS) analysis to quantitatively derive a set of factors from the categorisation results, and to assess their influence on the overall judgement. Classic MDS maps measurements of (dis-) similarity among pairs of stimuli to distances between points in a geometric space (Borg and Groenen 2005, p. 3). In this space, each dimension can be considered a factor which influenced the initial similarity judgement. The meaning of a dimension is a matter of interpretation, based on the distribution of stimuli along it and their properties.

First, we had to determine the pairwise similarities between stimuli. The effort of collecting this data manually increases exponentially with the number of stimuli; we therefore followed a different approach suggested by Wallraven et al. (2009), where pairwise similarities are derived from a categorisation task. This approach allowed us to re-use our previously collected data, while implicitly grounding the similarities in plausibility judgements. We started with an empty similarity matrix, and increased the similarity value of two stimuli if they were put into the same plausibility group. This was repeated for all participants and normalised.

We then determined how many dimensions have to be used to approximate the data well enough by looking at how much variance in the data each dimension accounts for (Borg and Groenen 2005, pp. 247). We cut off at three dimensions, accounting for 78% of the variance, with the first dimension covering 57%. We then visualised each of these as a one-dimensional axis with the stimuli projected along it, allowing us to compare the relative distribution of the stimuli. These visualisations were given to four of the authors for interpretation, informed by the think-aloud results. A consensus interpretation of each dimension was then agreed on. These are summarised in Table 4, along with some examples of high and low scoring stimuli from Table 1.

On the first dimension, statements that showed a strong deviation from reality were grouped in one extreme. On the other extreme were statements that were more aligned with the rules of what it is commonly agreed as possible. The dimension was identified as *feasibility*. The second dimension was strongly associated with interpretability, i.e. with the stimuli’s *openness to interpretation*. Interestingly, the stimuli in both extremes were found to have different interpretations; however, what separated one group from the other was how ambiguous the possible interpretations were assessed to be. In one extreme, an interpretation would allow for a more decisive judgement, while in the other, the interpretation would still be seen as not convincing. The third dimension was found to classify stimuli based on *familiarity*. Well known elements, similar stories, common characters and stereotypes were identified in one extreme, while the other extreme presented the same characteristics (i.e. familiar elements) used in contradictory ways.

Dim	Var	Example stimuli (Id)		Interpretation
		High	Low	
1	57%	14, 26, 28	8, 12, 24	Feasibility
2	12%	5	28	Openness
3	9%	14, 26	19, 28	Familiarity

Table 4: The first three dimensions identified by MDS.

Future Work

This study provides evidence for three factors — feasibility, openness to interpretation, and familiarity — that contribute to judgements about the plausibility of fictional ideas. Understanding these factors is a necessary step towards further experimental investigation in this area. We plan to further test and refine this theory, and use it to design studies on the perception of machine-generated fictions.

We intend to model these factors computationally within the What-If Machine, to control the plausibility of the fictional ideas it generates. This could enhance the usefulness of the software and perhaps increase the cultural value of the ideas it produces. Although further experimentation is needed, we believe that metrics which predict values for each of the factors can be devised. Moreover, these could be used to predict the plausibility judgement that certain types of people will make for particular fictional statements.

A heuristic approach to analyse whether a statement is open to interpretation can be based on the *concreteness scores* of its constituent keywords. This measures the level of ambiguity of these words and give an approximation of the concreteness of the overall scenario. We have formalised fictional statements within the WHIM project as short narratives composed of narrative points that are either linked through causal relations, assumed by the reader, or given by the knowledge base (Llano et al. 2016). This formalism allows us to represent each statement as a graph over which we can reason. For instance, analysing the connectivity within the graph may allow us to hypothesise the level of *contextual support* within the statement as a whole. Highly supported statements may be less open to interpretation.

Feasibility, on the other hand, could be accounted for through the use of techniques such as a distributional semantics vector space model (Mikolov et al. 2015). Specifically, how well the elements of a statement fit together could be measured by studying their *semantic similarity* as well as their shared contextual co-occurrences. Stereotypical properties of concepts can be mined from the web (Veale 2012). A similar method could be followed in order to assess stereotypes within a statement and compare the polarity between the stereotypes and the other elements in the statement.

Finally, although familiarity is a subjective factor, metrics could be defined by establishing links with the information in knowledge bases of common knowledge and narrative constructs. In this context, strongly linked data can be seen as connected to “known or familiar scenarios”.

Progress in fictional ideation has general implications for Computational Creativity. In the problem solving paradigm of AI, intelligent tasks to automate are broken down into a series of problems to be solved, and there is a usually a

'right answer' to these problems whether local or global, or at least a fitness function relating to the potential value of solutions, which ordinarily captures notions of value from the real world. In the artefact generation paradigm of AI, however, an intelligence task to automate is considered as an invitation to create something of value in a potentially interesting way. Value can be externally imposed, but some Computational Creativity projects have allowed software to invent its own measures of value and to motivate these through framing (Charnley, Pease, and Colton 2012).

We can use the above observations on plausibility to set ourselves apart somewhat further from mainstream AI. In particular, Computational Creativity research could be seen as the sub-field focused on AI for *what could be* rather than *what is best*. Approaches to generate information about possible worlds naturally includes making discoveries about the real universe around us. However, it also includes the invention of imagined scenarios specifically constructed to reflect alternative realities. Such scenarios, like those produced by The What-If Machine, are valuable not because of their explicit reflection of reality, but because they force us to see the realities of our own existence in new and thought-provoking ways (in addition to simply providing entertainment). Analyses building on the work presented here could be influential in the advancement of the automatic generation of fictional universes and other creative works. Predicting how plausible (or not) people judge an idea to be will be a key part of automatically producing imagined scenarios.

Conclusions

We conducted a study in which participants categorised human authored and machine generated fictional statements (including the one paraphrased in the title of this paper), in terms of their plausibility. Unlike previous studies, in which the notion of plausibility had only operational significance, we explored the constituent factors of plausibility, in order to determine which are most influential. We found that the three most influential factors when judging plausibility are: *feasibility*, which determines how well the elements of an idea fit within the overall scenario, *openness* to subjective interpretation, and *familiarity*, which specifies the level of awareness of the overall scenario relative to known ideas. Our findings can serve as a theoretical grounding for future cognitive and computational studies involving plausibility, as well as informing wider discussions about perceptions of fictionality. We hope to build on this work to improve the cultural value of machine-generated fictions and to make fictional ideation a central part of Computational Creativity.

Acknowledgments

This research was supported by the European Commission via the WHIM (FP7 grant 611560), the EPSRC IGGI CDT (EP/L015846/1) and by EPSRC Leadership Fellowship grant EP/J004049/2.

References

Adams, A.; Lunt, P.; and Cairns, P. 2008. A qualitative approach to HCI research. In Cairns, P., and Cox, A., eds.,

Research Methods for Human-Computer Interaction. Cambridge University Press. chapter 7.

Borg, I., and Groenen, P. J. F. 2005. *Modern multidimensional scaling: Theory and applications*. Springer.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proc. 3rd International Conference on Computational Creativity*, 77–81.

Connell, L., and Keane, M. T. 2004. What plausibly affects plausibility? concept coherence and distributional word coherence as factors influencing plausibility judgments. *Memory & Cognition* 32:185–197.

Corbin, J. M., and Strauss, A. 1990. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology* 13(1):3–21.

Dick, P. K. 1995. How to build a universe that doesn't fall apart two days later. In Sutin, L., ed., *The Shifting Realities of Philip K. Dick: Selected Literary and Philosophical Writings*. Vintage.

Eco, U. 2006. Weak Thought and the Limits of Interpretation. In Zabala, S., ed., *Weakening Philosophy: Essays in Honour of Gianni Vattimo*. McGill-Queen's U. Press.

Eco, U. 2009. On the ontology of fictional characters. *Sign Systems Studies* 37(1/2):82–97.

Gow, J.; Cairns, P.; Colton, S.; Miller, P.; and Baumgarten, R. 2010. Capturing player experience with post-game commentaries. In *Proc. 3rd Int. Conf. on Computer Games, Multimedia & Allied Technologies*.

Kockelman, P. 2012. The ground, the ground, the ground: Or, why archeology is so 'hard'. *The Yearbook of Comparative Literature* 58:176–184.

Llano, M. T.; Colton, S.; Hepworth, R.; and Gow, J. 2016. Automated fictional ideation via knowledge base manipulation. *Cognitive Computation* 1–22.

Lombardi, D.; Nussbaum, E. M.; and Sinatra, G. M. 2015. Plausibility judgments in conceptual change and epistemic cognition. *Educational Psychologist* 1–22.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2015. Efficient estimation of word representations in vector space. arXiv:1301.3781v3 [cs.CL].

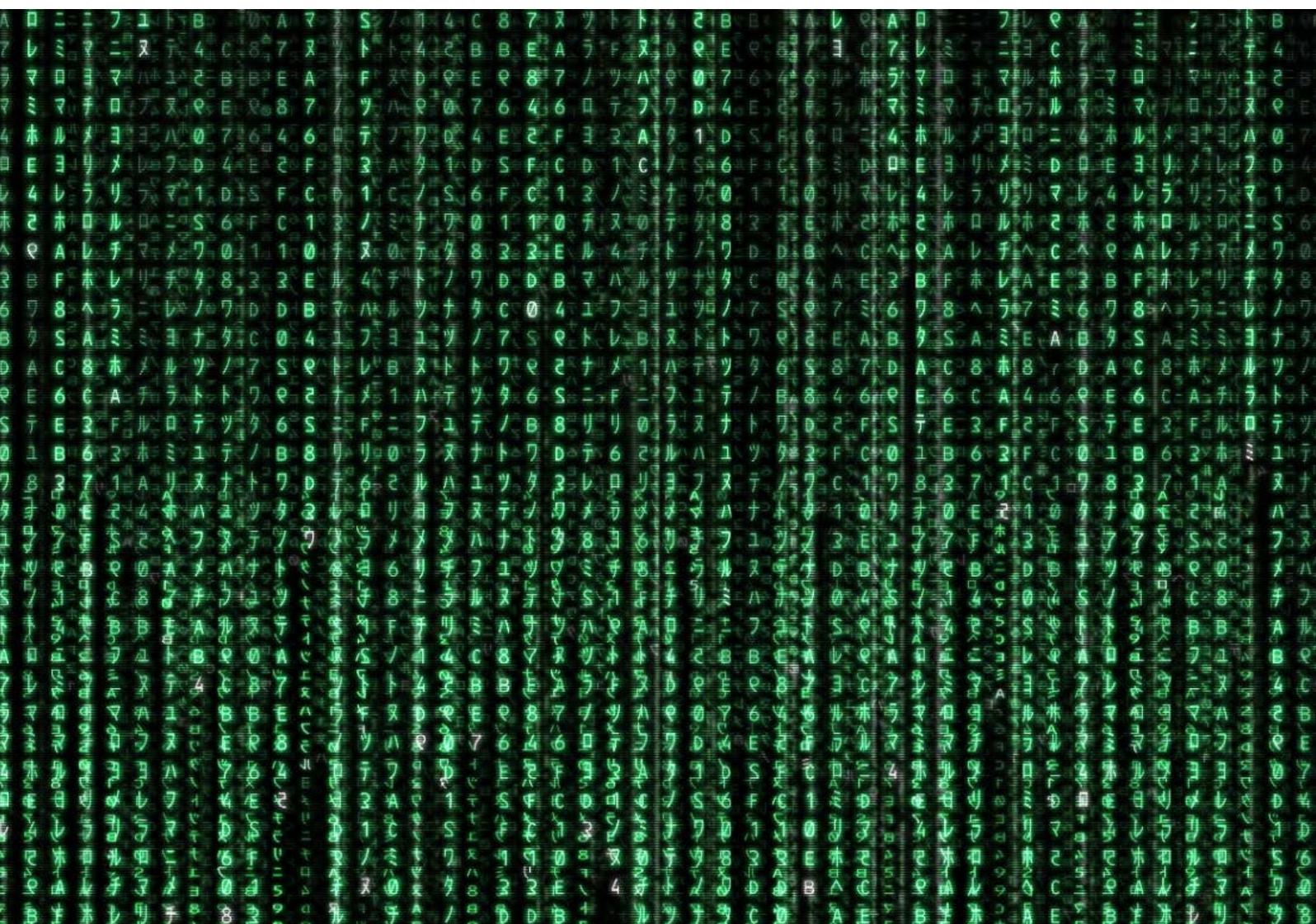
Rescher, N. 1976. *Plausible reasoning: An introduction to the theory and practice of plausibilistic inference*. K. Van Gorcum & Co.

Rothbauer, P. M. 2008. Triangulation. In Given, L. M., ed., *Encyclopedia of Qualitative Research Methods*. SAGE.

Veale, T. 2012. A context-sensitive, multi-faceted model of lexico-conceptual affect. In *Proc. 50th Annual Meeting of the Association for Computational Linguistics, Vol. 2: Short Papers*, 75–79. The Association for Computer Linguistics.

Wallraven, C.; Fleming, R.; Cunningham, D.; Rigau, J.; Feixas, M.; and Sbert, M. 2009. Categorizing art: Comparing humans and computers. *Computers & Graphics* 33(4):484–495.

LANGUAGE AND TEXT



Exploring the Role of Word Associations in the Construction of Rhetorical Figures

Paloma Galván¹, Virginia Francisco¹, Raquel Hervás¹, Gonzalo Méndez², Pablo Gervás²

¹Departamento de Ingeniería del Software e Inteligencia Artificial

²Instituto de Tecnología del Conocimiento

Universidad Complutense de Madrid, Spain

palomagalvan@ucm.es, {virginia, raquelhb, gmendez}@fdi.ucm.es, pgervas@sip.ucm.es

Abstract

Figurative language is a fundamental characteristic of elaborate forms of linguistic communication. We currently have very poor models of how figurative language may be constructed in computational terms. The overall aim is to identify possible regularities, intuitions or heuristics that may at a later stage be employed to drive a text generator that is capable of using this type of rhetorical figure.

Introduction

The use of figurative language is a fundamental tool in linguistic communication. One of the most easily identifiable characteristics of computer generated text is the tendency to stick to literal meanings. This is partly because literal meanings are unambiguous and have less risk of misinterpretation. But it is also in part due to the fact that we currently have very poor models of how figurative language may be constructed in computational terms. This paper explores the relationship between word associations as modelled in already available computational resources and the type of rhetorical figures that people employ regularly. The aim is to identify possible regularities, intuitions or heuristics that may at a later stage be employed to drive a text generator that is capable of using this type of rhetorical figure.

We consider three types of rhetorical figures or tropes. A *metaphor* is a widely-used literary mechanism which allows comparison between two disparate concepts. Metaphors transfer the qualities of one word to another, as in *Booger was a lion in the electoral arena*. Here, the qualities of *lion* (the source) are transferred to Booger (the target). A *simile* is a pointed, direct and explicit metaphor where two different things are compared to evolve a new meaning. A simile denotes the target to be like the source, and as such the target cannot totally be substituted by the source. A simile is a kind of metaphor where the comparison is made using the words “as” or “like”. For example, *Booger was like a lion*. An *analogy* links two disparate concepts by common properties, as in *Booger was as brave as a lion*. Here, the quality of being *brave* (the property) is used to link *lion* (the source) to *Booger* (the target).

Metaphors play an important role in communication, occurring as often as every third sentence (Shutova et al.

2012), so the generation of metaphors is essential for Natural Language Generation. The same occurs for analogies and similes.

Black (1955) made explicit that metaphors depend upon conceptual connections between networks of concepts. Inherent in this approach is the idea that metaphors are a matter of cross-domain mapping (Lakoff 1993). A metaphor is a cognitive process that builds or maps connections between networks of concepts as it occurs with similes and analogies. In consequence, to generate metaphors a conceptual structure is needed where every concept is placed not only taking into account its conventional usage but its diverse and unconventional usages (Veale 2014b). The best place to find this complex structure is the Web and that is where we are going to look for word associations in order to create our analogies, similes and metaphors.

This paper presents a new approach to finding word associations in the web using Thesaurus Rex (Veale and Li 2013). Then the potential of this system will be studied for the automatic generation of analogies, similes and metaphors.

This paper is organized as follows. The second section presents prior work on analogy, simile and metaphor generation. The third section explains our approach to finding word associations. In the fourth section the evaluation of our approach for rhetorical figures generation is presented. And finally, in the last section conclusions and future work are explained.

Related Work

Rhetorical figures have been the target of researchers in computational approaches to linguistics on and off for many years. However, only in recent years has the combination of available knowledge resources and accumulated insights allowed for the field to flourish. Metaphors have been widely studied in Natural Language Analysis but not so much in Natural Language Generation (NLG). There is a lot of work related to metaphor detection (Wilks et al. 2013), identification (Shutova, Sun, and Korhonen 2010), extraction and annotation (Wallington et al. 2003) but few related to metaphor generation. The reason can be that metaphor generation is as challenging as human creativity will allow. In this section the most important approaches for simile and metaphor generation are presented.

Approaches to Rhetorical Figures in NLG

In the field of natural language generation there have been a number of attempts to establish procedures for constructing rhetorical figures as important ingredients of generated spans of text. This has been attempted both in general terms (Hervás et al. 2006b) for different types of rhetorical figures, and for specific cases like analogies (Hervás et al. 2006a) or metaphors (Hervás et al. 2007). These attempts were all carried out before adequate sources of machine-readable knowledge were available and consequently suffered from a thirst of appropriate knowledge. The attempts considered the problem of rhetorical figure employment in text generation in general theoretical terms but lacked sufficient volume of explicit knowledge on the underlying semantics of words to be capable of practical generation.

Approaches to Conceptual Construction of Rhetorical Figures

The recent development of sources of knowledge that allow easy mining of large corpora of text for significant word associations has led to the emergence of a number of systems that rely on these for constructing rhetorical figures of different types.

Jigsaw Bard Jigsaw Bard (Veale and Hao 2011) is a web service that exploits linguistic readymades to generate similes on demand. Jigsaw Bard scans Google n-grams to index potential readymades which are then re-purposed as a simile. For example, given the adjectival property *quiet* Jigsaw Bard returns the simile “*The peaceful life of a monastery*”. The Jigsaw Bard is best understood as a creative thesaurus: for any given property (or blend of properties) selected by the user, the Bard presents a range of apt similes, and users must decide which similes are most suited to their descriptive purposes.

Thesaurus Rex Thesaurus Rex (Veale and Li 2013) is a web service that given two concepts (for example, War and Divorce) returns a phase cloud of the nuanced categories that are shared by both concepts (in the given example it returns a cloud that contains *traumatic-event*, *stressful-event*, *unexpected-event*...).

Thesaurus Rex organizes concepts according to categories they are placed into by speakers in everyday language (*food*, *drink*, *beverage*...). These categories have an associated weight that represents their relative importance for the given concept. Thesaurus Rex can show different categories for each concept and allows in turn to consult the concepts in each category. For example for the concept *coffee*, some of its categories with more weight are *beverage* or *drink* and some with less weight are *leaf* or *apposition*. Concepts in Thesaurus Rex have associated properties or modifiers which are accompanied by a non-standard weight indicating how strong its relation to the concept is. For example, for *coffee* some of the modifiers with more weight are *hot*, *acidic* or *stimulating*, and modifiers with less weight are *smaller* or *adult*.

Metaphor Magnet Metaphor Magnet (Veale and Li 2012) is a Web service that allows users to enter queries with sin-

gle terms (such as *leader*), compound terms with an affective spin (such as *good leader* or *+leader*), or copula statements (such as “*Steve Jobs is a +leader*”). For each input, the service marries its extensive knowledge of lexicalized stereotypes to the grand scale of the Google n-grams to generate the most appropriate affective elaborations and interpretations. In each case, Metaphor Magnet provides an explanation of its outputs. If *Steve Jobs* were to be viewed as a *master*, the properties *skilled*, *enlightened*, *free* and *demanding* are all highlighted as being most appropriate. Metaphor Magnet sees metaphor interpretation as a question of which properties are mapped from the source to the target.

Metaphor Magnet lacks a proposition level view of the world, in which stereotypes are linked to other stereotypes by arbitrary relations.

Metaphor Eyes Metaphor Eyes (Veale 2014a) employs a propositional model of the world that reasons with subject-relation-object triples rather than subject-attribute pairs (as Metaphor Magnet does). Metaphor Eyes acquires its world-model from a variety of sources and it views metaphor as a representational lever, allowing it to fill the holes in its weak understanding of one concept by importing relevant knowledge from a neighboring concept.

Metaphor Eyes metaphorize one concept (the source) as other concept (the target). Given *Scientist* and *Artist* it generates metaphors as “*Scientists develop ideas like artists*”.

Figure8 Figure8 (Harmon 2015) is a system that contains an underlying model for what defines creative and figurative comparisons, and evaluates its own output based on these rules. The system is provided with a model of the current world and an entity in the world to be described. A suitable vehicle is selected from the knowledge base, and the comparison between the two nouns is clarified by obtaining an understanding via corpora search of what these nouns can do and how they can be described. Sentence completion occurs by intelligent adaptation of a case library of valid grammar constructions. Finally, the comparison is ranked by the system based on semantic, prosodic, and knowledge-based qualities.

Word Association Generation

This section presents the proposed approach for the generation of word associations, which has been implemented as a web service. This service receives a common noun as an input, which is the target concept for the word association. Following the steps described in the Process section below, the system generates source concepts with similar properties to the target concept creating word associations.

Entry

The proposed approach receives a common noun as an input, which is the target concept for which the word association must be generated. Using Thesaurus Rex, the system unfolds a comparison between the target concept and another concept that acts as the source of the rhetorical figure, with similar properties to the target concept in order to create a word association.

Table 1: Examples of word associations obtained. Words in bold represent the choices made for each example.

Step	Target	snow	thunder	network
1	Categories	surface, elements, weather...	noise, sound, event...	system, structure, entity...
2	Modifiers soft, white...	natural, reflective, slippery , sudden, weather...	natural, loud , adaptive, physical...	social , complex,
3	Categories for the selected modifier	surface , ground, stuff...	instrument , thing...	institution , event activity, science...
4	New query	slippery surface	loud instrument	social institution
5	Obtained concepts	satin, silk, nylon, polyester... saxophone...	trumpet, drum, horn, religion...	family, government,

Process

Table 1 shows a few examples of target concepts and how Thesaurus Rex is used to obtain words associated to the target concepts. Taking the first concept, *snow*, as an example, the detailed process is the following:

- 1. Target concept categories.** To obtain the filtered categories to which the target concept belongs, we first extract a list of all the general categories of the concept using a Thesaurus Rex query. From this list, only the $N\%$ of categories with the highest weights are considered as candidates. The value of N is configurable (in this example, $N = 0.4$). If a high N value is set, we will have in the list categories with lower weights, which are less relevant to the target concept. In the same way, we can set N to a low value, facing the risk of shortening the list to a single element. In the *snow* example, the categories with higher weights in Thesaurus Rex are *surface* and *weather*.
- 2. Modifier extraction.** In addition to the categories, we also need a list of modifiers associated to the target concept, which is returned by a new query to Thesaurus Rex. From this list, the $N\%$ of attributes with the highest weights are considered as candidates (in this example, $N = 0.6$). For example, if our target concept is the noun *snow*, some of the most important properties extracted are: *natural*, *reflective*, *slippery*, *soft* and *white*.
Modifier selection. One of the modifiers previously obtained is randomly selected. This random selection makes the system less repetitive, as the words associated to the same target concept are not always the same as if only the modifier with the highest weight were selected. For the current example, we suppose that the system has chosen the modifier *slippery*.
- 3. Categories selection.** Using the modifier chosen in the previous step, a new query to Thesaurus Rex is performed in order to obtain categories that present this modifier as a highlighted property. In the *snow* example, the categories selected could be *surface*, *ground* and *stuff* which are categories that present the *slippery* property in Thesaurus Rex.
Category selection. One of the categories obtained in the previous step is selected. The system could be parametrized to select a category which contains the target concept (a category that matches one obtained in step 1). It could also be parametrized to choose a category in which

the target concept is not included (discarding categories that match those obtained in step 1). For the current example, *surface* is supposed to be the selected category.

- 4. New query composition.** A new query for Thesaurus Rex is then composed by using the category obtained in the previous step and the modifier selected in step 3. In the current example, we will assume this new query is *slippery surface*.
- 5. Final concept selection.** With the query composed in the previous step, we obtain a list of concepts that belong to the category selected in step 5 (*surface*) and at the same time present the property selected in step 3 (*slippery*). This list is usually quite extensive, so the system randomly chooses among the results that have an associated weight among the $N\%$ of concepts with the highest weights (in this example, $N = 0.1$). In our example, the final concepts associated to the target concept are *satin*, *silk*, *nylon* or *polyester*

Output

The system output is the source concept that gives rise to the rhetorical figure, related through a shared property with the original target concept provided by the user. The shared property is significant in both concepts, which means that the property has a high weight for both of them. The resulting source concept is randomly chosen from the list of generated concepts, and is subsequently used to create a rhetorical figure.

Evaluation

The aim of this evaluation has been twofold. On the one hand, we intended to test the appropriateness of the analogies, similes and metaphors generated by our system, in order for us to be able to refine the process followed to generate them. On the other hand, we also expected to find out what kind of rhetorical figure is more enlightening for the evaluators and which one is closer to a rhetorical figure generated by humans.

Rhetorical Figures Generation using Word Associations

This approach uses the simplest and purest copula form for analogies, similes and metaphors:

- Analogy: *TARGET is as PROP as SOURCE.*

- Simile: *TARGET is like SOURCE*.
- Metaphor: *TARGET is SOURCE*.

Design of the Evaluation

The evaluation set was composed by 36 analogies, 36 similes and 36 metaphors. To create these elements, 36 different words were used as target concepts and one analogy, one simile and one metaphor were created for each of them. In order to avoid the possibility that one evaluator could evaluate several rhetorical figures related to the same target concept, the original data set was divided in three different subsets of 36 rhetorical figures. Each subset had 12 metaphors, 12 similes and 12 analogies, all of them created from a different target concept.

The evaluation was carried out as an online survey using Google Forms, where each evaluator received a link to one of the three surveys and was asked to score each of the figures using a Likert scale. Evaluators were asked to rate how appropriate or natural sounding each trope was, giving them a score from 1 to 7 (where 1 symbolizes a completely inappropriate trope and 7 represents a completely natural sounding trope). We chose to use the median and the mode because when working with the Likert scale, these are the most interesting metrics. Interpreting the average when managing categories such as "totally meaningful" or "totally meaningless", would not provide useful information. Adding the "totally meaningful" value (5) to two "meaningless" values (2) would result in an average of 4, but that is not a very rich interpretation. Traditional statisticians do not recommend using the average of the data in the Likert scale, which offers ordinal values.

In order to have two different baselines in our experiment to measure the quality of the figures generated by our system, we have used a set of commonly accepted rhetorical figures, together with a set of random manually generated ones, to compare them against the ones generated by our system.

The way in which the analogies, similes and metaphors were created was the following:

- Commonly accepted figures: 6 words (3 abstract and 3 concrete) were used as target concepts to obtain commonly accepted metaphors, similes and analogies:
 - TIME: Time is money / Time is like money / Time is as valuable as money
 - KNOWLEDGE: Knowledge is light / Knowledge is like light / Knowledge is as attractive as light
 - ARGUMENT: An argument is a war / An argument is like a war / An argument is as violent as a war
 - BALLERINA: A ballerina is a swan / A ballerina is like a swan / A ballerina is as graceful as a swan
 - STAR: A star is a diamond / A star is like a diamond / A star is as bright as a diamond
 - THUNDER: A thunder is a lion / A thunder is like a lion / A thunder is as mighty as a lion
- Randomly generated figures: 6 words (3 abstract and 3 concrete) were used as target concepts to obtain randomly generated metaphors, similes and analogies:

- HUNGER: Hunger is knowledge / Hunger is like knowledge / Hunger is as mechanical as knowledge
- SAILING: Sailing is boyhood / Sailing is like boyhood / Sailing is as allergenic as boyhood
- SYLLOGISM: A syllogism is a nation / A syllogism is like a nation / A syllogism is as ungulate as a nation
- ELEPHANT: An elephant is a napkin / An elephant is like a napkin / An elephant is as holy as a napkin
- CORKSCREW: A corkscrew is a stamp / A corkscrew is like a stamp / A corkscrew is as furry as a stamp
- TRAIN: A train is a violin / A train is like a violin / A train is as observational as a violin

- Automatically generated figures: 24 words (12 abstract and 12 concrete) were used as target concepts by our system to obtain metaphors, similes and analogies. Half of them were generated with the system configured to obtain the source concept from the same category as the target, and the other half to take the source concept from a different category.
 - Source and target from the same category:
 - * WEDDING: A wedding is a party / A wedding is like a party / A wedding is as private as a party
 - * WISH: A wish is a desire / A wish is like a desire / A wish is as mental as a desire
 - * LIFE: Life is politics / Life is like politics / Life is as complex as politics
 - * ANGEL: An angel is a fairy / An angel is like a fairy / An angel is as invisible as a fairy
 - * DEVIL: Devil is love / Devil is like love / Devil is as spiritual as love
 - * GOVERNMENT: Government is family / Government is like family / Government is as social as family
 - * SNOW: Snow is a carpet / Snow is like a carpet / Snow is as soft as a carpet
 - * NEEDLE: A needle is a knife / A needle is like a knife / A needle is as sharp as a knife
 - * COTTON: Cotton is cashmere / Cotton is like cashmere / Cotton is as natural as cashmere
 - * HONEY: Honey is sugar / Honey is like sugar / Honey is as sticky as sugar
 - * BATTLE: A battle is a war / A battle is like a war / A battle is as historical as a war
 - * WRITER: A writer is a designer / A writer is like a designer / A writer is as creative as a designer
 - Source and target from different categories:
 - * SAVING: Saving is farming / Saving is like farming / Saving is as productive as farming
 - * ACCIDENT: An accident is an electric shock / An accident is like an electric shock / An accident is as unexpected as an electric shock
 - * NETWORK: Network is family / Network is like family / Network is as social as family
 - * IDEA: Idea is colors / Idea is like colors / Idea is as abstract as colors

Table 2: Metaphor results.

Source	Mode			Median		
	Abstract	Concrete	Total	Abstract	Concrete	Total
Random	1	1	1	1	1	1
Commonly accepted	7	7	7	6	5	5
Generated (different category)	1	1	1	2	2	2
Generated (same category)	7	1	7	5	4	5
Generated	1	1	1	3	3	3

Table 3: Simile results.

Source	Mode			Median		
	Abstract	Concrete	Total	Abstract	Concrete	Total
Random	1	1	1	2	1	1
Commonly accepted	7	5	7	6	5	5
Generated (different category)	1	1	1	2	3	3
Generated (same category)	7	6	6	5	4	5
Generated	1	1	1	4	3	4

Table 4: Analogy results.

Source	Mode			Median		
	Abstract	Concrete	Total	Abstract	Concrete	Total
Random	1	1	1	1	1	1
Commonly accepted	7	7	7	6	6	6
Generated (different category)	1	7	2	3	4	4
Generated (same category)	5	7	7	4	4	4
Generated	1	7	7	4	4	4

Table 5: General results of the evaluation.

Source	Mode			Median		
	Abstract	Concrete	Total	Abstract	Concrete	Total
Random	1	1	1	1	1	1
Commonly accepted	7	7	7	6	5	6
Generated (different category)	1	1	1	2	3	3
Generated (same category)	7	7	7	5	4	5
Generated	1	1	1	4	4	4

- * ASSEMBLY: An assembly is an aircraft / An assembly is like an aircraft / An assembly is as complex as an aircraft
- * WINTER: Winter is salad / Winter is like salad / Winter is as cold as salad
- * MOON: The moon is an halogen lamp / The moon is like an halogen lamp / The moon is as bright as an halogen lamp
- * REFUGEE: A refugee is an elderly / A refugee is like an elderly / A refugee is as vulnerable as an elderly
- * TEMPLE: A temple is a school / A temple is like a school / A temple is as public as a school
- * ACID: Acid is a tiger / Acid is like a tiger / Acid is as dangerous as a tiger
- * BULLET: A bullet is a bolt / A bullet is like a bolt / A bullet is as metal as a bolt
- * DRAWER: A drawer is a chesnut / A drawer is like a

chesnut / A drawer is as dark as a chesnut

Results of the Evaluation

The evaluation was carried out by 72 evaluators, so that each of the 3 subsets of rhetorical figures was assessed by 24 different evaluators.

The evaluation results for the metaphors are shown in Table 2. Overall the results obtained from the evaluation were as expected, random tropes turned out to be the ones with lower ratings, with a median of 1, and tropes with higher ratings were commonly accepted ones, with a median of 6. Regardless of the type of rhetorical figure and whether they represent specific or abstract concepts, the median of these figures is 4 (3 for those of different categories and 5 for those belonging to the same category).

Interestingly, the modes are the same for abstract and concrete concepts tropes, regardless of how they were generated. The mode of both the random tropes and the tropes

generated by our system with different categories is 1, and the mode of the commonly accepted ones and the tropes generated in the same category also matches, with a value of 7.

If we take a closer look at the data subsets of the metaphors, similes and analogies, we can observe that the value of the medians of all figures generated randomly in the three data sets is 1. The results are more satisfactory for the commonly accepted figures, with median values between 5 and 6, proving that the evaluators did not take risks awarding the maximum score.

When we continue to analyze the subsets, we can see that the results obtained for the tropes belonging to different categories are less promising than those obtained for tropes with the same category, with variations between 2 and 4. In the case of the analogies, the median is the same for the ones generated in the same category or in different categories, with a value of 4. The difference between medians of random tropes and commonly accepted tropes fluctuates between 4 and 5.

The graphs show comparative results for the different ways of generating the rhetorical figures: using concrete and abstract concepts, as well as the combined results. The first graph (see Figure 1), corresponds to the word association using abstract concepts and we can observe that the random tropes results are 1 except in the case of the similes median, which is 2. Commonly accepted rhetorical figures mode is 6 and the median is 7. The mode of generated tropes of different categories is always 1 while the median results range between 2 and 3. The mode for generated tropes of same category is between 5 and 7, and the median is between 4 and 5.

Concrete concepts results are shown in Figure 2. Similarly to the abstract concepts, both the results of the mode and the median of random tropes are 1. The mode and the median of commonly accepted rhetorical figures range between 5 and 7. Generated tropes mode of different categories is 1, except for the analogies, which is 7. The median in this case is between 2 and 4. Generated rhetorical figures median of the same category is always 4, while mode is 1 for metaphors, 6 for similes and 7 for analogies.

In Figure 3 the total results for all the rhetorical figures can be seen. Clearly, the result of the randomly generated rhetorical figures is 1. Commonly accepted tropes mode is 7, while the median is 5 for metaphors and similes, and 6 for analogies. The mode of generated tropes of different categories is 1 and 2, and the median is 2 for metaphors, 3 for similes and 4 for analogies. Generated tropes of the same category mode is 7 for metaphors and analogies, while simile mode is 6. The median is 5 for metaphors and similes, and 4 for analogies.

We can conclude that, although the process we have used to generate the rhetorical figures works quite well when concepts of the same category are used, according to the opinions of the evaluators, something different happens in the case of using concepts that belong to different categories, which, in general, obtain worse results. This fact points to the need of using additional properties or relationships in order to obtain concepts that can subsequently give rise to more meaningful rhetorical figures.

Discussion

As we can see, in all cases the randomly generated metaphors are rated as meaningless by the evaluators. In contrast, commonly accepted metaphors get the highest results, with a slight preference for the metaphors created using abstract concepts over the ones that are based on the use of concrete concepts. The automatically generated metaphors using concepts of different categories are also poorly rated, which points out that sharing only one property is not enough to generate a good metaphor. For the generated metaphors using concepts that belong to the same category, the difference that exists between the modes of the metaphors that use concrete and abstract concepts is remarkable. This suggests that abstract metaphors are more evocative and offer a wider range of interpretations than concrete ones. Finally, the overall median for the metaphors also suggests that more aspects need to be taken into consideration to increase the perceived quality of these rhetorical figures.

Table 3 shows the results for the evaluated similes. The ratings in this case are quite similar to the results obtained for the metaphors.

The results of the evaluation of the analogies can be seen in Table 4. The ratings in this case are slightly higher than in the two previous tropes, probably due to the fact that the aspect in which the two concepts are considered to be similar is explicitly stated. This same aspect may be the cause for the lower score obtained by the automatically generated analogies using abstract concepts that belong to the same category. In this case, the similarity perceived by the evaluators may be focused on a different characteristic than the one chosen by the system, which causes the score to be lower than the one granted to the previous figures. On the contrary, the analogies generated using concrete concepts belonging to different categories are much better rated than in the previous tropes. In this case, the reason seems to be the fact that the property used by the system to compare both concepts has been made explicit, so the evaluators can see the reason why the system considers the two concepts related to each other and they are more inclined to accept it as valid.

Finally, the overall results of the evaluation can be seen in Table 5. Although they don't differ much from the results obtained for the different tropes independently, the values of the modes are clearly shifted towards the limits of the scale. This effect suggests that human evaluators tend to accept or not accept a rhetorical figure as valid, but intermediate positions are less common. As for the value of the medians, the condensed results confirm the perception that, in terms of automatically generated tropes, the ones that use abstract concepts that belong to the same category are slightly better appreciated than the rest.

Conclusions and Future Work

We have proved that it is possible to evaluate the quality of rhetorical figures and get consistent results. One of the clearest conclusions is that in our system concepts generate tropes of the same category with significantly higher quality than the tropes based on concepts of different categories.

In view of the results, one of the paths we have to follow

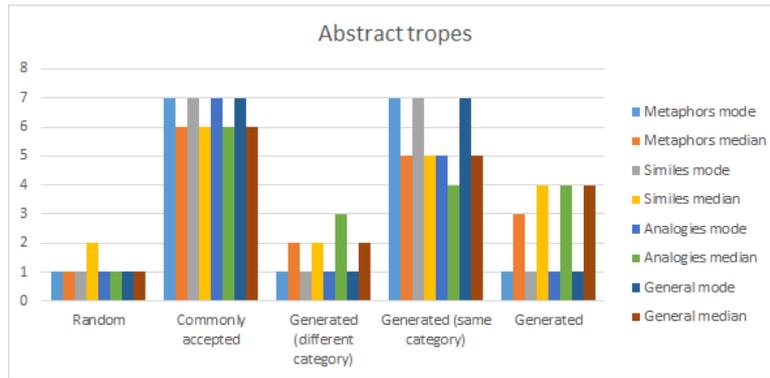


Figure 1: Abstract Tropes

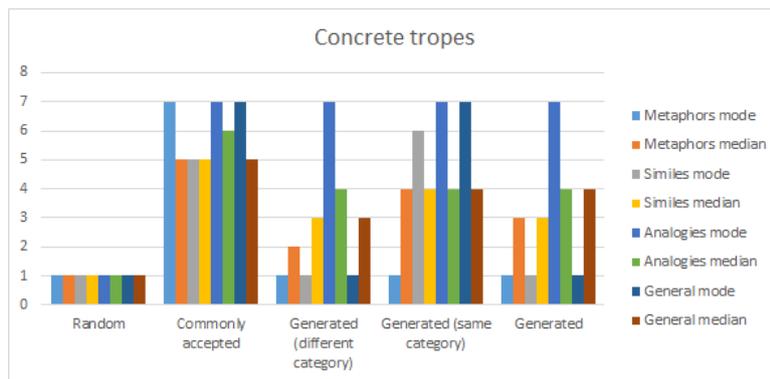


Figure 2: Concrete Tropes

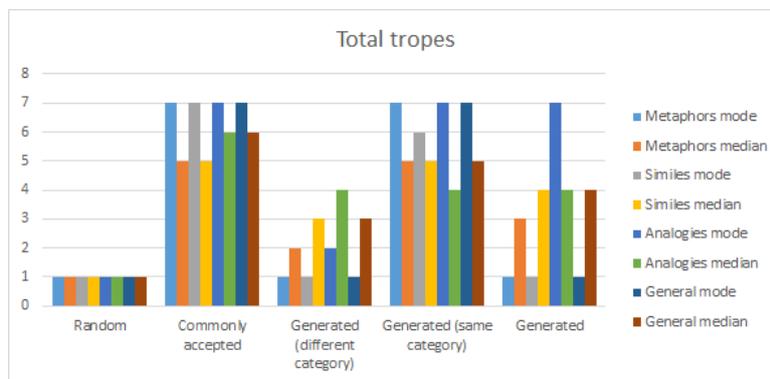


Figure 3: Total Tropes

is directed to find ways to generate good rhetorical figures from concepts of different categories, because in everyday life some of the best rhetorical figures are constructed from these kinds of terms, such as *time is money*. The categories with higher weights obtained for the concept *time* in Thesaurus Rex are *information*, *quantity* and *attribute*, while for the concept *money* they are *thing*, *property*, *value* and *assets*. As we have seen in the evaluation, this trope gets a good rating and we need more information about this type of rhetorical figures.

In order to generate appropriate figurative language depending on the content of a given text, it would be interesting to find sets of words grouped by topic. On the other hand, in order to adequate the figurative language to the goals of the reader, it would be helpful to have sets of concepts grouped by the complexity of their meaning.

Sometimes, constraints encountered arise from the web itself. This is because the information usually available on the web tends to be more literal than figurative. For the previous example, the attributes with more weight obtained from

Thesaurus Rex when searching for *time* are *physical, basic, measurable, relevant and abstract*. That suggests that it may be more appropriate for us to find or generate a specific knowledge resource that provide more evocative properties.

The highest mode for rhetorical figures generated by our system are obtained for analogies. In the case of the median of the total result tendencies are less clear. While the rhetorical figures in the same category produce better results in metaphors and similes, rhetorical figures with different category get better valuations in analogies.

In the future, we would like to continue doing assessments to find patterns or similarities among the best rated rhetorical figures, and we wish to test this with larger datasets. Thus the evaluation findings could serve to improve the quality of the resources generated by our system.

We have used the terms "concrete" and "abstract" when categorising input concepts. It would be interesting to check whether it makes a difference to use a concrete word to describe an abstract concept (e.g. "time is money") and vice-versa.

As future work we would also like to check the degree of similarity between the source and target concept. If the concepts are too similar, the resulting trope would be correct but not very practical.

With respect to the amount of information provided in the rhetorical figure, there are no significant differences between those that provide more or less information, because similar results are obtained for metaphors – in which only the original concept and the new concept are indicated – and analogies – in which the shared attribute is also shown.

The results obtained indicate that further attempts should be made to evolve our system and generate higher quality rhetorical figures, progressively evolving the quality of system results towards that of rhetorical figures generated by people. In the future, a useful feature that may improve our system is to relate the original concept with concepts that have more than one property in common. From now on another way that we should investigate is to generate rhetorical figures with concepts that are related through two or more attributes. In the example *A ballerina is a swan*, both concepts share properties as *pretty, graceful and stylized*.

Acknowledgements

This work is funded by ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

Black, M. 1955. XII. Metaphor. *Proceedings of the Aristotelian Society* 55(1):273–294.

Harmon, S. 2015. Figure8: A novel system for generating and evaluating figurative language. In *Proceedings of the Sixth International Conference on Computational Creativity*, 71–77.

Hervás, R.; Camara Pereira, F.; Gervás, P.; and Cardoso, A. 2006a. Cross-domain analogy in automated text generation.

In *3rd Joint Workshop on Computational Creativity, dentro de la 17th European Conference on Artificial Intelligence*, 43–48.

Hervás, R.; Camara Pereira, F.; Gervás, P.; and Cardoso, A. 2006b. A text generation system that uses simple rhetorical figures. *Procesamiento de Lenguaje Natural* 37:199–206.

Hervás, R.; Costa, R.; Costa, H.; Gervás, P.; and Camara Pereira, F. 2007. Enrichment of automatically generated texts using metaphor. In *6th Mexican International Conference on Artificial Intelligence (MICAI-07)*, volume 4827, 944–954. Aguascalientes, Mexico: Springer Verlag, LNAI Series.

Lakoff, G. 1993. The Contemporary Theory of Metaphor. *Metaphor and Thought* 202–251.

Shutova, E.; de Cruys, V.; T.; and Korhonen, A. 2012. Unsupervised Metaphor Paraphrasing using a Vector Space Model. In *Proceedings of the 24th International Conference on Computational Linguistics*, 1121–1130.

Shutova, E.; Sun, L.; and Korhonen, A. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 1002–1010. Beijing: Association for Computational Linguistics.

Veale, T., and Hao, Y. 2011. Exploiting Readymades in Linguistic Creativity: A System Demonstration of the Jigsaw Bard. In *ACL (System Demonstrations)*, 14–19.

Veale, T., and Li, G. 2012. Specifying Viewpoint and Information Need with Affective Metaphors: A System Demonstration of the Metaphor Magnet Web App/Service. In *Proceedings of the ACL 2012 System Demonstrations*, 7–12.

Veale, T., and Li, G. 2013. Creating Similarity: Lateral Thinking for Vertical Similarity Judgments. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 660–670.

Veale, T. 2014a. A Service-Oriented Architecture for Metaphor Processing. In *Proceedings of the Second Workshop on Metaphor in NLP*, 52–60.

Veale, T. 2014b. A service-oriented architecture for metaphor processing. In *Proceedings of the Second Workshop on Metaphor in NLP*, 52–60. Baltimore, MD: Association for Computational Linguistics.

Wallington, A.; Barnden, J.; Buchlovsky, P.; Fellows, L.; and Glasbey, S. 2003. Metaphor annotation: A systematic study. CSRP 03-04, School of Computer Science, The University of Birmingham, Birmingham.

Wilks, Y.; Galescu, L.; Allen, J.; and Dalton, A. 2013. Automatic metaphor detection using large-scale lexical resources and conventional metaphor extraction. In *Proceedings of the First Workshop on Metaphor in NLP*, 36–44. Atlanta, GA: Association for Computational Linguistics.

Meta4meaning: Automatic Metaphor Interpretation Using Corpus-Derived Word Associations

Ping Xiao¹
ping.xiao@cs.helsinki.fi

Khalid Alnajjar¹
alnajjar@cs.helsinki.fi

Mark Granroth-Wilding²
mark.granroth-wilding@cl.cam.ac.uk

Kat Agres³
kathleen.agres@qmul.ac.uk

Hannu Toivonen¹
hannu.toivonen@cs.helsinki.fi

¹Dept. of Computer Science and HIIT, University of Helsinki, Finland; ²Computer Laboratory, University of Cambridge, UK
³School of Electronic Engineering and Computer Science, Queen Mary University of London, UK

Abstract

We propose a novel metaphor interpretation method, *Meta4meaning*. It provides interpretations for nominal metaphors by generating a list of properties that the metaphor expresses. Meta4meaning uses word associations extracted from a corpus to retrieve an approximation to properties of concepts. Interpretations are then obtained as an aggregation or difference of the saliences of the properties to the tenor and the vehicle. We evaluate Meta4meaning using a set of human-annotated interpretations of 84 metaphors and compare with two existing methods for metaphor interpretation. Meta4meaning significantly outperforms the previous methods on this task.

Introduction

Metaphor has various linguistic manifestations, such as the metaphorical use of nouns, verbs, adjectives and adverbs, as well as at larger conceptual frames, for instance, an entire poem or story in a metaphor with something outside it. The present work focuses on interpreting nominal metaphors of the form ‘NOUN₁ is [a] NOUN₂’, where, following Richards (1936), NOUN₁ is called the *tenor* and NOUN₂ the *vehicle*.

The meaning of a metaphor is not fixed. It arises from the interaction between at least two conceptual spaces, the tenor’s and the vehicle’s (Black 1962), but often also the context’s (Ortony et al. 1978). Metaphors are different with respect to the number and the saliences of the individual interpretations, such as a few salient meanings, a few or many non-salient meanings, and no meaning. The meaning distribution of a metaphor is sensitive to context, which increases or decreases the saliences of certain meanings.

Metaphor meanings have been most often talked about in terms of *properties* – which properties of the tenor have been highlighted or newly attributed to it (Glucksberg 2001; Moreno 2004). A key objective of a metaphor interpretation program is to identify those *highlighted* properties.

Metaphor interpretation relies on knowledge about the tenor and vehicle. In this work, we propose *Meta4meaning*, a novel method for metaphor interpretation. Meta4meaning derives *word associations* from a large text corpus in order to obtain a concept’s properties and their saliences. To identify the properties highlighted by a metaphor, Meta4meaning

measures either the aggregation or the difference of the saliences of a property to the tenor and the vehicle, capturing distinct ways in which metaphor may function. Furthermore, we test two hypotheses regarding the saliences of properties involved in metaphor understanding, the salience imbalance hypothesis (Ortony 1979) and the requirement of pre-existing saliences of interpretations to the tenor and the vehicle.

Meta4meaning is evaluated against the interpretations of 84 metaphors acquired from human subjects by Roncero and Almeida (2014). The performance is also compared with two existing methods developed by Terai and Nakagawa (2008) and Veale and Li (2012). We find Meta4meaning using word associations to be the most successful method.

In the remainder of this paper, we first give a formal definition of the metaphor interpretation problem, and review the related work. The Meta4meaning method is described next in two parts: first, how it acquires concept properties, and second, how it uses the properties of the tenor and vehicle to provide metaphor interpretations. Then, we report an evaluation of the methods and discuss the results.

Problem Formalization

Consider a nominal metaphor of the form ‘NOUN₁ is [a] NOUN₂’, such as ‘alcohol is a crutch’. An interpretation of the metaphor is a property that the vehicle NOUN₂ (crutch) expresses about the tenor NOUN₁ (alcohol). In a study by Roncero and Almeida (2014), interpretations of ‘alcohol is crutch’ included properties such as ‘helpful’ and ‘addictive’.

Given a nominal metaphor, the objective of metaphor interpretation is to produce a ranked list of possible interpretations, such that highly ranked interpretations are likely to be considered interpretations by humans.

Related Work

Kintsch (2000) applied Latent Semantic Analysis (LSA) as a knowledge source for the computational modeling of nominal metaphor interpretation. A vector approximation of the Construction-Integration (CI) model is used for finding the representations of metaphor meanings. The author only uses the *term vectors* of LSA. The meaning of a metaphor is represented by the centroid of a set of vectors, including the tenor, the vehicle, and a few terms related to both (k terms

most related to the tenor are selected among m terms most related to the vehicle). The composed metaphor vector does not directly give the properties highlighted by a metaphor.

Terai and Nakagawa (2008) extended this work. They built a generative probabilistic model based on the dependency counts between nouns and adjectives and between nouns and verbs, treating the adjectives and verbs as the properties of the nouns. The statistical model captures the latent classes where the nouns and their properties are connected, and the latent classes are used as vector dimensions. To interpret a metaphor, a meaning vector is first constructed by applying the method of Kintsch (2000), which is subsequently used to assign saliences to the properties in the latent classes. As an additional step, the properties and their assigned saliences are used to construct a recurrent neural network, in order to model the dynamic interaction between properties. The properties with the highest activation, until the network converges, are taken as the metaphor meanings.

The system *Metaphor Magnet* developed by Veale and Li (2012) is based on the idea that metaphor interpretation works by stereotype expansion and property overlap. For each of the tenor and vehicle concepts, the concept is first expanded with a set of *stereotypes* that are commonly used to describe it. The stereotypes are obtained from Google n-grams using linguistic patterns, such as “NOUN₁ is [a] NOUN₂”. Then, the union of the properties of the concept and its associated stereotypes are all attributed to the concept. The properties, in the forms of adjectives, VERB+ings and VERB+eds, are harvested from the Web using another set of linguistic patterns. In addition, manual filtering was involved in constructing both knowledge sources. The properties highlighted by a metaphor are at the intersection of the tenor’s and the vehicle’s properties.

Meta4meaning differs from the above literature in both knowledge acquisition and modeling metaphor interpretation. We will compare it with the method of Terai and Nakagawa (2008) and *Metaphor Magnet* in the evaluation.

Acquiring Knowledge for Metaphor Interpretation

In this section, we describe the Meta4meaning method for interpreting metaphors. The method has two major components. First, a text corpus is analyzed for associations between pairs of words. Then, for each metaphor to be interpreted, plausible properties (interpretations) are ranked.

Extracting Word Associations

Meta4meaning extracts word associations from corpora based on the statistical significance of their co-occurrence. We consider the associated words as an approximation of a concept’s properties, and their association strengths as the properties’ saliences to the concept.

There are different ways of extracting word relations depending on what exactly is being searched for (Rapp 2002). Concepts and their properties are more likely to have syntagmatic than paradigmatic relations. *Syntagmatic* relations are between co-occurring words, e.g., ‘the shark has six fins’

(shark is *related* to fins). *Paradigmatic* relations in turn exist between words that appear in similar context but usually do not co-occur, e.g., ‘shark’ and ‘sawfish’ (shark and sawfish are *similar*). Statistical association measures are suitable for extracting syntagmatic associations (Rapp 2002; Evert 2008). LSA has also been used to this end (Sahlgren 2006); nevertheless, the bag-of-words distributional models seem more appropriate for capturing paradigmatic associations (Rapp 2002; Peirsman, Heylen, and Geeraerts 2008).

In acquiring word associations, we start with a basic method, applying association measures to the co-occurrence counts of words. We use a 2 billion word web text corpus, *ukWaC*¹, and follow a standard process of acquiring co-occurrence counts. Lemmatization and punctuation removal are first applied to the corpus. The co-occurrence of words is counted within a symmetrical window of size 4, i.e. allowing at most 3 words between the two words, and further limited by sentence boundaries (Lapesa, Evert, and Schulte im Walde 2014). The most frequent 50,000 words are selected as vocabulary, excluding closed class words. We use the *log-likelihood* measure of Evert (2008), more specifically the one for surface co-occurrence, as our association measure, with all negative values set to zero. Finally, the score of an association is normalized by the *L1-norm* of the scores of all the associations of a concept (McGregor et al. 2015).

Moreover, we experiment with two methods of dimensionality reduction, Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), in the hope of achieving better semantic representations. The rationale behind dimensionality reduction is to remove noise and to generalize individual word co-occurrences to associations between related concepts.

SVD and NMF both produce two matrices. One matrix has words as rows and the reduced dimensions as columns (henceforth, *term-dimension-matrix*). The other has the reduced dimensions as rows and the context terms as columns (*dimension-property-matrix*). A concept and its properties are connected via the reduced dimensions.

For both SVD and NMF, we employed the implementations provided by *Scikit-learn*², with default parameters. The SVD model has 900 dimensions, and was obtained with an oversampling factor of 2. When querying the SVD model, no dimension is skipped. The NMF model is built similarly, also with 900 dimensions.

Word associations of a concept, extracted from a large corpus in the above way, cover a long list of words with a big variance in association strength, including small values just above zero. This is in contrast with the properties produced by human subjects in psychological experiments, which is generally a much shorter list of salient properties (Roncero and Almeida 2014). We next describe how Meta4meaning further extracts the best features for metaphor processing.

¹<http://wacky.sslmit.unibo.it/doku.php?id=corpora>.

²<http://scikit-learn.org>.

Focusing on Metaphor Activated Properties

Adjectives typically denote properties (Murphy 2010), and metaphor processing frequently involves *abstraction* (Glucksberg 2001). We take advantage of these two characteristics to separate properties that are most likely to be activated in metaphor interpretation from those less likely.

Adjective properties have a dominant presence in metaphor processing. The three existing metaphor interpretation methods introduced in the Related Work section all include adjectives in their knowledge sources. On the other hand, for noun and verb properties, abstraction is often required for metaphors to work (Glucksberg 2001; Utsumi and Sakamoto 2011). We use a combination of the part-of-speech (POS) information of single words and a large-scale rating of term abstractness to only retain adjective, abstract noun and abstract verb associations.

Turney et al. (2011) derived the abstractness ratings of 114,501 WordNet terms with supervised learning. The abstractness ratings are between 0 and 1, 1 being more abstract. We determine the most frequent POS of a word by consulting WordNet and SUBTLEX-UK (Subtitle-based word frequencies for British English)³.

All adjectives, as well as nouns and verbs with abstractness ratings above 0.5, are retained as metaphor-activated properties. As an example, the resulting highly rated properties for *shark* include ‘bask’, ‘loan’, ‘white’, ‘attack’, ‘see’, ‘grey’, ‘marine’, ‘large’, etc.

Interpreting Metaphors

In Meta4meaning, the association strength between a concept and a property is regarded as an approximation of the property’s salience to the concept. We use the saliences of a property to the tenor and the vehicle respectively to devise a metric, in order to rank the properties for their likelihood of being the interpretations of the metaphor.

In a metaphor, a set of properties are *transferred* or *emerge* in the interaction between the tenor and vehicle’s conceptual spaces (Black 1962; Glucksberg 2001; Moreno 2004). In some metaphors, it seems that the salient properties of the vehicle are transferred to the tenor. In others, however, the metaphor meaning is not among the salient properties of either the vehicle or the tenor: these are called ‘emergent properties’.

In this work, we only consider properties associated with both the tenor and the vehicle. While this may seem similar to the ‘common features’ (Becker 1997) elicited in psychological experiments, there is an important difference: with word associations derived from a corpus, Meta4meaning can also find overlap of low-salience properties.

Ranking of Properties

Meta4meaning ranks properties by their saliences to both the tenor and the vehicle. Taking the product of the saliences emphasizes properties that are strongly associated with both.

Definition 1. The *product of saliences* p_i of property i is

$$p_i = t_i \cdot v_i,$$

where t_i and v_i are the association strengths of the property to the tenor and the vehicle, respectively.

The *salience imbalance hypothesis* predicts that the properties highlighted by a metaphor are among the common properties of the tenor and the vehicle and are more salient to the vehicle than to the tenor (Ortony 1979). Based on this, one can further hypothesize that a larger difference $v_i - t_i$ correlates with higher metaphor aptness.

Definition 2. The *difference of saliences* d_i of property i is

$$d_i = v_i - t_i.$$

We compare the salience difference d_i experimentally to the product of saliences p_i . These two weights measure different, possibly complementary aspects of metaphor properties; we therefore also consider their combination. To avoid making assumptions about the distributions of the values, we produce a combined measure by considering the *ranks* of properties with respect to p_i and d_i and associate the property with the better of these.

Definition 3. The *combined metaphor rank* c_i of property i is

$$c_i = \min(\text{rank}(i, p), \text{rank}(i, d))$$

where $\text{rank}(i, x) = |\{j \mid x_j \leq x_i\}|$.

Alternative Measures

An alternative to taking the product of saliences t_i and v_i is to take their sum. Addition promotes properties that are salient for at least one of the tenor and vehicle, and is consistent with those metaphors that highlight a salient property of either one. When the property is more salient to the vehicle than the tenor, the behavior pairs well with the salience imbalance hypothesis.

Definition 4. The *sum of saliences* p_i^+ of feature i is

$$p_i^+ = t_i + v_i.$$

The *combined metaphor sum rank* c_i^+ of property i is

$$c_i^+ = \min(\text{rank}(i, p^+), \text{rank}(i, d)).$$

We use p_i^+ and c_i^+ to emphasize that these are the variants of measures p_i and c_i . Which one is more appropriate depends, among other things, on how the association strengths have been derived and processed.

We also compare the performances of all the above measures when SVD or NMF has been used to reduce the dimensionality of the word co-occurrence matrix. The above measures are then applied to the term-dimension-matrix, and the obtained score vector is multiplied by the dimension-property-matrix. The resulting vector contains the saliences of the properties to the metaphor.

Matrices factorized using SVD contain both positive and negative values, meaning that the above intuition regarding the pointwise product of vectors does not apply. For this reason, we do not apply p_i or the combined c_i to SVD matrices.

³<http://crr.ugent.be/archives/1423>.

Testing Hypotheses

We also empirically test two popular hypotheses about metaphor interpretation. First, the salience imbalance hypothesis (Ortony 1979), that metaphor interpretations are among the common properties of the tenor and the vehicle and are more salient to the vehicle than to the tenor, corresponds in our setting to $v_i > t_i > 0$.

Definition 5. The *salience imbalance hypothesis* is that

$$v_i > t_i > 0.$$

Second, emergent properties, not associated with either the tenor or the vehicle, have been observed dominating metaphor interpretations in psychological experiments. We investigate whether it is still the case for association strengths derived from large text corpora through statistical association measures, which cover both strong and weak associations.

Definition 6. The *emergent property hypothesis* in its weakest form is that

$$v_i \neq 0 \text{ and } t_i \neq 0.$$

Comparison to Other Methods

We compare the performance of Meta4meaning to two others from the literature: the method of Terai and Nakagawa (2008), and Metaphor Magnet by Veale and Li (2012).

The method of Terai and Nakagawa (2008) consists of two processes: categorization followed by dynamic interaction. Meta4meaning is comparable to the categorization process, which composes the metaphor vector by adding together not only the tenor and the vehicle vectors but also a few vectors corresponding to words related to both the tenor and the vehicle. To find the related words, the cosine similarity of vectors has to be computed across the entire vocabulary, and a few parameters (m, k) have to be tuned.

In implementing the categorization method of Terai and Nakagawa (2008), we use the current NMF model, which is similar to the generative probabilistic model employed in the original work, as well as SVD. In both cases, we use $m = 250$ and $k = 5$ following the original paper.

Metaphor Magnet (Veale and Li 2012) is available via an application programming interface⁴. Given a metaphor of the form ‘*tenor* is [a] *vehicle*’, it returns a list of entries in the format of ‘property:stereotype(score)’. *Score* indicates the salience of the *property* to the *stereotype*. We tallied the scores of each *unique* property (stereotypes may overlap on properties), and ranked the properties based on their accumulated scores.

Evaluation

We evaluate Meta4meaning using metaphor interpretations acquired from human subjects by Roncero and Almeida (2014). Our empirical goals are to test the various ways of deriving interpretation rankings (Definitions 1–3). We compare them to the alternative measures described above – Definition 4 and use of SVD or NMF for dimensionality reduction – and to other methods proposed in the literature.

⁴<http://ngrams.ucd.ie/metaphor-magnet-acl/>.

Table 1: Example metaphor interpretations from Roncero and Almeida (2014).

Metaphor	Interpretation	Frequency
Alcohol is a crutch	Helpful	5
	Dependable	4
	Addictive	3
	Support	3
	Disability	2
	Aid	2
	Problem	2

Evaluation Dataset

Roncero and Almeida (2014) collected interpretations of 84 nominal metaphors. Subjects were given a metaphor such as ‘alcohol is a crutch’ and were then asked to provide up to three properties which “the vehicle word [crutch] was expressing about the topic [alcohol]”. For each metaphor, the dataset of Roncero and Almeida (2014) provides those interpretations (properties) that were mentioned at least twice, together with their frequencies. Table 1 shows the relevant information for the metaphor ‘alcohol is a crutch’.

We use this dataset in our evaluation as follows. First, we only use the salient interpretations. Following Roncero and Almeida (2014), we consider as salient those interpretations that have been mentioned by at least 25% (i.e. 4) of the twenty participants. There are eight metaphors which have no salient interpretations, leaving 76 metaphors.

Second, we carried out some simple linguistic processing by hand on the terms (tenors, vehicles and interpretations) in the dataset and the properties output by Meta4meaning. For the tenors and vehicles, we decapitalized all terms except proper nouns, changed plurals to singulars, and converted phrases to single words when possible (e.g., from “hard cover” to “hardcover”). The tenors and vehicles do not include complex phrases so they are amenable to the methods for nominal metaphors described in this paper. We then stemmed both the human-given interpretations and the properties produced by Meta4meaning before comparing them. The same has been done in the work of Kintsch and Bowles (2002) and Roncero and Almeida (2014).

In total, 76 metaphors are used in the evaluation, each of which has from one to four salient interpretations (together accounting to 145 interpretations). Therefore, a metaphor has slightly under two salient interpretations on average.

Evaluation Metric

We measure the performances of metaphor interpreters using *recall*. Given the human interpretations of a metaphor (from Roncero and Almeida (2014), as described above) and a list of properties given by the method (predicted interpretations), the recall for the metaphor is defined as the fraction of human interpretations that were also predicted by the computer. If a metaphor has n human interpretations and n' of them are among the predicted properties, the recall is n'/n . For the whole set of metaphors, the recall is computed as the average over all individual metaphors.

Since the automated methods described in this paper and elsewhere can produce long lists of possible interpretations,

Table 2: Examples of metaphor interpretations by Meta4meaning (salient interpretations of the evaluation dataset in bold).

Ranking	Metaphor		
	cloud is cotton (per p_i)	life is a joke (per d_i)	alcohol is a crutch (per c_i)
1	white	funny	psychological
2	cover	story	dependence
3	thick	make	drug
4	black	anecdote	emotional
5	blue	good	addiction
6	fluffy	humour	week
7	thin	hilarious	help
8	soft	trivia	mental
9	layer	cruel	cope
10	heavy	fun	dependent

it is relevant to ask how good the most highly ranked properties are. We therefore report *recall at k* (or “@k” for short), defined as the recall when using only the top k properties of a computer-generated ranking.

For instance, recall @10 considers for each metaphor the ten properties ranked highest by the computer, and calculates how many of the salient interpretations are included in these ten. Table 2 shows the top ten interpretations for three different metaphors and for different measures introduced above, representing successful results of the method. For the metaphor ‘cloud is cotton’, Meta4meaning (with measure p_i) provides all three salient interpretations and thus has a recall of 100% @10. Recall @5 is 33% since only one out of three salient interpretations is among the top five properties. In the experiments, we report the average recalls @5, @10, @15, @25 and @50.

Results and Analysis

We first report on the performance of Meta4meaning, and then compare it to other methods. To compare the recall performances of different measures, we use the Wilcoxon signed-rank test, which tells whether the difference between the average recalls of two measures is significant or not. To test for a statistical difference between recalls, two measures are compared by pairing the recalls for every metaphor at each of the five k s.

Metaphor Interpretation Performance Recall of metaphor interpretations by Meta4meaning, using the product of saliences p_i , the difference of saliences d_i , and the combined rank c_i are given in the first three rows of Table 3.

Among the three measures, the combined rank achieved the best recalls, followed by the product of saliences, while the recalls obtained with the difference of saliences are consistently inferior ($p < .05$ in both tests). For the combined rank, the recall @10 of 0.303 indicates that about 30% of the salient interpretations are among the top ten properties listed by this variant. This can be considered a strong result given the difficulty of the metaphor interpretation task. While the difference of saliences is on its own inferior, the good performance of the combined rank suggests that the difference

Table 3: Recall of metaphor interpretations by Meta4meaning (best performance in bold).

Meta4meaning variant	Recall				
	@5	@10	@15	@25	@50
Product of saliences p_i	0.215	0.274	0.304	0.325	0.466
Difference of saliences d_i	0.193	0.227	0.27	0.308	0.391
Combined rank c_i	0.221	0.303	0.339	0.397	0.454
Sum of saliences p_i^+	0.164	0.239	0.316	0.368	0.41
Combined sum rank c_i^+	0.184	0.254	0.299	0.384	0.462

of saliences and the product of saliences are complementary and recognize different interpretations.

The two last rows of Table 3 show the recalls of metaphor interpretations by the alternative measures using the sum of saliences p_i^+ and the respective combined sum rank c_i^+ rather than the product of saliences. The combined sum rank performs significantly worse than the combined rank ($p < .01$). Other differences between the measures are not statistically significant.

Effect of Dimensionality Reduction We now evaluate the effect of dimensionality reduction (SVD or NMF) on the metaphor interpretation performance. Tables 4 and 5 show the recall of metaphor interpretations when using SVD and NMF. The results are clearly inferior to the results obtained without dimensionality reduction in Table 3 above. For instance, to reach a recall of about 20%, one needs to consider around top 50 predicted properties for each metaphor, while top five were sufficient in Table 3.

An analysis of the relative performances of different measures within tables shows some interesting results. In the case of SVD, the sum of saliences performs significantly better than the combined sum rank ($p < .01$), whereas there are no significant differences between measures for NMF. Clearly, the way the association strengths of properties have been obtained has a strong effect on how to best combine them.

Table 6 gives an overview of the best performing variants with and without dimensionality reduction. The variants with dimensionality reduction perform much worse than the variant without dimensionality reduction ($p < .001$ in both tests).

According to a preliminary observation, dimensionality reduction seems to select certain aspects of a concept and generalize those over the vocabulary, i.e. properties not salient to the concept per se but similar to the salient ones, rise to the top. At the same time, the unselected dimensions are downplayed. Moreover, the accuracy of selecting the salient aspects of a concept subjects to the methods chosen for dimensionality reduction and the associated parameter setting. To achieve optimal results, it requires systematic experiments, which is out of the scope of this paper. As to be discussed below, it is possible for Meta4meaning to capture those interpretations that are at least salient to one of the tenor and the vehicle. In the unlucky cases where dimensionality reduction actually reduces the saliences of the interpretations (properties), the recalls decrease consequently, although the contrary may happen – the saliences of the interpretations are raised by dimensionality reduction.

Table 4: Recall of metaphor interpretations using SVD (best performance in bold).

Method	Recall				
	@5	@10	@15	@25	@50
SVD, Difference of saliences d_i	0.037	0.064	0.075	0.088	0.188
SVD, Sum of saliences p_i^+	0.057	0.081	0.099	0.145	0.226
SVD, Combined sum rank c_i^+	0.053	0.061	0.079	0.121	0.191

Table 5: Recall of metaphor interpretations using NMF (best performance in bold).

Method	Recall				
	@5	@10	@15	@25	@50
NMF, Product of saliences p_i	0.069	0.076	0.091	0.113	0.144
NMF, Difference of saliences d_i	0.054	0.061	0.061	0.096	0.114
NMF, Combined rank c_i	0.073	0.078	0.089	0.107	0.16
NMF, Sum of saliences p_i^+	0.076	0.098	0.098	0.115	0.192
NMF, Combined sum rank c_i^+	0.073	0.08	0.102	0.11	0.173

It seems that the former occurred more frequently than the latter in this evaluation.

Comparison to Other Methods The recall performance of salient metaphor interpretations by the categorization method of Terai and Nakagawa (2008) (T&N) and by Metaphor Magnet are given in Table 7, together with the combined rank results for Meta4meaning.

T&N performs better using SVD than using NMF (which is closer to the original version) ($p < .01$). T&N with the current SVD model is not as good as Metaphor Magnet ($p < .01$), and Meta4meaning outperforms all the other methods ($p < .001$).

Testing the Saliency Imbalance Hypothesis and the Emergent Property Hypothesis Above, we formulated two hypotheses for metaphor interpretations: the saliency imbalance hypothesis ($v_i > t_i > 0$) and the emergent property hypothesis $v_i \neq 0$ and $t_i \neq 0$.

We now study how often the hypotheses hold for the association strengths. Of the 76 metaphors, four have either the tenor or the vehicle missing from our 50k vocabulary: ‘cigarette is a timebomb’, ‘desk is a junkyard’, ‘tree trunk is a straw’, and ‘sermon is a sleeping pill’, which together have nine human interpretations in the dataset. In addition, there are two other human interpretations not in our vocabulary: ‘breakable’ for the metaphor ‘health is glass’, and ‘extinguished’ for ‘typewriter is a dinosaur’. We used the remaining 134 metaphor interpretations in testing the hypotheses, looking at the association strengths of the interpretations with respect to the corresponding tenor (t_i) and vehicle (v_i).

Among the 72 cases where $t_i > 0$ and $v_i > 0$, there are 20

Table 6: Recall of metaphor interpretations, effect of dimensionality reduction (best performance in bold).

Method	Recall				
	@5	@10	@15	@25	@50
Meta4meaning (c_i)	0.221	0.303	0.339	0.397	0.454
SVD, Sum of saliences p_i^+	0.057	0.081	0.099	0.145	0.226
NMF, Sum of saliences p_i^+	0.076	0.098	0.098	0.115	0.192

Table 7: Recall of metaphor interpretations (with best performance in bold), compared to the method of Terai and Nakagawa (2008) (T&N) and to Metaphor Magnet (Veale and Li 2012).

Method	Recall				
	@5	@10	@15	@25	@50
T&N, NMF	0.053	0.061	0.068	0.121	0.195
T&N, SVD	0.053	0.072	0.094	0.167	0.252
Metaphor Magnet	0.102	0.155	0.181	0.193	0.239
Meta4meaning (c_i)	0.221	0.303	0.339	0.397	0.454

cases (28%) where the saliency imbalance hypothesis does not hold, i.e., where the property is actually more salient to the tenor than to the vehicle ($t_i > v_i > 0$). This leads to the conclusion that a metaphor interpretation does not have to be more salient to the vehicle than to the tenor, at least for properties acquired from corpora using this method.

The emergent property hypothesis was tested in a similar fashion. Among the 134 metaphor interpretations, there are 48 cases (36%) where the saliency of the interpretation to the tenor is $t_i = 0$ and 24 cases (18%) where the saliency to the vehicle is $v_i = 0$. In ten of these cases (7%), $t_i = 0 = v_i$. Altogether, 62 metaphor interpretations (46%) do not appear in the corpus-derived list of properties of either the tenor or the vehicle. This result may be evidence for emergent properties of metaphors. It also highlights issues with our current approach. We address them next.

Error Analysis

Let us now take a closer look at the cases where Meta4meaning is not successful in recalling metaphor interpretations.

First, some of the properties proposed by Meta4meaning are actually semantically the same as or very similar to interpretations given in the dataset. For instance, the metaphor ‘city is a jungle’ has interpretation ‘crowded’, while Meta4meaning suggests ‘dense’, at rank 3. Other examples of semantically similar interpretation–property pairs include ‘scary’/‘fear’, ‘challenging’/‘difficult’, ‘destructive’/‘destroy’. These properties can be considered to be correct interpretations, and one could argue that the issue is more in the evaluation methodology than in the metaphor interpretation method.

As mentioned previously, four metaphors and additionally two metaphor interpretations are not included in our 50k vocabulary, so Meta4meaning has no way to provide (correct) interpretations for these. Increasing the size of the vocabulary could help here, but it could also add noise and reduce recall. Computationally, a larger vocabulary is not a problem for calculating association scores but might impose a challenge for dimensionality reduction.

Recall that 46% of metaphor interpretations have zero saliency to either the tenor or the vehicle. They are thus entirely missed by all the variants of Meta4meaning, since Meta4meaning only considers overlap properties ($t_i > 0$ and $v_i > 0$). Dropping the overlap requirement might potentially increase recall, but it would add a lot of noise as well.

As an example, the metaphor ‘the woman is a cat’ has only one interpretation – ‘independent’. Independent, in our method, is not associated with ‘cat’ at all, but ‘feral’ and ‘wild’ are among the salient properties of cat (but neither one is associated with ‘woman’ too). Clearly, feral and wild both touch upon independent, but they are in the language of talking about cats, not women. A method allowing one to find analogies could be helpful to solve some cases like this, by finding the property (‘independent’) of women that is similar to the properties ‘feral’ and ‘wild’ of cats.

In 7% of metaphor interpretations, $v_i = 0 = t_i$, and Meta4meaning has no means of identifying them. Examples of such interpretations are ‘annoying’ for the metaphor ‘obligation is a shackle’ and ‘life’ for ‘money is oxygen’. Part of this can be considered a failure at Meta4meaning’s word association extraction step. One might expect matrix factorization to help with these cases; more work is needed here.

54% of the metaphor interpretations are among the overlap properties of the tenors and the vehicles. Meta4meaning achieved a recall of about 30% when only considering the top ten ranked properties. The variants of Meta4meaning promote the properties that are relatively salient, among the overlap properties, to both the tenor and the vehicle (i.e. the overlap properties may not be salient to the tenor and the vehicle after all), and partly the ones relatively salient to either the tenor or the vehicle. Metaphor interpretations of such characteristics have the chance to be captured. Nevertheless, Meta4meaning can not spot the interpretations that have relatively low saliences to both the tenor and the vehicle.

Conclusions and Future Work

We have described Meta4meaning, a method for interpreting metaphors. Meta4meaning uses corpus-derived word associations so it has a large vocabulary and can potentially be applied to languages other than English. We evaluated Meta4meaning empirically using salient human interpretations of metaphors and compared its performance to other leading methods. The results indicate that Meta4meaning has high recall performance, considering the difficulty of the task, and substantially outperforms other methods.

We proposed and compared several ways of combining the salience of a property to the tenor with its salience to the vehicle in order to rank the properties as possible metaphor interpretations. The combinations are based on three principles: salience aggregation (the product or sum of saliences), salience difference, and combining the results of the two. Salience aggregation captures more correct metaphor interpretations than the difference. Combining the two improves the results. However, the current combination method is simple, and a more sophisticated method may bring further improvement. Moreover, future research could be dedicated to better understanding when salience aggregation and difference work best.

In addition to direct co-occurrence-based word associations, we also experimented with dimensionality reduction (SVD and NMF). However, the results we obtained with

them were inferior to those obtained directly with association strength. Further work is needed to investigate how to make better use of the co-occurrence matrix in the context of metaphor interpretation, possibly with dimensionality reduction.

Our analysis of the emergent property hypothesis shows that it holds for our corpus-derived word associations: almost half of the interpretations have no association at all with the tenor or the vehicle. It would be interesting to discover a mechanism of how non-salient properties emerge as interpretations of metaphors.

Given a metaphor, Meta4meaning provides a list of interpretations with varying weights. The interpretations cover multiple aspects of the tenor and vehicle, including various linguistic forms and closely related meanings. Such a multitude of interpretations can be a great benefit at least in two ways. First, it offers opportunity for context adaption. Metaphors are always used in a context, and this context could potentially be used to increase the weights of context-relevant properties so that different contexts result in different interpretations. Second, when Meta4meaning is used as part of a creative system, such as a computer novelist, its rich repository of semantically adjacent words can help find suitable metaphors.

Acknowledgments

This work has been supported by the European Commission under the FET grant 611733 (ConCreTe) and by the Academy of Finland under grant 276897 (CLiC).

References

- Becker, A. H. 1997. Emergent and common features influence metaphor interpretation. *Metaphor and Symbol* 12(4):243–259.
- Black, M. 1962. *Models and Metaphors: Studies in Language and Philosophy*. New York: Cornell University Press.
- Evert, S. 2008. Corpora and collocations. In Lüdeling, A., and Kytö, M., eds., *Corpus Linguistics. An International Handbook*, volume 2. Berlin: Mouton de Gruyter. 1212–1248.
- Glucksberg, S. 2001. *Understanding Figurative Language: From Metaphor to Idioms*. Oxford University Press.
- Kintsch, W., and Bowles, A. R. 2002. Metaphor comprehension: What makes a metaphor difficult to understand? *Metaphor and Symbol* 17(4):249–262.
- Kintsch, W. 2000. Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review* 7(2):257–266.
- Lapesa, G.; Evert, S.; and Schulte im Walde, S. 2014. Contrasting syntagmatic and paradigmatic relations: Insights from distributional semantic models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics, SEM '14*, 160–170.
- McGregor, S.; Agres, K.; Purver, M.; and Wiggins, G. 2015. From distributional semantics to conceptual spaces: A novel computational method for concept creation. *Journal of Artificial General Intelligence* 6(1):55–86.

- Moreno, R. E. V. 2004. Metaphor interpretation and emergence. *UCL Working Papers in Linguistics* 16:297–322.
- Murphy, M. L. 2010. *Lexical Meaning*. Cambridge University Press.
- Ortony, A.; Schallert, D. L.; Reynolds, R. E.; and Antos, S. J. 1978. Interpreting metaphors and idioms: Some effects of context on comprehension. *Journal of Verbal Learning and Verbal Behavior* 17(4):465–477.
- Ortony, A. 1979. The role of similarity in similes and metaphors. In Ortony, A., ed., *Metaphor and Thought*. Cambridge University Press. 186–201.
- Peirsman, Y.; Heylen, K.; and Geeraerts, D. 2008. Size matters: Tight and loose context definitions in english word space models. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, 34–41.
- Rapp, R. 2002. The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of the Nineteenth International Conference on Computational Linguistics*, 1–7.
- Richards, I. A. 1936. *The Philosophy of Rhetoric*. London: Oxford University Press.
- Roncero, C., and Almeida, R. G. 2014. Semantic properties, aptness, familiarity, conventionality, and interpretive diversity scores for 84 metaphors and similes. *Behavior Research Methods* 47(3):800–812.
- Sahlgren, M. 2006. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. Ph.D. Dissertation, University of Stockholm, Stockholm, Sweden.
- Terai, A., and Nakagawa, M. 2008. A corpus-based computational model of metaphor understanding incorporating dynamic interaction. In *Proceedings of The Eighteenth International Conference on Artificial Neural Networks, ICANN '08*, 443–452.
- Turney, P. D.; Neuman, Y.; Assaf, D.; and Cohen, Y. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, 680–690.
- Utsumi, A., and Sakamoto, M. 2011. Indirect categorization as a process of predicative metaphor comprehension. *Metaphor and Symbol* 26(4):299–313.
- Veale, T., and Li, G. 2012. Specifying viewpoint and information need with affective metaphors: A system demonstration of the metaphor magnet web app/service. In *Proceedings of the ACL 2012 System Demonstrations, ACL '12*, 7–12.

One does not simply produce funny memes!

– Explorations on the Automatic Generation of Internet humor

Hugo Gonalo Oliveira, Diogo Costa, Alexandre Miguel Pinto

CISUC, Department of Informatics Engineering

University of Coimbra, Portugal

hroliv@dei.uc.pt, dcosta@student.dei.uc.pt, ampinto@dei.uc.pt

Abstract

This paper reports on the automatic generation of image macro based Internet memes – potentially funny combinations of an image and text, intended to be spread in social networks. Memes are produced for news headlines for which, based on linguistic features, a suitable macro is selected and the text is adapted. The generation method is described together with its current implementation, which integrates a variety of tools and resources. Illustrative examples are also presented. Results of a human evaluation showed that, despite positive and neutral assessments, overall, automatically-generated memes are still below those produced by humans.

Introduction

Internet memes are a current trend, typically jokes, that gain influence through transmission in social media (Davison, 2012). A popular kind is the image macro – a set of stylistic rules for adding text (e.g. “*One does not simply X*”, “*What if I told you Y*”) to images, with a specific semantics. Most memes are a product of human creativity and their automatic generation is thus a challenge for computational creativity.

This paper reports on an exploratory approach for the automatic generation of Internet memes for news headlines – or better, protomemes, which may eventually become memes if spread through the Web. Following our previous work (Costa, Gonalo Oliveira, and Pinto, 2015), where memes were generated for trendy people, towards whom famous quotes were adapted and added to their images, we now present a system with a similar goal, but with significant differences. Given a headline, MEMEGERA 2.0: automatically selects a suitable well-known image macro from a predefined set; adapts the text according to the macro rules; and combines the text with the image. The result is ready to be consumed. Following the recent trend of using Twitter as a showcase for linguistic creativity (Veale, Valitutti, and Li, 2015), an implemented Twitterbot posts a new meme every hour. Besides the main challenge, the bot’s feed can be used as an alternative and funny way of following recent news.

The illustrative examples presented here, as well as the memes posted on Twitter, confirm that our main goal was achieved: given a headline, a coherent combination of image and text, easily recognisable as an Internet meme, and with a relation to the headline, is produced. A more challenging goal, discussed in the end of the paper, involves the

production of novel artefacts with humor value. Though limited in terms of size, an evaluation survey pointed out that, despite frequently producing coherent text, there is work to do in the selection of the suitable image macros, as well as on the surprise and humor value of the produced artefacts. Yet, although better than MEMEGERA 2.0 overall, human-generated memes also failed to consistently reach top-performance, which shows that the creation of memes is challenging, even for humans.

In the remainder of this paper, we provide background knowledge and work on related topics, including computational linguistic creativity and computational humor. The meme generation method is then described and followed by details on its current implementation, for Portuguese, our native language. MEMEGERA 2.0 exploits a variety of tools and resources for collecting data, processing text, and for combining and publishing the results. Several illustrative memes generated by MEMEGERA 2.0 are then shown and contextualised. Before concluding, the conducted evaluation and its results are discussed.

Background and Related Work

Given a news headline, MEMEGERA 2.0 generates Internet memes that combine an image and a piece of text. This involves the automatic selection of an adequate image macro for the headline and the adaptation of the headline text according to the macro, with a humor intent. After a short introduction on the concept of Internet memes, this section enumerates previous work on computational linguistic creativity with a focus on humor generation. Not forgetting the role the image plays on the memes, the section ends with a reference to visual humor and its combination with text.

Internet Memes

The term *meme* originally refers to *an idea, behaviour, or style that spreads from person to person within a culture* (Dawkins, 1976). In the Web 2.0 era, it was adopted to denote *a piece of culture, typically a joke, which gains influence through online transmission* (Davison, 2012). A popular kind of meme is the image macro, which involves a set of stylistic rules for adding text to images. The same text can be added to different images or different text can be added to a common image. We focus on latter. Yet, the new piece of text should be analogous to the original, which is

either achieved by using a similar linguistic template or by transmitting a similar idea. Well-known memes of this kind include an image of Boromir, from the “Lord of the Rings”, with a phrase that fills the template “*One does not simply X*”, as an analogy to the original “*One does not simply walk into Mordor*”; Morpheus, from the “Matrix” movie, with “*What if I told you Y*”; or Batman slapping Robin, with a personalised text in their speech balloons.

Davison (2012) separates a meme into three components: manifestation – the observable part of the meme phenomenon; behavior – which creates the manifestation and is the action taken by an individual in service of the meme; and ideal – the concept or idea conveyed. For the memes by MEMEGERA 2.0, the manifestation is the image, the behavior involves adding a piece of text to the image, and the ideal is to make fun of an event through its analogy with previous uses of the macro.

Linguistic Creativity with a focus on Humor

The domain of computational linguistic creativity is discussed by Veale (2012), who highlights the Web as a large and open source of everyday knowledge, especially on the way language is used, and suitable for exploitation by creative systems. Linguistic creativity can take familiar knowledge, sometimes old-forgotten references, and re-invent it in novel and surprising ways. It often relies in the intelligent adaptation of well-known text to a new context.

Notable examples of computational linguistic creativity include the generation of metaphors, neologisms, slogans, poetry and humor. On the former, Veale and Hao (2008) exploit a small set of common textual patterns in the Web for acquiring salient properties of nouns, then used for explaining known metaphors and generating new ones (e.g. *Paris Hilton is a pole*). Smith, Hintze, and Ventura (2014) create neologisms by blending two concepts, either from language, or from pop culture lists (e.g. *neologism + creator = Nehovah*). Gatti et al. (2015) adapt well-known expressions (e.g. clichés, song and movie titles) to suit as creative slogans or news headlines in a four-step approach: (i) retrieval of recent news; (ii) keyword extraction; (iii) pairing news with expressions, based on their semantic similarity; (iv) replacing one word of the expression by a word related to the news, based on dependency statistics. For instance, given an article about the Euro crisis, the expression *What the world is coming to* may be adapted to *What the Euro is coming to*.

Lexical replacement has also been applied to other creative domains, such as poetry or humor. For instance, Toivanen, Gross, and Toivonen (2014) generate poems inspired by a news article through the replacement of certain words, in human-created poems, with associations obtained from Wikipedia and from the given article. Valitutti et al. (2013) explored the generation of adult humor, based on the replacement of a word in a short message. The new word should introduce incongruity and lead to a humorous interpretation, achieved by three constraints: (i) match the part-of-speech and either rhyme or be orthographically similar to the original word; (ii) convey a taboo meaning (e.g. an insult or sexual); (iii) occur at the end of the message and keep the

coherence of the original sentence. An illustrative output is: *I've sent you my fart.. I mean 'part' not 'fart'....*

Humor has been studied from a variety of perspectives, such as psychology, philosophy, linguistics, and also via the computational approach. Raskin (2008) compiles research on humor, also covering an overview on computational approaches to verbal humor, up until 2008. Those cover different types of jokes, such as punning riddles or funny acronyms.

Early work by Binsted and Ritchie (1994) implemented the JAPE system for generating punning riddles. It exploits: a lexicon with syntactic and semantic information on words and their meaning; a set of schemata for combining two words based on their lexical or phonetic relationships; and a set of templates that render the riddle (e.g. *What do you get when you cross X with Y?*).

The HAHAcronym (Stock and Strapparava, 2005) system rewrites existing acronyms with a humor intent. It relies on an incongruity detector and generator that, after parsing existing acronyms, decides what words to keep unchanged and what to replace. Replacing words should keep the initial letter of the original and, at the same time, belong to opposing domains or be antonym adjectives, while also considering rhythm and rhymes (e.g. the acronym FBI may become *Fantastic Bureau of Intimidation*). Given a concept and an attribute, HAHAcronym can also generate new acronyms from scratch, which must be words in a dictionary (e.g. ‘processor’ and ‘fast’ results in OPEN – *Online Processor for Effervescent Net*).

Besides English, there were attempts for generating puns in Japanese (e.g. Sjöbergh and Araki (2007)), but we are not aware of any work of this kind for Portuguese.

In addition to those that share our final goal – to generate humor – the aforementioned works reuse familiar knowledge and adapt it to a new context, as MEMEGERA 2.0 does with the macros, known by the general audience and adapted to the context of a headline, obtained from the Web. Depending on the selected macro, text adaptations may range from none, to replacing a single word or longer fragment, in a similar fashion to those that rely on lexical replacement for producing different kinds of linguistically-creative artefacts. Given the key role of the images of memes, the following section focuses on humor through images or their combination with text.

Humor Generation with Images

Internet memes present some differences towards verbal humor and share some similarities with cartoons, which have also been studied from a scientific point of view Hempelmann and Samson (2008). For instance, meme characters may transmit emotions, which would have to be described in verbal jokes; and incongruity can be found in the picture, in the text, or in their combination.

Besides our previous approach (Costa, Gonçalo Oliveira, and Pinto, 2015), where an adapted quote was added to the image of a character, we are not aware of published material on the autonomous generation of Internet memes. Existing web services for aiding meme generation rely only on the user input of both images and text.

There is work, however, on exploring images to make chat conversations more enjoyable. CAHOOTS (Wen et al., 2015) is an online chat system that suggests humorous images, including memes, to be used in a conversation, based on the last message or image received. Although the system does not produce humor autonomously, it is designed to maximize its use by humans, who decide whether to send the images or not.

Other automatic approaches for combining images and text include Grafik Dynamo (2005) and “Why Some Dolls Are Bad” (2008), by Kate Armstrong¹, where a narrative is dynamically generated by combining sequences of images, retrieved from social networks, with speech balloons. But the result is often non-sense.

Meme Generation Procedure

This section provides a high-level description of the current procedure for meme generation. Specific implementation details are provided in the next section.

Our current approach is significantly different our previous (Costa, Gonalo Oliveira, and Pinto, 2015), where memes were focused on public figures (characters) trending on Twitter, and built from their image, retrieved from the Web, and a famous quote, where the last word was replaced by one associated to the character, given its presence in tweets that mentioned its name. Generated quotes were ranked by their humor potential, considering features such as ambiguity and slang. Besides being more limited than the current approach, it could use unlicensed images, and the result was potentially offensive to the involved characters.

On the other hand, the memes currently produced are more “traditional”, in a sense that they reuse well-known macros with a text adapted according to their features. Another key difference is that memes can now be seen as an answer or commentary to a headline, and often serve as its creative rewriting.

For a given headline, the current generation procedure automatically: (i) selects a suitable image macro from a predefined set; (ii) adapts the text according to the selected macro; and (iii) adds the text to the image. More precisely, given a set of headlines H and a set of trigger rules for each supported macro M :

1. It checks whether each $h \in H$ matches any $m \in M$. If so, add the headline-macro pair (h, m) to the set of candidate pairs P ;
2. If $|P| > 0$, select a subset $T \subset P$ where $(h, m)_t \in T$ mentions a Twitter trend. If $|T| = 0$, keep using P ;
3. Select a random pair $(h, m)_r \in T$, adapt h 's text according to m 's stylistic rules, add the result to m 's image, and return it.
4. If $|P| = 0$, the system can either do nothing or use a fallback meme, if there one is set (see next section).

Implementation

Although our method is language-independent, similarly to its predecessor, MEMEGERA 2.0 targets Portuguese. It is implemented in Java and exploits several available resources for the computational processing of Portuguese, as well as

¹<http://katearmstrong.com/>

available Web APIs. It is also working as a Twitterbot, under the name *@MemeGera*. The generation procedure is repeated every hour, for the 25 most recent Portuguese news, and the result is posted in Twitter. A high-level architecture of MEMEGERA 2.0 is depicted in figure 1. First stage deals with data collection, the second assigns image macros to headlines, and the third combines the produced text and selected image.

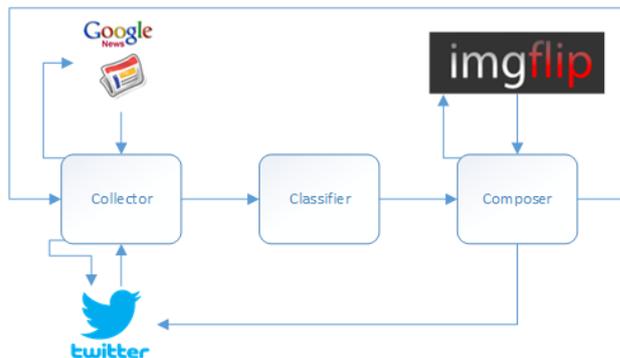


Figure 1: High-level architecture of MEMEGERA 2.0.

After describing the tools and resources involved in the implementation of MEMEGERA 2.0, this section uncovers lower level details of the developed headline classifier, including the currently supported image macros and the rules for pairing macros with headlines and to adapt their text.

Tools and Resources

Portuguese news headlines are collected automatically from the Google News RSS feed². To minimise the generation of ‘black humor’, a filter is applied to discard headlines that mention deaths or casualties. In the last stage, the meme text is added to the macro with the help of the Imgflip API³. The resulting meme, ready to be consumed, is posted on the social network Twitter⁴ with the help of Twitter4J⁵.

The second stage is where linguistic resources are used. To select a suitable meme for a headline, the headline text is first part-of-speech tagged and lemmatised. For this purpose, a tagger based on the OpenNLP⁶ toolkit is used with the Portuguese models, and with the LemPORT (Rodrigues, Gonalo Oliveira, and Gomes, 2014) lemmatiser.

To identify the sentiment of the headline words, we use SentiLex (Silva, Carvalho, and Sarmiento, 2012), where Portuguese words have their polarity annotated. When inflections are required to produce the resulting text, we resort to LABEL-LEX⁷, a morphological lexicon for Portuguese. When verbs need to be nominalised, we resort to Nomlex-PT (de Paiva et al., 2014), a nominalisation lexicon for Portuguese.

²https://news.google.com/news?cf=all&hl=pt-PT&pz=1&ned=pt-PT_pt&output=rss

³<https://api.imgflip.com/>

⁴<https://twitter.com/>

⁵<http://twitter4j.org/>

⁶<https://opennlp.apache.org/>

⁷http://label.ist.utl.pt/pt/labellex_pt.php

The identification of the most relevant word in the headline is simplified by the selection of the less frequent noun, verb or adjective, according to the frequency lists of the AC/DC project (Santos and Bick, 2000). The selected word has still to be in those lists. We also use the proverbs available in the scope of project Natura⁸. The semantic similarity between the headline and a proverb is computed by the average similarity between the nouns, verbs and adjectives they contain, using the PMI-IR (Turney, 2001) method on the Portuguese Wikipedia.

Covered Macros

A broad range of image macros is used nowadays on the social web. Some are more popular than others and each macro has its own style and semantics, expressed as a specific kind of message, either through a fixed textual template, an intention, or a sentiment, among others. We have looked both at popular memes and at a sample of headlines to manually identify textual regularities that would suit certain macros. Currently, MEMEGERA 2.0 covers the following, for which we describe the meaning, according to the *KnowYourMeme* website⁹ (examples are shown in the next section):

- *Brace Yourselves* is used as an announcement of something.
- *One Does Not Simply* points out a difficult task.
- *Not Sure If* represents an internal monologue with underlying uncertainty.
- *Success Kid* transmits a successful achievement.
- *Sad Keanu* transmits a sad event.
- *Bad Luck Brian* transmits an embarrassing event.
- *Condescending Wonka* expresses a sarcastic message.
- *Ancient Aliens* explains inexplicable phenomena as the direct result of aliens.
- *Money Money* is related to (large amounts of) money.
- *Matrix Morpheus* reveals something unexpected.
- *Wise Confucius* gives an advice that turns out to be a pun.
- *Am I The Only One* voices the feeling of not following a trend.
- *X, X Everywhere* points out an emerging trend.

Pairing macros with headlines

In order to assign the most suitable macro to a news headline and to produce a meme, a rule-based classifier was developed to run on the headline text. Classification is currently based on a set of trigger rules over features extracted by the aforementioned linguistic resources.

Table 1 displays the rules applied for each macro and the text resulting after its adaptation to the macro. Some rules are very simple, such as those for *Am I The Only One* and *X, X Everywhere*, which are based on Portuguese trends in Twitter and do not use a headline as input. All the other rules require a linguistic processing of the headline and may rely on the occurrence of specific tokens (e.g. *One Does Not Simply*, *Not Sure If*), linguistic constructions (e.g. *Brace Yourselves*, *Condescending Wonka*), or sentiment-related features (e.g. *Success Kid*, *Bad Luck Brian*).

⁸<http://natura.di.uminho.pt/~jj/pln/proverbio.dic>

⁹<http://knowyourmeme.com/>

Besides those in the table, two macros are used as a fallback, in case not a single headline is paired with a macro.

- For *Matrix Morpheus*, the system looks for proverbs using the most relevant word of the text to add after “*E se eu te disser que*” (What if I told you). If more than one proverb mentions the word, the most semantically-similar with the headline is used.
- *Wise Confucius* is applied to headlines without a matching proverb and can be seen as an application of lexical replacement humor. It first selects a proverb that rhymes with the most relevant headline word, possibly computing the semantic similarity to solve ties. The last word of the proverb is then replaced by the headline word. The proverb is added after the text “*Provérbio Chinês:*” (Chinese Proverb).

The previous macros have less restrictive rules and are thus applicable to most pieces of text. The result might be more surprising than for the previous macros but, despite the computed similarity, it may also be non-sense.

Results

Figure 2 shows the results of MEMEGERA 2.0 with a selection of examples, originally posted on Twitter. For each, we present the original headline, in Portuguese, followed by an English translation. Behind the title, the meme is displayed, followed by a rough translation of its text, with the name of the macro in bold. When the headline text remains unchanged, only the name of the macro is displayed.

Evaluation

To have an appreciation of the produced memes, an evaluation survey was conducted in two stages. First, from a set of collected news headlines, a random selection was made. The same headlines were shown to three humans, familiar with the concept of Internet Meme, but not aware of MEMEGERA. Each human was asked to select a suitable macro for each headline, out of those supported by our system, and to write a suitable text for a related meme.

After that, a survey was created with the nine headlines and the four produced memes – one by MEMEGERA and three by humans – presented in a random order. For each meme, the following four features were to be classified with a Likert scale – *strongly agree* (5), *partially agree* (4), *neutral* (3), *partially disagree* (2) and *strongly disagree* (1):

1. Coherence: the text is syntactically and semantically coherent.
2. Suitability: the macro and text are suitable for the headline.
3. Surprise: the result is surprising.
4. Humour: the result produces a humorous effect.

We soon noticed that the surveys were too long, and divided the original survey into three parts, each with three of the original nine headlines and three memes for each – one of the human-created memes was randomly discarded. Volunteers were then asked to answer the survey online, through a web page that would randomly redirect them to one of the three parts. In the end, responses were given by 52 different subjects, without any special control, except that they were

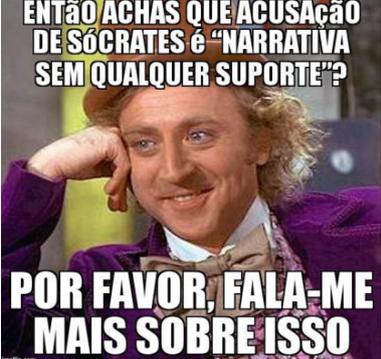
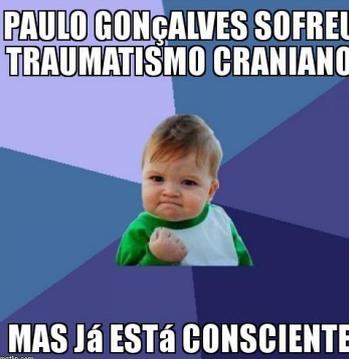
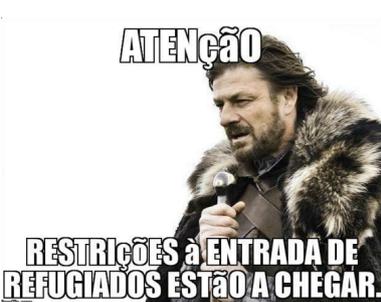
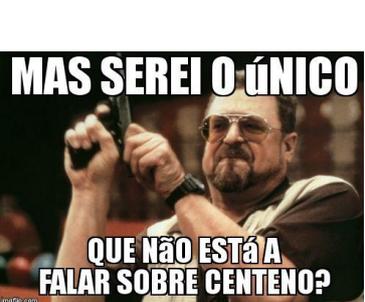
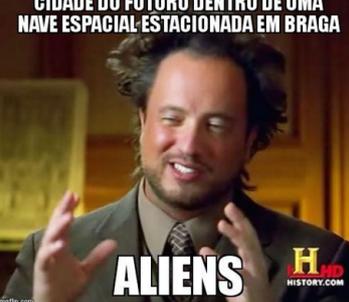
<p>França: Sarkozy adverte políticos para não esquecerem primeira volta. (France: Sarkozy warns politicians not to forget the first round)</p>	<p>Riade, Moscovo, Caracas e Doha acordam congelar produção de petróleo (Riyadh, Moscow, Caracas and Doha agree to freeze oil production)</p>	<p>“Maduro vai entregar um milhão de casas, ou corta o bigode” (Maduro will provide one million homes, or he will cut his mustache)</p>
		
<p>One Does Not Simply forget the first round</p>	<p>Chinese proverb: money was made to be frozen (Wise Confucious)</p>	<p>Not sure if provide a one million homes, or if I cut my mustache (Futurama Fry)</p>
<p>Magistrados dizem que acusação de Sócrates é “narrativa sem qualquer suporte” (Judges say that Sócrates’ indictment is an unsupported narrative)</p>	<p>Erdogan ganhou mas perdeu. (Erdogan won but lost)</p>	<p>Paulo Gonçalves sofreu traumatismo craniano mas já está consciente (Paulo Gonçalves suffered head trauma but is already aware)</p>
		
<p>So you think that Sócrates’ indictment is an unsupported narrative? Please, tell me more about it (Condescending Wonka)</p>	<p>(Bad Luck Brian)</p>	<p>(Success Kid)</p>
<p>Merkel anuncia restrições à entrada de refugiados (Merkel announces restrictions on the arrival of refugees)</p>	<p>#Centeno</p>	<p>Cidade do Futuro dentro de uma nave espacial estacionada em Braga (Future city inside a spaceship parked in Braga)</p>
		
<p>Brace Yourselfs restrictions on the arrival of refugees are coming</p>	<p>Am I the Only One not talking about Centeno?</p>	<p>(Ancient Aliens)</p>

Figure 2: Examples of produced and published memes of different types.

Macro	Trigger (in headline h)	Resulting text
<i>Brace Yourselves</i>	h mentions an announcement, expressed by verbs in the present or future, e.g.: X <i>preparar/plenear/projectar/anunciar</i> Y	<i>Preparam-se/Acautelem-se/Atenção ... Y (está a chegar)</i>
<i>One Does Not Simply</i>	h refers to an unfinished action, expressed by the adverb <i>não</i> (no) followed by a verb v , possibly followed by additional text and a preposition <i>prp</i> (<i>a, para, ...</i>), e.g.: X <i>não v (... prp)* Y</i>	<i>Simplesmente não se ... v Y / Y</i>
<i>Not Sure If</i>	h contains the alternative conjunction <i>ou</i> (or) opposing two ideas, e.g.: ... X <i>ou</i> Y ...	<i>Não sei se X ... ou Y.</i>
<i>Success Kid</i>	h either: expresses a highly positive sentiment with at least three positive words; has a negative phrase ($P-$) followed by an adversative conjunction c (e.g. <i>mas</i> , but) and a positive phrase ($P+$)	$h/P- ... c P+$.
<i>Sad Keanu</i>	h is highly negative because it has at least three negative words.	h
<i>Bad Luck Brian</i>	h has a positive phrase ($P+$) followed by an adversative conjunction c (e.g. <i>mas</i> , but) and a negative phrase ($P-$)	$P+ ... c P-$
<i>Condescending Wonka</i>	h mentions someone's opinion or belief by the linguistic constructions: X <i>dizer/achar/acreditar/pensar que* Y</i>	<i>Então achas que Y? ... Por favor, fala-me mais sobre isso</i>
<i>Ancient Aliens</i>	h contains words related to the outer space domain (e.g. <i>NASA</i> , planet names, <i>extraterrestre</i> , <i>ovni</i> , <i>astronauta</i> , <i>espacial</i> , ...)	$h ... Aliens$
<i>Money Money</i>	h mentions large amounts of money through expressions such as: <i>milhão de euros/dólares</i> (million of euros/dollars)	h
<i>Am I The Only One?</i>	Twitter trend T	<i>Mas serei o único ... que não está a falar sobre T?</i>
<i>X, X Everywhere</i>	Twitter trend T	$T ... fala-se sobre T em todo lado$

Table 1: Covered image macros, their triggers and resulting text.

not the creators of the survey memes. We believe that most were not aware of the existence of MEMEGERA and those that were did not know much about its internal operations.

Figure 3 shows the median of all the responses given for the memes produced by MEMEGERA and by each human creator. Results show that MEMEGERA could not beat any human. On the linguistic coherence, though with a median of 4 (partially agree), it is one level below the humans, which reflects occasional errors that may occur in the text adaptation. On the remaining features, the median is always 3 (neutral), below all the humans on the humor feature, below two humans on suitability and the same as all humans in the surprise feature. This confirms that the generation of humor is a challenging task, also for humans, who did not get the maximum scores either, but especially for machines. Although based on a fixed set of rules, the surprise feature is comparable to the human-created memes. Here, MEMEGERA probably benefited not only from the surprise of the fallback memes, but also from the fact that its Twitterbot was not followed by the subjects. Otherwise, some memes would probably become less surprising, because their macro ends up being used several times with different headlines. To give a better picture of how responses spread, table 2 shows the number of memes assessed for each creator (#), followed by the means and standard deviation of the survey responses.

	#	Coherence	Suitability	Surprise	Humor
MEMEGERA	156	3.81±1.44	2.98±1.44	3.06±1.24	3.10±1.37
Human 1	133	4.14±1.22	3.17±1.55	3.27±1.27	3.29±1.44
Human 2	67	3.87±1.42	2.82±1.47	3.07±1.29	3.12±1.44
Human 3	112	3.95±1.38	3.40±1.50	3.24±1.32	3.16±1.57

Table 2: Means and standard deviation of survey responses.

Figure 4 depicts MEMEGERA's meme with the highest average scores in the survey, after the two human-created memes for the same headline. We highlight the high median on the surprise and humor features (4). Figure 5 depicts

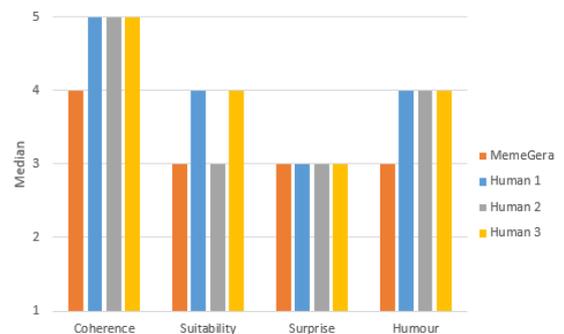


Figure 3: Median of the survey responses for MEMEGERA and for the human-generated memes.

the human-created meme with the best scores on average, followed by a meme by another human and MEMEGERA's. Although MEMEGERA's meme used the same macro as the first human-created, it is not as funny and the text is not even positioned with a top and a bottom part, as usual in these macros. Finally, figure 6 depicts MEMEGERA's meme with lowest scores, after the two human-created memes for the same headline. Once again, one human selected the same macro but, among other issues, MEMEGERA's had low coherence scores, to which a concordance mistake (*tenho* instead of *tem*) has contributed. This was already fixed.

Concluding remarks

We have presented a novel approach to the automatic generation of Internet memes based on news headlines, for which a suitable image macro is selected, with the help of a rule-based classifier that relies on linguistic triggers. The headline, possibly adapted according to the specific style of the macro, is added to the image.

Although focused on a short list of handcrafted linguistic rules, our main goal is achieved, as the produced artefacts

Headline: Acidente faz nove feridos e condiciona trânsito no IC2 (Accident causes nine wounded people and conditions traffic on the IC2 road)		
Human 1	Human 2	MEMEGERA
		
(Caused an accident... now the traffic is conditioned)	(Accident causes nine wounded people... and conditions traffic on the IC2 road)	(What if I told you that... among the dead and the wounded, someone will survive)
coherence = 5; suitability = 3; surprise = 4; humor = 4	coherence = 5; suitability = 1; surprise = 3; humor = 2	coherence = 4; suitability = 4; surprise = 4; humor = 4

Figure 4: Best MEMEGERA's meme (on average) after the two human-created for the same headline.

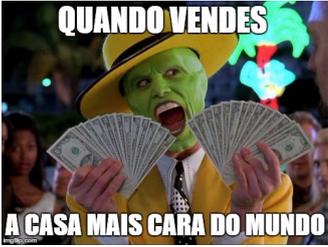
Headline: Casa mais cara do mundo foi vendida em Paris por 275 milhões de euros (World's most expensive house sold in Paris for 275 million euros)		
Human 1	Human 2	MEMEGERA
		
(When you sell... the most expensive house in the World)	(You sell house for 275 million? You homeless!)	(World's most expensive house sold in Paris for 275 million euros)
coherence = 5; suitability = 5; surprise = 4; humor = 4.5	coherence = 2.5; suitability = 3.5; surprise = 4; humor = 3	coherence = 5; suitability = 4; surprise = 3.5; humor = 3.5

Figure 5: Best classified human-created meme, the other human meme for the same headline, and MEMEGERA's.

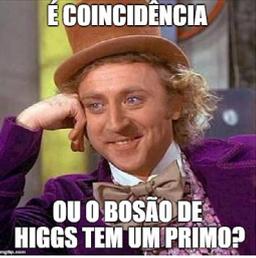
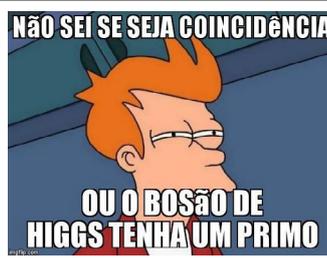
Headline: É coincidência ou o bóson de Higgs tem um primo? (Is it a coincidence or Higgs boson has a cousin?)		
Human 1	Human 2	MEMEGERA
		
(Is it a coincidence ... or Higgs boson has a cousin?)	(What if gravity does not come from Higgs ... but from its ninja cousin?)	(Not sure if it is a coincidence... or Higgs boson has a cousin)
coherence = 5; suitability = 2; surprise = 3; humor = 4	coherence = 5; suitability = 4; surprise = 3; humor = 3	coherence = 1; suitability = 3; surprise = 3; humor = 2

Figure 6: The worst of MEMEGERA's memes, after the two human-created for the same headline.

are easily recognisable as memes. Another strong aspect of this work is the integration of different available tools and resources which enabled us to go further. Current implementation targets Portuguese and uses a variety of natural language processing resources for this language, as well as Web APIs for collecting news, trends, producing the memes and posting a new meme, every hour, on Twitter. Despite other issues, the Twitterbot can be used for an alternative and funnier way of following recent news with a novel creative headline.

The first impression on the results is positive. They show coherence and are related to the headline. Yet, a comparison with human-created memes MEMEGERA 2.0 shows that there is still a long way to go, especially on producing actual humor. In fact, much humor value of the produced memes lies on the macros and the meaning they already carry.

Another limitation is the short range of covered macros and the closed set of rules. We admit that, after following the Twitterbot for a few days, one may get tired of the most frequently selected macros. Although we can add more macros, as we recently did, this opens up the discussion on whether MEMEGERA 2.0 is creative or not. Points for include the output, typically a product of human creativity, as well as the (creative) combination of different sources of knowledge for producing something new, but familiar. On the other hand, the selection of a macro is (almost) deterministic and, with the exception of the fallback memes, not that surprising, at least for frequent followers. Besides supporting more macros, in the future, variations of the current text transformations will be added, as well as refinements to the classifier towards making better-supported decisions. For instance, instead of relying on a binary classification – the headline suits the macro or not – the new classifier will consider additional features to score the headline-macro pair, such as the number of specific expressions (e.g. uncertainty-related for *Not Sure If*, difficulty-related for *One Does Not Simply*, sentiment words for *Success Kid* and *Sad Keanu*, or mysterious for *Aliens*). Moreover, given that most of the memes are commonly used with English text, it would definitely be interesting to adapt MEMEGERA to this language.

Acknowledgments

This work was supported by the project ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

- Binsted, K., and Ritchie, G. 1994. An implemented model of punning riddles. In *Procs 12th National Conf. on AI*, volume 1 of AAAI '94, 633–638. Menlo Park, CA, USA: AAAI Press.
- Costa, D.; Gonçalo Oliveira, H.; and Pinto, A. 2015. “In reality there are as many religions as there are papers” – First Steps Towards the Generation of Internet Memes. In *Procs 6th International Conference on Computational Creativity*, ICCV 2015, 300–307.
- Davison, P. 2012. The language of internet memes. In Mandiberg, M., ed., *The Social Media Reader*. NYU Press. 120–134.
- Dawkins, R. 1976. *The Selfish Gene*. Oxford University Press, Oxford, UK.
- de Paiva, V.; Real, L.; Rademaker, A.; and de Melo, G. 2014. NomLex-PT: A lexicon of Portuguese nominalizations. In *Procs. 9th Intl. Conf. on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: ELRA.
- Gatti, L.; Özbal, G.; Guerini, M.; Stock, O.; and Strapparava, C. 2015. Slogans are not forever: Adapting linguistic expressions to the news. In *Procs 24th International Joint Conference on Artificial Intelligence, IJCAI 2015*, 2452–2458. AAAI Press.
- Hempelmann, C. F., and Samson, A. C. 2008. Cartoons: drawn jokes? In *A Primer of Humor Research*. De Gruyter Mouton. 609–640.
- Raskin, V., ed. 2008. *The Primer of Humor Research*. De Gruyter Mouton.
- Rodrigues, R.; Gonçalo Oliveira, H.; and Gomes, P. 2014. LemPORT: a high-accuracy cross-platform lemmatizer for portuguese. In *Procs. 3rd Symp. on Languages, Applications and Technologies (SLATE 2014), Bragança, Portugal, OASICS*, 267–274. Schloss Dagstuhl.
- Santos, D., and Bick, E. 2000. Providing Internet access to Portuguese corpora: the AC/DC project. In *Procs 2nd Intl. Conf. on Language Resources and Evaluation, LREC 2000*, 205–210.
- Silva, M. J.; Carvalho, P.; and Sarmiento, L. 2012. Building a sentiment lexicon for social judgement mining. In *Procs. 10th Intl. Conf. on Computational Processing of the Portuguese Language (PROPOR 2012)*, volume 7243 of LNCS, 218–228. Coimbra, Portugal: Springer.
- Sjöbergh, J., and Araki, K. 2007. Automatically creating word-play jokes in Japanese. In *Procs. of NL-178*, 91–95.
- Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A neologism creator nomen ipsum. In *Procs 5th International Conference on Computational Creativity, ICCV 2014*.
- Stock, O., and Strapparava, C. 2005. The act of creating humorous acronyms. *Applied AI* 19(2):137–151.
- Toivanen, J.; Gross, O.; and Toivonen, H. 2014. The officer is taller than you, who race yourself! In *Procs 5th International Conference on Computational Creativity, ICCV 2014*.
- Turney, P. D. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Procs. 12th European Conf. on Machine Learning, ECML 2001*, volume 2167 of LNCS, 491–502. Freiburg, Germany: Springer.
- Valitutti, A.; Toivonen, H.; Doucet, A.; and Toivanen, J. M. 2013. “Let everything turn well in your wife”: Generation of adult humor using lexical constraints. In *Procs 51st Annual Meeting of the Assoc. for Computational Linguistics*, volume 2, 243–248. Sofia, Bulgaria: ACL Press.
- Veale, T., and Hao, Y. 2008. A fluid knowledge representation for understanding and generating creative metaphors. In *Procs. 22nd Intl. Conf. on Computational Linguistics*, volume 1 of COLING '08, 945–952. ACL Press.
- Veale, T.; Valitutti, A.; and Li, G. 2015. Twitter: The best of bot worlds for automated wit. In *Procs 3rd Intl. Conf. on Distributed, Ambient, and Pervasive Interactions, DAPI 2015*, 689–699.
- Veale, T. 2012. *Exploding The Creativity Myth: The Computational Foundations of Linguistic Creativity*. Bloomsbury Publishing.
- Wen, M.; Baym, N.; Tamuz, O.; Teevan, J.; Dumais, S.; and Kalai, A. 2015. OMG UR funny! Computer-aided humor with an application to chat. In *Procs 6th International Conference on Computational Creativity, ICCV 2015*, 86–93.

Poetry from Concept Maps – Yet Another Adaptation of PoeTryMe’s Flexible Architecture

Hugo Gonalo Oliveira¹, Ana Oliveira Alves²

¹ CISUC, Department of Informatics Engineering, University of Coimbra, Portugal

² CISUC, Polytechnic Institute of Coimbra, Portugal

hroliv@dei.uc.pt, ana@dei.uc.pt

Abstract

This paper describes a preliminary effort on adapting an existing poetry generation platform, PoeTryMe, to produce poetry from concept maps, extracted from textual documents. Instead of a set of given words that would constrain a general language semantic network, the presented adaptation dynamically sets the semantic network to the given concept map. As the relations in the concept maps are open, a new generation grammar had also to be created. Besides an architectural overview of the system, this paper is illustrated with several generated poems, together with the maps that originated them. Still, although poetic features are present and the content of the maps is reflected, they do not transmit exactly the meaning of the original document, due both to limitations on the grammars and issues on the quality of the maps.

Introduction

Computational approaches to linguistic creativity include the generation of narratives (Gervás et al., 2005), verbally-expressed humor (Binsted and Ritchie, 1994), or poetry, among others. In the last years, we have seen the birth of a diversity of poetry generation systems and approaches, driven by the advances on natural language processing and generation tools and on the huge amounts of textual data currently available.

Poetic text is typically recognised by the usage of certain features, such as a regular metre, rhymes or the presence of figurative language. To achieve such results, which should additionally be interpreted under a certain domain, current systems deal with several sources of knowledge, including natural language processing tools, lexicons, semantic knowledge-bases, or data extracted from human-created poems, among others. Available knowledge is exploited by a variety of approaches, frequently driven by some stimuli, which can be in form of a set of seed words, a short phrase, or a textual document. Although a minority of the efforts reported in the literature have some architectural concerns and could potentially be adapted to different situations, most of them are tailored for a specific purpose and end up having a reduced scope.

On the other hand, PoeTryMe (Gonalo Oliveira and Cardoso, 2015) is a generic platform for poetry generation, with a modular architecture that enables different instantiations and its reimplementations as poetry generation sys-

tems with different purposes. The flexibility of PoeTryMe’s architecture is confirmed by its adaptation to generate poetry in different languages (Gonalo Oliveira et al., 2014), poetry inspired by Twitter trends (Gonalo Oliveira, 2016), or song lyrics (Gonalo Oliveira, 2015). This paper reports on yet another effort to use PoeTryMe’s architecture, this time to produce poetry from the content of prose documents. For this purpose, PoeTryMe uses the output of TextStorm (Oliveira, Pereira, and Cardoso, 2001), an Open Information Extraction tool for acquiring concept maps automatically from textual documents. In opposition to other adaptations of PoeTryMe, which involved varying the poem structures, using different line templates, or plugging in different language resources, this new adaptation is not based on a set of seed words and involves changing the underlying semantics for the poem dynamically. Instead of a general language semantic network, current generation relies only on the semantic predicates extracted by TextStorm for the given document.

We begin with a reference to related work, namely poetry generation systems, focused on those efforts that reuse an existing model or architecture and those that produce poetry based on a prose document. After that, PoeTryMe and its architecture are briefly addressed, which is followed by a short description of TextStorm, the system used for extracting concept maps from text. The effort involved in the integration of the previous systems, including the acquisition of line templates for the TextStorm predicates, is then described. Before concluding, illustrative examples of poems, their seed maps and some of the original documents are presented and their quality is briefly discussed. While this work confirms the flexibility of PoeTryMe’s architecture, we are not fully satisfied with the achievements regarding the generation of poetry from a textual document. We see the reported work as a preliminary step to achieve this goal in the future, following some of the lines remarked in the last section.

Related Work

A variety of paradigms has been applied to automatic production of text with poetic features, including case-based reasoning (Gervás, 2001), evolutionary algorithms (Manurung, Ritchie, and Thompson, 2012), constraint programming (Toivanen, Järvisalo, and Toivonen, 2013), or multi-agent systems (Misztal and Indurkha, 2014). Several po-

etry generation systems are based on poem or line templates, but most of them go further and combine the previous with other techniques (e.g. Colton, Goodwin, and Veale (2012); Toivanen, Järvisalo, and Toivonen (2013)). Produced word sequences usually evolve to meet the desired constraints at different levels, such as form (lines, stress pattern), rhymes, syntax, and semantics. On the semantic level, the choice of relevant words may be achieved either with the help of semantic knowledge bases (Agirrezabal et al., 2013), by exploring models of word associations extracted from corpora (Netzer et al., 2009; Toivanen, Järvisalo, and Toivonen, 2013), or both (Colton, Goodwin, and Veale, 2012). Generation can be driven by a given stimuli, which can be in the form of a prose message (Gervás, 2001), a theme (Toivanen, Järvisalo, and Toivonen, 2013), or a set of seed words (Netzer et al., 2009), which constrain the poem search space and set the semantic domain.

More recently, systems that produce poetry based on a textual document have also been presented. Colton, Goodwin, and Veale (2012) analyse news articles to set the mood of the day and then select an article and a poem template to produce a new poem. Replacement words are selected through a rich process, based on a collection of similes, that considers features like aesthetics, lyricism, sentiment and flamboyancy. Misztal and Indurkha (2014)'s system is inspired by an input text, analysed to set the theme and the mood of the poem. Then, a set of artificial experts suggest related words, organise them into phrases, possibly exploring figurative language, and select the best phrases based on form constraints. Toivanen, Gross, and Toivonen (2014) produce poems inspired by news stories, from which novel word associations are extracted, when compared to long-established associations, such as those in Wikipedia. The resulting poem is obtained by replacing content words in an existing poem with the acquired associations. Tobing and Manurung (2015) rely on a dependency parser to extract the predicate-argument structure of a document, which it tries to keep during the generation of a poem, where additional constraints on form are considered. Generated poems explicitly capture the meaning of the input document but, from a computational point of view, dealing with these together with the poetic constraints seems to be impractical.

Whether or not they are reusable or adaptable to other situations, the previous systems have been presented as a specific instantiation, with a specific workflow, or even tested for producing a specific kind of poetry. An exception is the architecture of Colton, Goodwin, and Veale (2012) and the constraint satisfaction approach of Toivanen, Järvisalo, and Toivonen (2013), which happen to be combined in Rashel and Manurung (2014)'s system, even though the latter targets a different language.

Gervás (2015) argues that abstractions of the various functionalities involved in a poetry generation system should be available as services that may be invoked by other systems. This would allow the development of different systems that would, nevertheless, share some of their modules. Among other benefits, this would ease the development process and ease, for instance, the evaluation or the impact of adding new components. In ConCreTeFlows (Žnidaršič

et al., 2016), several widgets, including PoeTryMe and TextStorm, are available as independent processes, which can (and have) been manipulated to create novel and creative workflows. A similar platform is FloWr (Charnley, Colton, and Llano, 2014) which, among others, has been used to build poetry generation systems.

PoeTryMe and its Flexible Architecture

PoeTryMe (Gonçalo Oliveira and Cardoso, 2015) is a poetry generation platform developed since 2009 at CISUC, University of Coimbra, Portugal. It relies on a modular architecture (see figure 1 further ahead) that enables the independent development of each module and provides a high level of customisation, depending on the needs of the system and ideas of the user or developer. Among other parameters, users may define the structure of the poem, the transmitted sentiment, the generation strategy, the semantic network to use and the rules for generating lines based on the available relations. Developers may reimplement some of the modules and reuse the others.

A Generation Strategy organises lines, such that they suit, as much as possible, the structure of a poetic form and exhibit certain features. A structure file sets the poem form with the number of stanzas, lines per stanza and of syllables per line. An instantiation of the Generation Strategy does not generate the lines, but exploits the Lines Generator module to retrieve natural language fragments, which might be used as such. Syllable-related features are assessed with the help of the Syllables Util. Given a word, this module may be used to divide it into syllables, to find its stress, or to extract its termination, useful to identify rhymes.

The Lines Generator produces natural language renderings of semantic relations with the help of: (i) a semantic network, managed by the Relations Manager, that connects words according to relation predicates; and (ii) a generation grammar, processed by the Grammar Processor, with textual templates that render fragments expressing semantic relations. The generation of a line is a three-step interaction:

1. A random relation instance, in the form of a *triplet* = (*word*₁, *predicate*, *word*₂), is retrieved from the semantic network. To constrain the space of possible generations, a set of seed words can be provided to the Relations Manager. This set defines the generation domain, represented by a subgraph of the main network that will contain all the triplets involving seed words, or indirectly connected, depending on a surprise factor.
2. A random rendering for the *triplet*'s predicate is retrieved from the grammar. There must be a direct mapping between the relation predicates, in the graph, and the rules' name, in the grammar. Besides terminal tokens, that will be present in the poem without change, rules have placeholders that indicate the position of the relation arguments (<arg1> and <arg2>).
3. The resulting rendering is returned after inserting the arguments of the *triplet* in specific placeholders of a rule for its predicate.

In addition to the previous modules, the Contextualizer explains why certain words were selected and what is their

connection to the seed words, as a list of triplets for each line. It can be used for debugging or evaluation purposes.

The flexibility of PoeTryMe’s architecture is confirmed by its adaptation to generate poetry in different forms, including song lyrics that suit a given rhythm (Gonçalo Oliveira, 2015), poetry inspired by Twitter trends (Gonçalo Oliveira, 2016) and, although originally developed for Portuguese, PoeTryMe has been adapted to other languages, including Spanish (Gonçalo Oliveira et al., 2014) and, more recently, English.

TextStorm

TextStorm (Oliveira, Pereira, and Cardoso, 2001) is an Open Information Extraction tool also developed at CISUC, starting in 2001, with the original aim of creating concept maps for Clouds, an inductive learning tool, which would then be used as input to the Divago concept blending system Pereira (2007). Its original aim was to iteratively create concept maps from the Clouds, which could then be used as input to the Divago concept blending system Pereira (2007). TextStorm extracts conceptual relations from a textual document, written in natural language. Based on a Definite Clause Grammar, TextStorm extracts binary predicates from a text file, using syntactic and discourse knowledge, without requiring any previous knowledge on the document’s domain. The resulting set of predicates – represented in a common Prolog notation, *functor(Argument 1, Argument 2)* – constitute a graph where nodes are concepts and edges are relations between them, a knowledge representation format also known as Concept Maps (Novak, 1998).

TextStorm tags the input text file using WordNet (Fellbaum, 1998), and then builds predicates that map relations between two concepts, based on the parsed sentences. For instance, utterances like “*Cows, as well as rabbits, eat only vegetables, while humans eat also meat*” would result in the following predicates, that form a concept map:

- *eat(cow, vegetables)* • *eat(rabbit, vegetables)*
- *eat(human, vegetables)* • *eat(human, meat)*

Since, in natural language text, the same concept is not always referred by the same word or expression, TextStorm resorts to synonymy relations in WordNet to identify the concepts previously mentioned, even if through a different expression.

Poetry based on Concept Maps

This section reports on the effort of integrating the systems described earlier in a new instantiation of PoeTryMe, for producing poetry based on concept maps, extracted by TextStorm from a given textual document. The resulting architecture, with the relevant PoeTryMe modules, is depicted in figure 1.

The key of the present instantiation involved changing the semantic network dynamically, but ended up requiring the creation of a new grammar. No low-level changes were needed to the original Relations Manager and Grammar Processor modules.

All of the previous instantiations of PoeTryMe relied in a general language lexical-semantic knowledge base, where a subnetwork was selected, based on the given seeds. Although it could be quite large, the knowledge base was closed, both in terms of covered lexical items and relations. This meant that the poems would only be able to use seeds covered by the knowledge base, and related words, which had to be related by one of the covered predicates (e.g. synonymy, antonymy, hypernymy and meronymy). But TextStorm extracts an open set of predicates, most of them not covered by our previous grammars.

In our integration solution, a new grammar had to be created, in order to cover as many TextStorm predicates as possible. For that purpose, a large set of concept maps had to be extracted, hopefully covering a varied set of predicates. Those maps were then used as the knowledge base for the acquisition of the new grammar. Although they can be handcrafted, previous grammars of PoeTryMe have been acquired automatically from given textual lines (e.g. of human-created poems) where two related words occur, then generalised as possible renderings of their relation. For instance, for a semantic network with the relations:

- *abstraction* hypernym-of *poem*
- *flower* hypernym-of *dahlia*
- *animal* hypernym-of *cat*

Lines such as:

- “*the poem itself is a kind of abstraction*”,
- “*abstraction, in the form of a poem*”,

Could be generalised to originate the following rules:

- *hypernym-of* → *the <arg2> itself is a kind of < arg1 >*
- *hypernym-of* → *<arg1> in the form of a < arg2 >*

During the generation of a poem, this rule may result in variations, such as:

- *the dahlia itself is a kind of flower*
- *animal, in the form of cat*

In order to obtain large quantities of text, written as directly as possible, we resorted to a selection of articles from the Simple Wikipedia¹, then processed by TextStorm. The selection of this resource relied on its wide-coverage of different topics, described using basic English vocabulary and shorter sentences, which would hopefully improve the extraction of concept maps. The grammars were not extracted from the Simple Wikipedia though. The extracted concept maps were only used as a knowledge base for the later acquisition of relation renderings from any other text collection. Figure 2 illustrates the grammar extraction procedure.

The new instantiation of PoeTryMe works as follows:

1. A textual document is provided to TextStorm, which extracts a concept map;
2. The resulting concept map is converted to PoeTryMe’s notation – which instead of *predicate(arg1, arg2)* becomes *arg1 PREDICATE arg2*;

¹<https://simple.wikipedia.org>

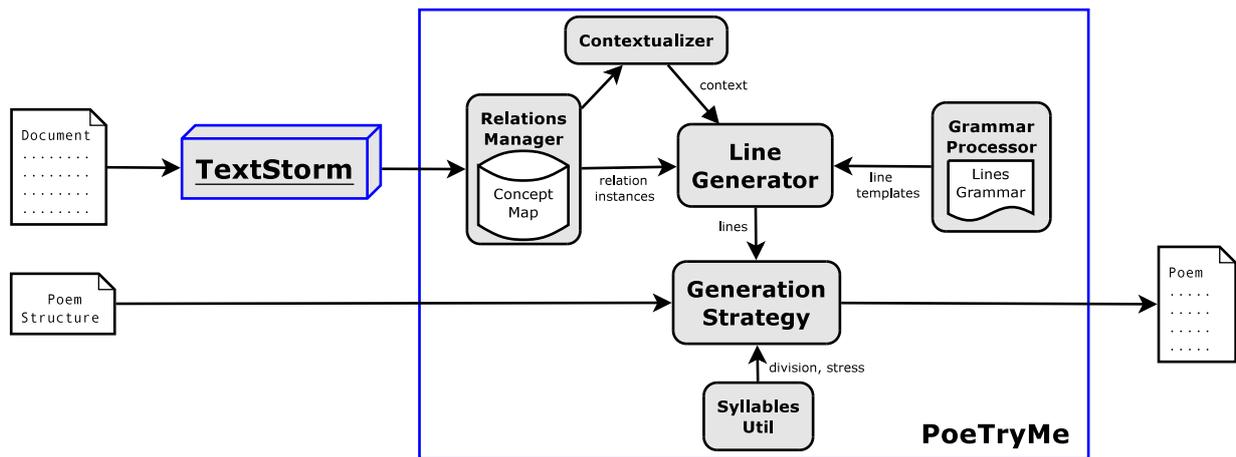


Figure 1: High level diagram of the resulting architecture (TextStorm + PoeTryMe)

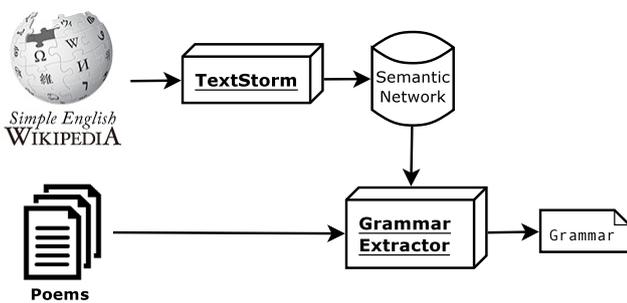


Figure 2: Grammar extraction procedure

3. PoeTryMe is run for a given poetry form, using the concept map as input, instead of a set of seed words;
 - (a) The concept map is dynamically set as the semantic network to use;
 - (b) Lines are produced by the Line Generator, using all the relations of the concept map;
 - (c) Most suitable lines are selected according to a generate & test strategy, the same of previous instantiations of PoeTryMe.
4. The output is a poem in the target form, where each line is a rendering of a relation instance in the concept map.

Moreover, since TextStorm only supports English, this work relied on a previous adaptation of PoeTryMe to English, following similar lines as the Spanish adaptation (Gonçalo Oliveira et al., 2014). A major difference of the current adaptation is related to metric scansion, which is more complex for English than for Portuguese and Spanish. While, for the latter, most cases were covered with a rule-based approach, relying only on the orthography, for English, this would not work, because there are many different combinations of letters that are pronounced the same way (e.g. *eye* rhymes with *lie*, *apply* and *levi*; *air* rhymes with *aware* and *bear*). Therefore, in order to perform syllable division, stress and rhyme identification, we relied

on the CMU Pronouncing Dictionary², which contains over 134,000 words and their pronunciations in North American English. A new implementation of the Syllable Utils interface was developed to interact with this dictionary and perform the syllable-related operations on English words. For non-covered words, a fallback mechanism uses the Portuguese rules. In order to acquire the rules of the lines grammar, a collection of human created English poems was exploited. Those were obtained from the Representative Poetry Online (RPO), a web anthology of poetry by the University of Toronto Libraries³.

The rules of the new grammar were thus obtained from the lines of the previous poems where two related words co-occurred, according to the concept maps extracted from the Simple Wikipedia. Each of those lines had the related words replaced by the argument placeholders and resulted in a rule for the relation predicate. Using the concept maps obtained from 4,096 Simple Wikipedia articles, and about 3,400 human-created poems, the grammar used in this work had 2,289 rules and covered a total of 171 distinct predicates.

Results

To illustrate the results of the presented integration, several poems were produced. As mentioned earlier, instead of seed words, the full concept map was given as input and used as PoeTryMe's semantic network. The generation of lines followed a generate & test strategy at the line level, similar to that of previous instantiations (e.g. Gonçalo Oliveira et al. (2014)). For each line, up to $n = 2,000$ textual renderings of relations in the concept map were sequentially produced and tested against the target size and rhyme, while keeping the best one. To increase the probability of rhymes, an increasing factor $\sigma = 0.8$ was used, meaning that the number of renderings produced for the i^{th} line of a stanza were at most $n + n * (i - 1)$. For $n = 2,000$, this results in 2,000

²<http://svn.code.sf.net/p/cmuspinx/code/trunk/cmudict/>

³<http://rpo.library.utoronto.ca/timeline/>

renderings for the first line of a stanza, 3,600 for the second, 5,200 for the third and 6,800 for the fourth. In order to select the best lines, each syllable deviating from the target number lead to a penalty of 1 point, while each rhyme resulted in a bonus of 2 points.

Examples

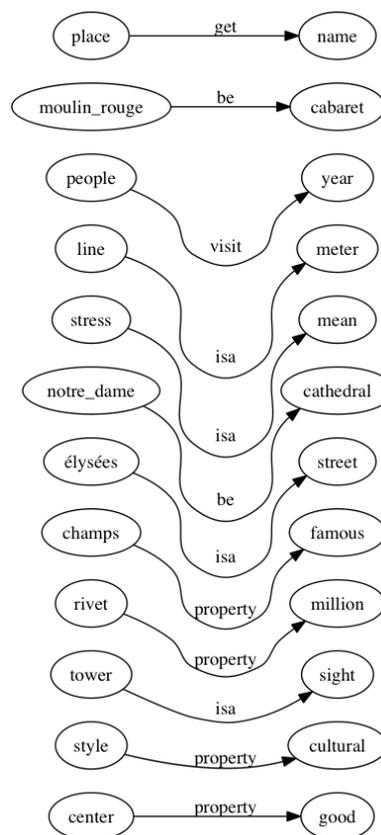
Figures 3 to 6 show (partial) concept maps on diverse subjects and selected generated poems. In addition to the map, figures 3 and 6 show part of the textual document that originated it, omitted in the remaining examples, due to lack of space. All but the example in figure 6 are based on Simple Wikipedia articles, which was our original source for extracting the generation grammars. The poems in figure 3 are blocks of four 10-syllable lines, based on the article on Artificial Intelligence. The following two examples are based on articles about named entities, more precisely, one shows blocks of four 10-syllable lines based on the city of Paris (figure 4), and the other a sonnet based on Alan Turing's article (figure 5).

The final example, in figure 6, is an attempt to go one step further and confirm that the integration of both systems enables the generation of poems from any given textual document. Its concept map is extracted from a news article, published in *The Atlantic* online newspaper on 2nd March 2016⁴.

Discussion

Similarly to those selected, generated poems frequently match the target size of a line and, when the size is different, the difference is rarely more than 1. We recall that, in the scoring system used, the bonus for rhymes is two times higher than the penalty for one additional syllable or one less. Rhymes are also frequent. Although we manually selected poems with rhymes in almost all the lines, if the semantic network and the generation grammar are large and varied enough, generating poems with many rhymes is just a matter of increasing n and σ . One issue regarding the form of the poems is the presence of a few syntactically-odd lines. This is partially due to our grammar acquisition procedure, where the part-of-speech (POS) of the arguments is not considered. As we know, most verbs can also be nouns (e.g. *break, cover*), and many nouns can behave as adjectives (e.g. *red, young*). In fact, TextStorm also extracts this kind of information for terms in WordNet, but it was not exploited in the current instantiation.

Despite the previous issue, we can say that the concept map is reflected on the produced poems. A minority of deviations occur due to the presence of fixed words in the template, especially open class words, which may sometimes be out of the desired context. This is a limitation of PoeTryMe, which could be minimised by handling the previous issue regarding the POS, creating rules with no more than two content words (to be replaced), or using more general sources of language as the source of the grammars. But if those are not



*why ask my tower? that old sight will swear
a name of weight; line little meter heir
thus the great people of almighty year
and élysées, and street shall disappear*

*moulin rouge and mother, cabaret bless
his stress, while the mean spirit's plastic stress
leave me to my cathedral notre dame
though from another place i take my name*

*famous of champs, and the better part choose
that which a good center only could refuse
the million sort by thir own rivet fell
once cultural, now in style, and to dwell*

Figure 4: Concept map of the Simple Wikipedia article on Paris and generated blocks of four 10-syllable lines.

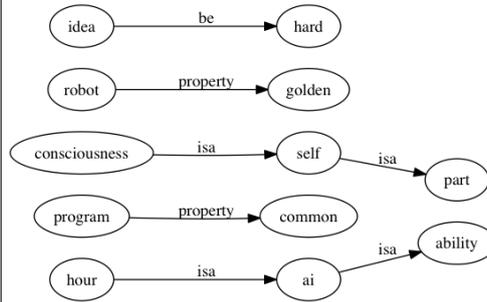
poems, this may have a negative impact on the poeticness of the generated text.

A bigger problem is the quantity and quality of the maps, which results in poems that cannot be said to be about the original documents. Some relevant relations are not captured, and some others are not exactly what should have been extracted. This happens because TextStorm does not handle elaborate sentences very well, especially those with implicit co-references. In fact, it does not perform two important natural language processing tasks: anaphora resolution and named entity recognition. Not resolving anaphoras results in some relations held between pronouns (e.g. *he*,

⁴<http://www.theatlantic.com/politics/archive/2016/03/donald-trump-the-protector/471837/>

Artificial intelligence (AI) is the ability of a computer to think like a human (or eventually better) – to be able to learn and to have “new ideas”. ... For example, a **common computer program** can turn a report of names and hours worked into paychecks for the workers at a company. ... That is the difference between a program and AI. ... In some cases, AI can be simulated (imitated), at least in certain areas. ... The question of what it means **to be self-aware** or having **consciousness** (knowing that you have a physical body, and how you think about your self) **is part** of it. ...

The idea of thinking machines had been around before this. In 1950 Alan Turing wrote a paper called “Computing Machinery and Intelligence”. He started with the question “Can machines think?”. Since “thinking” is hard to define, Turing went to another question instead – “Can machines trick a human into thinking they are talking to another human (instead of to a machine)?”. Even earlier, thinking machines and artificial beings appear in Greek myths, such as Talos of Crete, the **golden robots** of Hephaestus and Pygmalion’s Galatea.



*through all the common green programs has spread
golden robots and nights he has lain abed
let part since self can little more supply
packs, picking her abilities, fleece, ai*

*there lived a program once, a common bard
upon these ideas, so wild and hard
the pungent commons and bright, program wings
ideas, hard gossip, oddments of all things*

*o aching self! o moments big as parts!
a hour on an island; such an ai
all sorts of programs, by my common arts
and let this consciousness of selves go by*

*and thou, my ai, aspire to higher hours
took marvellous robots; golden domes and towers
at consciousness self silent as the air
consciousness thing the hand of self shall spare*

Figure 3: Text of the Simple Wikipedia article on Artificial Intelligence, the extracted concept map, and selected blocks of four 10-syllable lines generated.

Donald Trump: The Protector

He will make you safe. He will give you health care. He will give you jobs. He will build a wall. Protecting you is his prime directive...

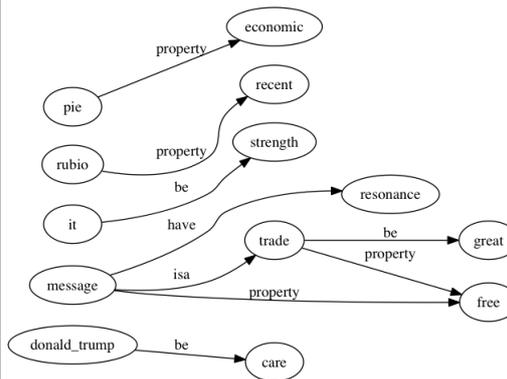
This **message has powerful resonance**, especially for voters who feel the Republican Party has failed to protect their interests...

Free trade is great, Trump says, but it has to be fair. His opponents just adhere to pure free trade, which does increase the **economic pie**.

But economic research shows that free trade harms some subsets of voters, particularly the working-class voters flocking to Trump. The message to his voters: I will favor free trade only to the extent that I can protect you from harm, perhaps by compensating you using the gains of trade. My opponents will favor free trade even if it harms you...

It is because, to his voters, these attacks have stressed what, to them, is **Trump’s strength**...

The **recent Rubio**-led attacks on Trump have been more telling because their nature is different...



*care she, donald trump, of these last
is rubio recent past
and all trade free from before them
like free blossoms on message stem*

*does this economic pie go?
trade seems message vain, fleeting show
no trade is good, no pleasure free
and trade of those message was me*

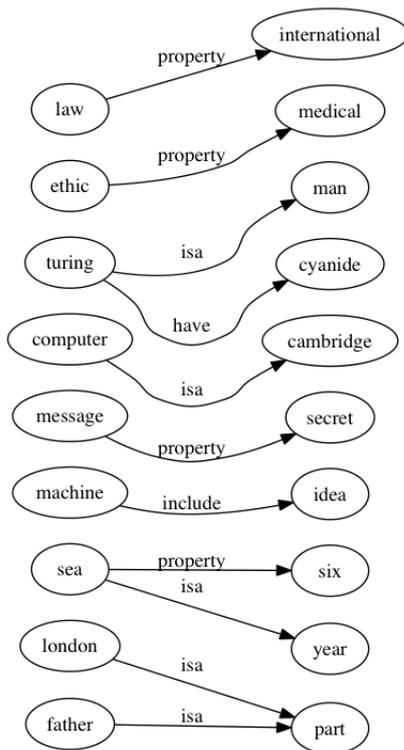
*how chill is a donald trump care
how great a part of trade they share
to get the free trade out of bed
resonance, is thy message dead?*

Figure 6: Part of the news on Donald Trump, part of their concept map, and selected generated blocks of four 8-syllable lines.

it), but this situation was minimised in the presented poems, with the application of a filter for relations with a stopword, in most generations. Not recognising named entities has negative consequences on documents about locations (e.g. Paris), people (e.g. Turing) or organisations. It is especially critical in news articles, where several named entities are generally mentioned. This is also why some named entities are only partially recognised and none of them is capitalised.

While the evaluation of poetry remains quite a subjective task, models have been proposed to evaluate the process of creative systems. The FACE descriptive model has been used for this purpose, and has been applied to poetry generation systems (Colton, Goodwin, and Veale, 2012; Misz-

tal and Indurkha, 2014). In order to be assessed positively by this model, a creative system must create a concept (C), with several examples (E), include an aesthetic measure (A) for evaluating the examples, and provide framing information (F) that will explain the context or motivation of the outputs. The combination of TextStorm or PoeTryMe covers the previous four criteria: concepts, represented as maps extracted from text, are expressed by different poems; lines are organized in poetic forms according to which metre and rhymes are assessed (aesthetics); and the selection of words and patterns are explained either by the Contextualizer module of PoeTryMe, or by the underlying concept map, where the relations used to create the poem are uncovered.



*the secret message in their waxen cells
has made for cyanide this two-penny turing
we international are waiting law
when thou and i six sea another saw*

*sole star of all that cambridge and computer
there but one law he doth make international
part of sin, london th' irrational
meek idea in the machine of christ!*

*what ethic i, how medical she be?
the ice-blue calm of a year sea
law, glossy green, and velvet international
man of sin, turing th' irrational*

*all the undone sea of the speeding year
and father, and part shall disappear*

Figure 5: Concept map of the Simple Wikipedia article on Alan Turing and selected generated sonnet.

Concluding Remarks

We have described the effort involved on the adaptation of a flexible architecture for poetry generation, PoeTryMe, this time with the purpose of producing poetry based on textual documents. To this end, concept maps are first extracted from a document, by another system, TextStorm. The resulting map is used as input for the generation of a poem that should transmit the same meaning. Generation also requires a grammar with textual renderings for most of the predicates that may be included in the concept maps.

The reported work confirms the flexibility of PoeTryMe's architecture with yet another adaptation, this time changing

the base semantics dynamically, given a textual document. We believe to have shown that the goal of generating poetry based on concept maps was achieved. Yet, although the poems are framed by the concept maps, they do not effectively transmit the meaning of the original document. Besides a few odd constructions, resulting from limitations on the grammar acquisition procedure, the quality of the maps could be better, and this has also a negative impact on the semantics of the poem.

We admit that we are not completely satisfied with the obtained results, and we are already working on future improvements. On the poetry generation side, the current size and the variety of the patterns in the grammar will be increased. The grammar should be acquired from a larger set of concept maps, possibly extracted from the full Simple Wikipedia, and on a larger set of documents, possibly including other kinds of text, and not just poems. Moreover, the grammar might cover more generic patterns, where words in previously unseen relations could still fit without changing the semantics, and the POS of the relation arguments should also be considered. TextStorm could be further explored for the latter purpose and also to augment the used vocabulary, using the synonyms it extracts from WordNet.

Work is being carried out to improve the quality of the TextStorm concept maps, we are also devising alternative relation extraction systems. A possible TextStorm improvement would be to train a shallow parser, instead of using a definitive cause grammar. But we are still unsure whether a regular Open Information Extraction system (e.g. ReVerb (Fader, Soderland, and Etzioni, 2011)) would be more suitable, because most of the extracted relation predicates are too long and thus too diverse to be useful without any kind of simplification. Although TextStorm also extracts an open set of predicates, they are typically shorter (a verb or a verb and a preposition), which is an advantage in this case. Moreover, in order to generate poetry based on certain entities, PoeTryMe could also be tested with other kinds of semantic networks, such as DBpedia. But that goal would be slightly different from that of producing a poem from a given textual document.

A limited version of PoeTryMe is available as a simple web application that communicates with PoeTryMe's REST API, in the TryMe section of <http://poetryme.dei.uc.pt/>. It enables the generation of a poem in one of the supported languages (Portuguese, Spanish, English), given an open set of seed words, a poem form (from a closed set), and a surprise factor. Yet, in this limited version, generation relies on fixed semantic networks and grammars for each language, and it is not possible to provide a different network nor grammar. Although both PoeTryMe and TextStorm have widgets in ConCreTeFlows, due to the previous limitation, the workflow reported here is still not possible to replicate there. Following Gervás (2015), in the future, we will devise decoupling each module of PoeTryMe as an independent web service, which will hopefully enable their exploitation by even more natural language generation systems.

Acknowledgments

This work was supported by the project ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733. We also acknowledge Pablo Gervás, Alberto Diaz and Raquel Hervás, who implemented the English version of the Syllable Utils interface, collected the human-created English poems, and were involved in the overall process of adapting PoeTryMe to Spanish and English.

References

- Agirrezabal, M.; Arrieta, B.; Astigarraga, A.; and Hulden, M. 2013. POS-Tag based poetry generation with wordnet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, 162–166. Sofia, Bulgaria: ACL Press.
- Binsted, K., and Ritchie, G. 1994. An implemented model of punning riddles. In *Proceedings of 12th National Conference on Artificial Intelligence (Vol. 1)*, AAAI '94, 633–638. Menlo Park, CA, USA: AAAI Press.
- Charnley, J.; Colton, S.; and Llano, M. T. 2014. The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the 5th International Conference on Computational Creativity*, ICCV 2014.
- Colton, S.; Goodwin, J.; and Veale, T. 2012. Full FACE poetry generation. In *Proceedings of 3rd International Conference on Computational Creativity, Dublin, Ireland*, ICCV 2012, 95–102.
- Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing*, EMNLP 2011. Edinburgh, Scotland, UK: ACL Press.
- Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.
- Gervás, P. 2001. An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems* 14:200–1.
- Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2005. Story plot generation based on CBR. *Knowledge-Based Systems* 18(4):235–242.
- Gervás, P. 2015. Deconstructing computer poets: Making selected processes available as services. *Computational Intelligence*.
- Gonçalo Oliveira, H., and Cardoso, A. 2015. Poetry generation with PoeTryMe. In Besold, T. R.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity Research: Towards Creative Machines*, Atlantis Thinking Machines. Atlantis-Springer. chapter 12, 243–266.
- Gonçalo Oliveira, H.; Hervás, R.; Díaz, A.; and Gervás, P. 2014. Adapting a generic platform for poetry generation to produce Spanish poems. In *Proceedings of 5th International Conference on Computational Creativity, Ljubljana, Slovenia*, ICCV 2014.
- Gonçalo Oliveira, H. 2015. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence* 6(1):87–110. Special Issue: Computational Creativity, Concept Invention, and General Intelligence.
- Gonçalo Oliveira, H. 2016. Automatic generation of poetry inspired by Twitter trends. In *Post-conference Proceedings of IC3K – Revised Selected Papers*, CCIS, in press. Springer.
- Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.
- Misztal, J., and Indurkha, B. 2014. Poetry generation system with an emotional personality. In *Proceedings of 5th International Conference on Computational Creativity*, ICCV 2014.
- Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku: generating haiku with word associations norms. In *Proceedings of the NAACL 2009 Workshop on Computational Approaches to Linguistic Creativity*, CALC '09, 32–39. Boulder, Colorado: ACL Press.
- Novak, J. 1998. *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. Mahwah, NJ: Lawrence Erlbaum, 1 edition. 2nd edition published in 2010.
- Oliveira, A.; Pereira, F. C.; and Cardoso, A. 2001. Automatic reading and learning from text. In *Proceedings of the International Symposium on Artificial Intelligence*, ISAI'2001, 69–72.
- Pereira, F. C. 2007. *Creativity and AI: A Conceptual Blending Approach*. Applications of Cognitive Linguistics (ACL). Mouton de Gruyter, Berlin.
- Rashel, F., and Manurung, R. 2014. Pemuisi: A constraint satisfaction-based generator of topical Indonesian poetry. In *Proceedings of 5th International Conference on Computational Creativity*, ICCV 2014.
- Tobing, B. C. L., and Manurung, R. 2015. A chart generation system for topical metrical poetry. In *Proceedings of the 6th International Conference on Computational Creativity, Park City, Utah, USA*, ICCV 2015.
- Toivanen, J. M.; Gross, O.; and Toivonen, H. 2014. The officer is taller than you, who race yourself! using document specific word associations in poetry generation. In *Proceedings of 5th International Conference on Computational Creativity*, ICCV 2014.
- Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the 4th International Conference on Computational Creativity*, ICCV 2013, 160–167. Sydney, Australia: The University of Sydney.
- Žnidaršič, M.; Cardoso, A.; Gervás, P.; Martins, P.; Hervás, R.; Alves, A. O.; Gonçalo Oliveira, H.; Xiao, P.; Linkola, S.; Toivonen, H.; Kranjc, J.; and Lavrač, N. 2016. Computational creativity infrastructure for online software composition: A conceptual blending use case. In *Proceedings of 7th International Conference on Computational Creativity*, ICCV 2016.

Analysis of the correlations between the knowledge structures of an automatic storyteller and its literary production

Iván Guerrero Román¹, Rafael Pérez y Pérez²

¹Posgrado en ciencia e ingeniería de la computación. Universidad Nacional Autónoma de México, D.F., México

^{1,2}División de ciencias de la comunicación y diseño. Universidad Autónoma Metropolitana, Cuajimalpa, D.F., México

¹cguerrero@uxmcc2.iimas.unam.mx, ²rperez@correo.cua.uam.mx

Abstract

In this paper, we describe a process to identify relations among the features of the knowledge base of an automatic storyteller and the narratives that it generates. We define structures to analyze the internal composition of the information available for an agent. We also establish a set of metrics to identify diverse story characteristics. Next, we perform experiments utilizing Mexica, an automatic storyteller, to generate narratives and to evaluate them according to our set of metrics. Then, we compare such assessments with the visual structures that we built from the agent's original knowledge base, in order to obtain correlations between them. The results suggest that such correlations are useful to study the links between the agents' knowledge base and the kind of stories they might produce.

Introduction

During the last 15 years, members of our research group have developed a wide variety of models related to storytelling, and we have implemented them in computational programs, or agents. Among these models there is an automatic storyteller, Mexica, and a collaborative story generator, Mexica-Impro; models for evaluating stories and for identifying social norms in the generated outputs. In all of them, the knowledge structures (KS) available in each of the agents have played an essential role. We have utilized emotional and tension links between the characters in a story to represent these KSs, and we have obtained this information from two major sources: a dictionary of story-actions, and a set of previous stories (narratives written by humans that are considered benchmarks for our models). Nevertheless, one pending task, tackled in this work, is the study of how features of the agents' knowledge base influence the narratives that they generate. The direct antecedents of this research arise from a three-fold base: automatic story generation and evaluation, and description of high-level structures emerging from the knowledge bases of our agents.

From the first text generation works in the early 60's (Klein 1965), to the latest storytellers such as Fabulist (Riedl 2004), Mexica (Pérez y Pérez 2001 and 2007) or Minstrel (Turner 1994), automatic narrative generation has intrigued researchers for decades in an attempt to better understand diverse aspects of this process. Despite the fact that they have descriptions of how internally represent their

knowledge, it is commonly missing how these structures affect the overall quality of the generated stories. Moreover, they lack of high-level representations of the available knowledge to identify emergent structures, and to analyze how these structures prevent unpleasant behaviors and promote desirable features in their outputs.

Regarding to the evaluation of the generated stories, Pérez y Pérez (2014) proposed a layered model describing how features such as opening, climax, closure... in a story, could be measured to determine how coherent, novel and interesting they are. In our work, we rely on these metrics and extend them to identify additional story features and structural elements of the agent's knowledge bases.

To identify high-level structures of the agent's knowledge, Pérez y Pérez (2015) describes contextual structures maps. They represent how the acquired wisdom of an agent is distributed throughout the space of all the possible structures, and identifies different types of elements according to their number of components. In this work, we build upon this idea to present alternative high-level structures to represent relations according to the similarity among the elements inside the KS of our storyteller.

We claim that if we are able to link previous stories with the agents' KSs, and find out how KSs' features influence the characteristics of the generated plots, we will improve our understanding about the importance of previous experiences for the plot generation process. In this way, our agents will be able to identify for example what type of knowledge is still missing in their repositories, and develop stories to explore specific topics with the purpose of filling these gaps.

In general, we review how Mexica, an automatic storyteller, builds its own knowledge structures, and we present a high-level structure which provides us additional information about the knowledge of the computer agent. Then, we present a set of metrics to describe features of stories generated by an agent implementing Mexica, and we also present features to describe the structure of its knowledge base. Finally, we identify relations between these two different types of features.

In this paper, we describe a methodology to visualize characteristics of the agents' KSs, referred to as connectivity maps (C-maps), to show the similarities among KSs in memory. Then, we illustrate how the topology of such maps affects several features of the computer generated plots.

Next, we present our findings about how these knowledge structures affect diverse features of stories generated by our agents. Finally, we reflect on these results and speculate on possible extensions to this work.

Gathering information

We rely on two main component types to identify the relations that we are looking for: a story generator and story evaluator, a computer program to assess the outputs of the generator. For the first, we utilized Mexica (Pérez y Pérez 1999, 2007, and Pérez y Pérez & Sharples 2001), and we extended it by incorporating Social Mexica (Guerrero and Pérez y Pérez 2014), a computer model for social norms in narratives to provide additional social information to the story generation process. For the second component, we extended a model for evaluating the interestingness of a narrative proposed by Pérez y Pérez (2014). We now describe each of these components.

Generating knowledge structures

Mexica is a storyteller based on the E-R creativity model (Pérez y Pérez 1999), which describes the creative activity of writing as an iterative two-phased process: engagement and reflection. During engagement the agent selects diverse actions to produce a partial story; whereas in reflection, the system evaluates and updates the material previously generated. Additionally, diverse guidelines to constrain the production of material during engagement are set according to the evaluations performed during this stage. These evaluations also serve to determine when a story is considered to be finished. If this is not the case, the system initiates a new engagement stage and the cycle starts all over again until the story is considered to be finished.

Mexica employs two information sources to generate a variety of knowledge structures utilized during the story generation process: a dictionary of story-actions, and a set of previous stories. Actions in the dictionary have associated a name, and a set of preconditions and post-conditions to represent their requirements and consequences when added to a story. These conditions are defined in terms of emotions (such as love or friendship) and tensions (such as life or health at risk, character prisoner...). Every story (either generated or previous) is defined as a sequence of instantiated actions. This occurs when characters (a performer and an optional receiver) are added.

'Virgin fell in love with Warrior', represents a valid instantiated action. Here, Virgin and Warrior represent the characters, and 'fell in love with' corresponds to the action phrase. Some of these actions consist of only one character, like 'Hunter went to the forest'.

We use contextual structures (CS) to represent the knowledge available for our agent. They are built from the previous stories to be further utilized during the generation of new narratives. Mexica internally transforms a story into emotional relations and tensions between characters, and from this representation, called story-context, CSs are extracted. They consist of two elements: a set of relations (emotions or tensions) between characters, and a list of desirable continuations. Figure 1 represents a story-context obtained from

the following story: 'Tlatoani (T) was father of the Princess (Ps)', then 'the priest (Pt) made Princess her prisoner'. The link from Tlatoani to Princess, represent a positive friendship relation with high intensity (+3); the link from Princess to Priest, represent a negative friendship relation (representing hate) with high intensity (-3); and the seesaw link from Priest to Princess, represent a tension between them ('Pr' represents the type of tension, prisoner).

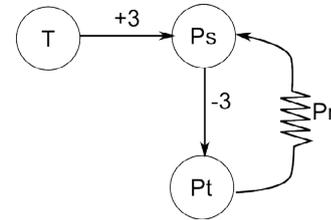
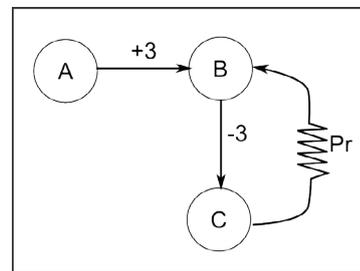


Figure 1: Visual representation of a story-context. Here, nodes represent characters and edges represent relations between characters. The lines with arrow heads represent emotions, whereas the seesaw lines represent tensions.

From the previously described context, Mexica extracts the context of the CS displayed in figure 2. Here, characters are replaced with variables (represented by the letters A, B and C), and the next action in the story is linked to represent a desirable continuation (in our example, the story continued with the action 'T rescued Ps'). A CS can have several actions linked to it. This occurs when identical story-contexts are obtained from different stories, and instead of generating two CS with the same context, we group them into one single CS with multiple actions.



Following action: 'A rescue B'

Figure 2: Visual representation of a contextual structure. The rectangle at the top represents a CS-context, and at the bottom is displayed a desirable action for this context.

To identify features related to these knowledge structures, we developed a map to obtain additional information regarding to their similarity. A connectivity map (C-map) represents CSs and relations among them. Every node in this map represents a CS, and two nodes are linked if they are similar enough. The agent determines such similarity by identifying the number of corresponding relations between two structures according to the following rules:

- One emotion is similar to another when they share the same type, valence (positive or negative), and the first has

an intensity lower or equal to the second.

- Tensions: Two tensions are similar when they share the same type.
- Once a similar emotion or tension is identified, the character variables of the nodes utilized in the relations are mapped and they cannot be utilized to identify similar relations creating new mappings.

In figure 3, we display a context similar to the one in figure 2. To determine the similarity of the second context with respect to the first, we look for emotions with the same type, valence, and with an intensity lower or equal; we then look for tensions with the same type. In this case, the emotional link between A' and B' in the first context is similar to the emotional link between B and C in the second context. This generates a mapping of the characters A' with B , and B' to C , preventing the generation of new mappings for the variables A' , B' , B and C . Next, the tension between A' and B' is similar to the tension between B and C , and preserves the original mappings. The only missing element is the emotional link between A and B in the second context. This results in a similarity value of 0.66 (two out of three similar links).

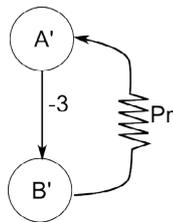


Figure 3: Visual representation of the context in a CS.

From this C-map, nodes are categorized into three different groups according to the number of connections among them. Due to the lack of similar studies, and after analyzing the values obtained, we empirically determined two threshold values of 5% and 10% to create our categories, but we will perform further studies to identify the implications of this values in our study.

- Isolated nodes: Those connected with less than 5% of the total number of nodes
- Regular nodes: Those connected with 5% to 10% of the total number of nodes
- Focal nodes: Those connected with more than 10% of the total number of nodes

When the nodes inside a C-map are linked, they form clusters of similar elements. According to their members, clusters are classified into three categories: islands, towns and cities. After analyzing the number of nodes inside the clusters, we determined two threshold values of 20% and 50% to classify them, but we will develop further studies to determine the implications of this values in our studies.

- Island: Contains less than 20% of the nodes inside the C-map

- Town: Contains between 20–50% of the nodes inside the C-map
- City: Contains more than 50% of the nodes inside the C-map

We present in figures 4 and 5 two samples of C-maps. A gray node represents an isolated node; a red node, a regular; a blue node, a focal. Their size in the picture relies on the number of identical contexts grouped into them. In figure 4, two town-clusters are displayed at the top, and five island-clusters at the bottom of the image. In figure 5, a city-cluster is displayed with an island-cluster at the top.

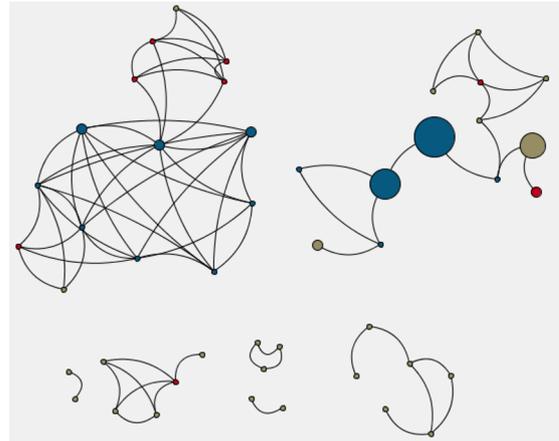


Figure 4: C-map with two town-clusters (top) and five island-clusters (bottom)

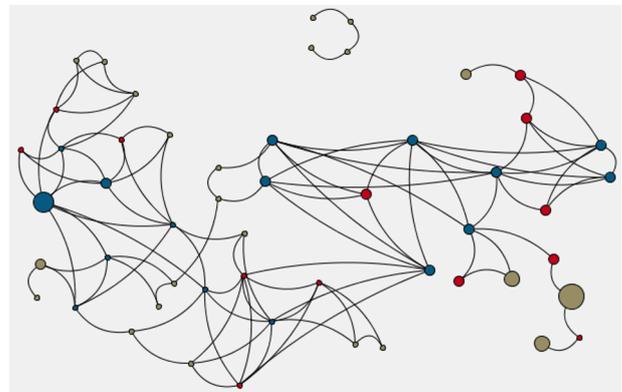


Figure 5: C-map with a city-cluster and an island-cluster

Evaluation process

In grounds of our previous work in this area, we have selected a set of features, known as story-characteristics, for evaluating a plot, and a different set of features, known as structural-characteristics, for evaluating the structures inside the knowledge base of a storyteller.

Evaluating story-characteristics The features utilized to evaluate a story are the following: preconditions fulfilled,

novel contextual structures, opening, climax, closure, originality, E-R ratio, number of actions and impasses in a story, character threads and social resolutions. The first eight are part of the feature set described in the evaluation model proposed by Pérez y Pérez (2007 and 2014), and the remaining features are additions to this evaluation model that we considered relevant for this work.

The preconditions fulfilled metric evaluates the number of action requirements satisfied within a story. This value corresponds to a number between 0 and 1 determined by the ratio between the number of preconditions fulfilled for every action versus the total number of preconditions in all the actions.

The novel contextual structures metric determines the amount of new knowledge that a story can generate if it were added to the set of previous stories of an agent. This value is determined by the ratio between the number of new buildable CSs from the story-actions and the total number of CSs that could be generated. We consider a CS new when its context is different from all the existing CSs.

The following metrics are related to the tension curve of a story and to the identification of the three main stages of a story in accordance with the Freytag's pyramid (Freytag 1896). Mexica considers a story to be properly built when it follows this structure. This is why we use it as a reference for these subset of metrics. A story has a correct opening when, at the beginning, there are no tensions and then they begin to grow; it has a correct climax when its highest tension value is similar to a reference value obtained from the set of previous stories; it has a correct closure when all the tensions in the story are solved when the last action is performed.

The originality feature determines the portion of a story that could be generated by the evaluating agent. Mexica is capable of generating a story by itself, from the beginning to a given action, when the following conditions are fulfilled:

1. The story-context associated with the action is similar to the context of one of the available CSs
2. The action is similar to one of the linked actions of the CS with the similar context

The result of this metric is the ratio between the number of actions that could be generated by the evaluating agent, and the total number of actions in a story. If one agent could generate the story on its own, the result is zero; if none of the story contexts are similar to any CS of the agent, the result is one (see figure 6).

$$originality = 1 - \frac{regeneratable\ actions}{total\ number\ of\ actions} \quad (1)$$

Figure 6: Originality

The ER-ratio feature determines the relation between the actions added during the reflection phases versus the actions added during the engagement phases. According to the E-R model, both the engagement and reflective stages should provide a similar number of actions to a story. We claim that a story with engagement actions will be novel, but lack

of coherence (since actions requirements are not validated at this stage). On the other hand, a story with reflective actions will be coherent, but lack of novelty (causal constraints are validated during reflection). In general, the result for this metric corresponds to one minus the absolute value of the difference between the engagement ($actions_E$) and reflection ($actions_R$) actions divided by the total ($actions$) number of actions (see figure 7). When the actions added in engagement and reflection are the same, the result is one. When the actions added in engagement or in reflection is zero, the result is zero.

$$ER - ratio = 1 - \left| \frac{actions_E - actions_R}{actions} \right| \quad (2)$$

Figure 7: ER-ratio

An impasse occurs when, during engagement, the context of a story is not similar to any context from the available CSs, and the stage finishes. We claim that this behavior occurs when the current story is interpreted as an unknown context for the agent. This feature determines the number of times this situation occurs during the generation of a story.

The character threads feature determines the number of groups (threads) of characters inside the story. We state that two characters belong to a thread when they have a significant relationship inside a story. This condition is fulfilled when two characters participate together in an action that generates or removes a tension between them. For this work, we narrow the number of groups in a story to maintain it simpler and to prevent the existence of parallel stories. The result of this evaluation is a number between 0 and 1 calculated as one divided by the number of character threads inside a story.

The social resolutions feature determines the number of social tensions that remain unsolved by the end of a story. These tensions are added by the Social Mexica component every time a social norm is broken inside a story. We are interested in determining how accurately Mexica finishes these additional tensions within a story in order to fit into the Freytag's pyramidal model. The result of this feature corresponds to the ratio between the number of social tensions solved versus the total number of these tensions in a story. When every social tension was solved, the result is one. When none of the social tensions were solved, the result is zero.

Evaluating structural-characteristics With regards to the knowledge structures, we analyze the C-maps defined to obtain the following set of metrics:

- Percentage of clusters of each type (cities, towns, and islands)
- Percentage of nodes of each type (focal, regular, and isolated)

The percentage of city-clusters describes the ratio between the number of them contrasted against the total number of clusters inside the C-map. Similar calculations are performed to determine the percentage of town-clusters and

island-clusters. For the percentage of focal-nodes we count the number of such nodes inside any cluster of the C-map and divide this by the total amount of nodes. We obtain the percentage of regular-nodes and isolated-nodes in a similar way.

Identifying relations

Here, we describe a process to identify relations between the story-characteristics and the structural-characteristics described above.

The relations identified in this work are classified into two categories: cluster ratios and node ratios. In the following paragraphs, we explain the process to identify such relations.

The first step consisted in gathering 40 previous stories and partitioning them into sets. With this, ten stories were located into each set, conforming four story-sets (SS). Then, we split them into two story-banks (SB) with two story-sets each. Next, we recombined the stories on every bank to generate two additional sets, each with 70% of the stories of one story-set and 30% of the stories of the other (see figure 8). We performed the same process in both of the SBs.

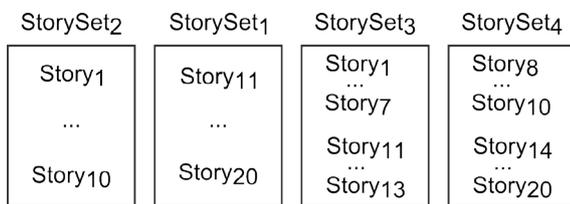


Figure 8: Visual representation of the first SB consisting of four SS. 70% of the stories in SS_3 came from SS_1 , and 30% from SS_2 . This proportions were inverted to generate SS_4 .

After these, we obtained four story-sets on each bank (eight story-sets in total divided into two banks) with the following characteristics:

- Every story-set has totally different stories from one of the story-sets in its story-bank
- Every story-set has 70% similar stories from one of the story-sets in its story-bank
- Every story-set has 30% of stories from one of the story-sets in its story-bank

We utilized each story-set as input for each of the eight different story-generation agents. We let each one of them to generate thirty stories, and we repeated this process three times. By the end of this process, we collected 90 stories per agent. The next step consisted on evaluating every generated story. Each of them was evaluated by every agent in the same bank of the generator, obtaining four evaluations per story. Each evaluation comprised the metrics previously described.

Once these evaluations were completed, we removed those outputs that we did not considered as valid stories according to the following criteria: its preconditions are fulfilled in at least 75%, it has only one character thread, and it contains at least four actions. We collected the evaluations of the remaining stories, obtained the averages for each metric

(considering the four evaluations), and we validated if there were differences among them for each of the agents. For this task, we performed an analysis of variance preceded by a K-S test -Kolgomorov-Smirnof test (Massey 1951)- to validate that the data was normally distributed (a request for the variance analysis).

Once we obtained the average values for every metric for every agent, we analyzed the knowledge utilized during the story generation process. The first step consisted in generating the corresponding C-map for every agent to obtain ratios between the different types of nodes and clusters.

We calculated the coefficient of determination (R^2) and the Pearson correlation for every metric utilized during the evaluation process against every metric utilized to describe the knowledge structure. These values leded us to identify relations between the story-characteristics and the structural-characteristics.

In general, the Pearson-correlation coefficient is a decimal value between -1 and 1. A positive value represents a direct relation between two data sets (when one grows the other does it too), whereas a negative value represents an inverse relation (when one grows, the other decreases), and a value close to 0 represents no linear relation between them. The R^2 value represents how close a data set behaves according to a polynomial of degree n . When $n = 1$, it represents how close is the data to a linear behavior. A value of one for this metric corresponds to a perfect match with a linear behavior, whereas a value of zero represents the absence of a linear correspondence. We now present the relations between every pair of metrics whose values were close to one, which identifies highly related data sets.

Results

Now we present only the results obtained for those relations found between story-characteristics and structural-characteristics with a strong Pearson correlation value (greater than 0.5 or lower than -0.5). The rest of the possible pairings were removed since their Pearson correlation values were not significant. Further studies will determine whether exist additional nonlinear relations among these banned pairings.

In figure 9, we present the novel contextual structures evaluation averages contrasted against the percentage of focal and isolated nodes for each agent. The Pearson correlation values obtained were 0.8 for focal nodes and -0.71 for isolated nodes, and the R^2 values for $n = 1$ were 0.51 and 0.64 respectively. The first values represent a positive linear relation between the novelty of a story and the number of focal nodes inside the story generator, and the second values represent a negative linear relation between the novelty and the number of isolated nodes.

In figure 10, we present the opening averages against the percentage of city and island clusters for each agent. The Pearson correlation values obtained were 0.78 for city clusters and -0.86 for island clusters, and the R^2 values for $n = 1$ were 0.60 and 0.74 respectively. The first values represent a positive linear relation between the opening of a story and the number of city clusters inside the story generator, and the second values represent a negative linear relation

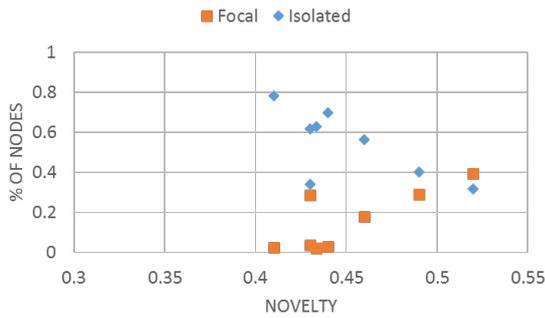


Figure 9: Novel contextual structure averages versus percentage of focal and isolated nodes

between the opening and the number of island clusters.

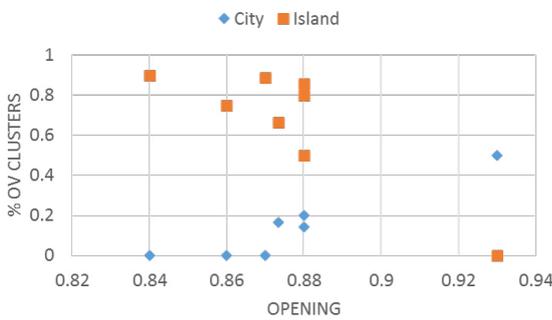


Figure 10: Opening averages versus percentages of city and island clusters

In figure 11, we present the climax averages against the percentage of focal and isolated nodes for each agent. The Pearson correlation values obtained were 0.83 for focal nodes and -0.85 for isolated nodes, and the R^2 values for $n = 1$ were 0.69 and 0.72 respectively. The first values represent a positive linear relation between the climax of a story and the number of focal nodes inside the generator knowledge base, and the second values represent a negative linear relation between the climax and the number of isolated nodes.

In figure 12, we present the closure averages against the percentage of city and island clusters for each agent. The Pearson correlation values obtained were -0.66 for city clusters and 0.67 for island clusters, and the R^2 values for $n = 1$ were 0.44 and 0.45 respectively. The first values represent a negative linear relation between the closure of a story and the number of city clusters inside the story generator, and the second values represent a positive linear relation between the closure and the number of island clusters.

In figure 13, we present the character threads' averages against the percentage of focal and isolated nodes for each agent. The Pearson correlation values obtained were 0.79 for focal nodes and -0.86 for isolated nodes, and the R^2 values for $n = 1$ were 0.62 and 0.74 respectively. The first

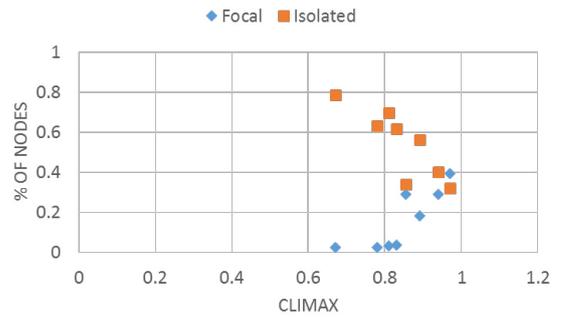


Figure 11: Climax averages versus percentages of focal and isolated nodes

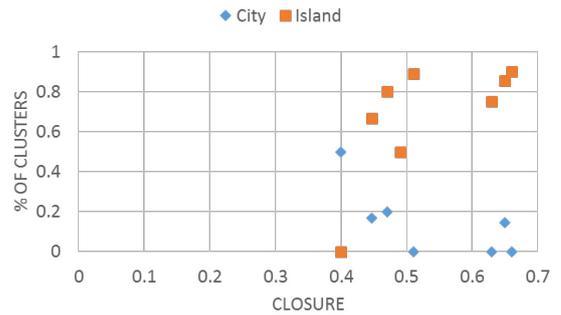


Figure 12: Closure averages versus percentages of city and island clusters

values represent a positive linear relation between the character threads of a story and the number of focal nodes inside the generator knowledge base, and the second values represent a negative linear relation between the character threads and the number of isolated nodes.

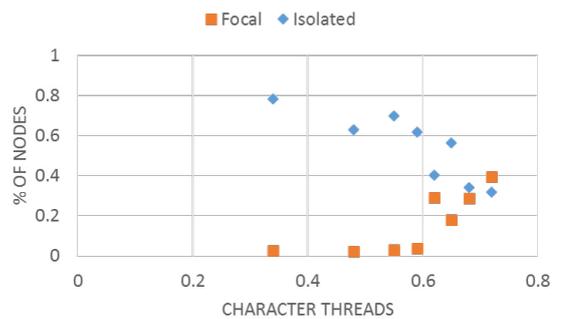


Figure 13: Character threads averages versus percentages of focal and isolated nodes

In figure 14, we present the social resolution averages against the percentage of city and island clusters for each agent. The Pearson correlation values obtained were -0.87 for city-clusters and 0.67 for island-clusters, and the R^2 values for $n = 1$ were 0.75 and 0.45 respectively. The first

values represent a negative linear relation between the social resolutions in a story and the number of city-clusters inside the generator knowledge base, and the second values inside represent a positive linear relation between social resolutions and the number of island-clusters.

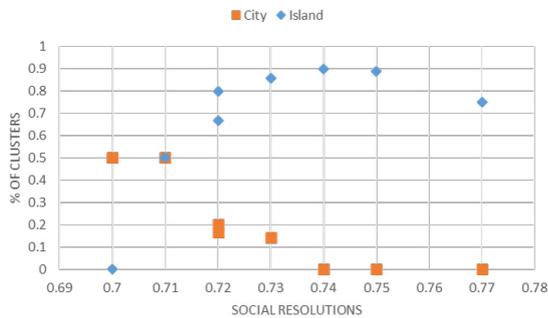


Figure 14: Social resolution averages versus percentages of city and island clusters

In figure 15, we present the originality evaluation averages contrasted against the percentage of town and island clusters for each agent. The Pearson correlation values obtained were -0.75 for town clusters and 0.61 for island clusters, and the R^2 values for $n = 1$ were 0.56 and 0.38 respectively. The first values represent a negative linear relation between the originality of a story and the number of town clusters inside the story generator, and the second values represent a weak positive linear relation (since values are not close to 1) between the originality and the number of island clusters.

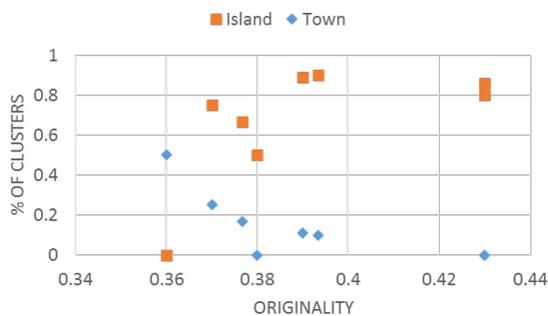


Figure 15: Originality averages versus percentages of town and island clusters

In figure 16, we present the E-R ratio averages against the percentage of focal and isolated nodes for each agent. The Pearson correlation values obtained were 0.87 for focal nodes and -0.86 for isolated nodes, and the R^2 values for $n = 1$ were 0.75 and 0.73 respectively. The first values represent a positive linear relation between the E-R ratio and the number of focal nodes inside the generator knowledge base, and the second values represent a negative linear relation between the E-R ratio and the number of isolated nodes.

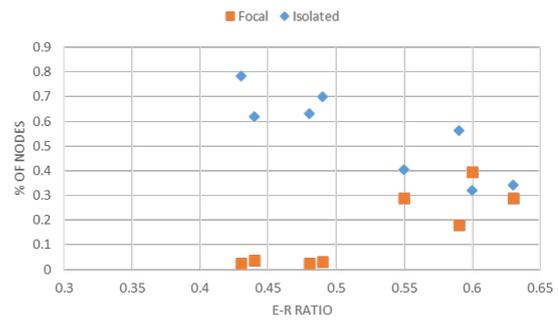


Figure 16: E-R ratio averages versus percentages of focal and isolated nodes

In figure 17, we present the average story size (in actions) contrasted against the percentage of focal and isolated nodes for each agent. We also present the average number of impasses against the percentage of regular nodes for each agent. The Pearson correlation values obtained were 0.87 for focal nodes, -0.86 for isolated nodes, and the R^2 values for $n = 1$ were 0.75 , 0.74 and 0.55 respectively. The first values represent a positive linear relation between the story size and the number of focal nodes inside the story generator, the second values represent a negative linear relation between the story size and the number of isolated nodes, and the third values represent a negative linear relation between the number of impasses and the number of regular nodes.

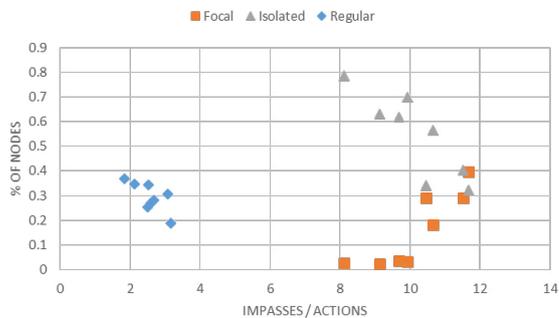


Figure 17: Story size averages versus percentages of focal and isolated nodes, and impasse averages versus percentages of regular nodes

Discussion

The main goal of this project was to identify how the knowledge structures of an automatic generator of narratives influence the presence of diverse features of its generated stories.

In table 1, we present a summary of the linear relations found between the story-characteristics and the structural-characteristics described on this paper. These results are divided into two sections: node relations and cluster relations.

Element	Positive relations	Negative relations
Focal nodes	novelty, climax story size character threads E-R ratio	impasses novelty, climax story size character threads E-R ratio
Regular nodes Isolated nodes		
City clusters	opening	closure soc. resolutions
Town clusters Island clusters	originality closure soc. resolutions	originality opening

Table 1: Summary of the obtained relations

We utilized as story-characteristics opening, climax, closure, originality, novel contextual structures, impasses, E-R ratio, story size, character threads and social resolutions, in grounds of a previous work on evaluation of stories to determine its interestingness, novelty and coherence (Pérez y Pérez 2014), features considered relevant for a story to be considered creative.

We made use as structural-characteristics the percentage of nodes and clusters inside the knowledge base of the analyzed agents. Nodes represent CSs obtained after interpreting the previous stories of the agents, and clusters represent groups of similar nodes. We defined the concept of similarity between nodes in terms of the similarity between the relations of the CS-contexts. Representing the internal knowledge of an agent as CSs let us qualitatively describe it, which lead into the creation of structures, called C-maps, to visualize the similarities among its information.

Our findings let us now formulate questions about the process of incorporating new stories into the agent’s knowledge base. Before this research, we envisioned to have an agent with as many previous stories as possible, but know we have evidence that this is not always the best scenario. For instance, this agent would be a deficient evaluator since its evaluations, in particular for novel CSs and originality, will often be low. This assumption lead us to redefine our definition of novelty. Now, we perceive diverse scenarios where novel CSs emerge: when a new story originates different context from those in the evaluating agent; when a new story utilizes the existing contexts but in different ways; when a new story utilizes rare contexts. With the categorization presented for nodes and clusters, we are able to identify these new context types, to measure its presence, and to validate how it affects the story generation process.

These results lead us to think on the optimal number of previous stories that an agent should have to generate higher-evaluated stories, and to become an accurate story evaluator. If an agent had enough stories to cover all the possible story-contexts, its evaluations of novelty and originality will always be zero, and the number of possible continuations

for every story would be so vast that unusual and even incoherent stories could be generated. Its C-map would consist of focal nodes galore, and a big city-cluster. In general, as we develop a better understanding of the implications of diverse knowledge arrangements for the story generation process, we will be able to progress in the construction of more accurate ways of generation and evaluation of such outputs.

It is worth to mention that our final averages does not consider all the 90 stories generated by every agent, since some of these outputs lacked of what we considered as basic characteristics to be considered stories (a minimal number of actions, preconditions satisfied and one character thread). We also measured the ratio of these valid stories against the invalid stories and we looked up for relations with our structural metrics, but we did not find any linear relation. These results give us an inkling of the complexity of generating valid stories. In further research, we will look for non-linear relations and multifactorial relations to cast light on which structures might diminish the generation of invalid stories.

In grounds of our presented results, we showed that, in general, focal nodes improve the novelty of the generated stories because of its conception process. These nodes are built from similar inspiring stories when their CS are extracted and incorporated to the repository. In fact, these nodes provide a wide variety of continuations for a single context since every connection to a focal node comes from a similar CS that can be employed to progress a new story. Moreover, the size of the generated stories is bigger when focal nodes come into play because of this higher number of possibilities, and becomes easier to reach an appropriate number of tensions during the story climax, and to maintain a unique character thread. On top of that, the number of $actions_E$ increases, and is closer to the number of $actions_R$, resulting in a higher E-R ratio.

We also found that, in general, isolated nodes play the opposite role of focal nodes. For instance, they diminish the novelty, climax and the size of the generated stories. Nevertheless, an isolated node can be perceived as a focal node in an early developmental stage, so they are required for the focal nodes to come into play.

Regarding to clusters, cities provide a solid ground for the stories to initiate, but as the process continues, cities widen the number of possible continuations and the stories tend to have closures with multiple unsolved tensions. Contrasting with our initial assumptions, we did not find any evidence of a strong negative relation between cities and originality nor a strong positive relation between them and valid stories.

On the other hand, the presence of islands in the early stages of a story originates multiple impasses, but they incorporate original paths and bounded closures. Finally, towns diminish the originality of the stories since they provide solid structures with multiple similar contexts, but they still lack of focal nodes so the continuations are still not too different.

These results support our claim about the existence of linear relations between structural elements in the knowledge base of our storyteller and features of its generated stories. In our model, these elements are obtained from a set of previous stories, which shows how previous experiences affect

the generation of new narratives. Nevertheless, we still need to do additional research efforts to validate if the obtained relations are causal (i.e. the structural-characteristics are the origin of the story-characteristics), or circumstantial (i.e. the structural and the story characteristics are both generated by additional factors). This research has widened our scope to identify the existence of these additional factors, to progress in our understanding of how the structural elements inside the knowledge base of any agent affects the characteristics of its generated narratives.

Conclusions

We showed in this paper relations among structural settings of the knowledge base of an automatic storyteller (Mexica) and features of its generated stories.

We introduced the concept of nodes and clusters built upon CSs inside the agents' knowledge bases. We classified nodes into three different categories: focal, regular and isolated, and also classified clusters of these nodes into three different sets: cities, towns and islands. We have described connectivity maps (C-maps), which reflect how similar the nodes inside the knowledge base of a storyteller are.

We described a set of metrics to identify story features such as preconditions fulfilled, novel contextual structures, opening, climax, closure, character threads, social resolutions, originality, E-R ratio, and number of impasses, and a set of metrics to describe knowledge structures inside the agents based on the nodes and clusters they contain.

We hypothesized how nodes and clusters, when present in the knowledge structure of an automatic storyteller, affect diverse story features. Next, we validated these claims by implementing our model utilizing Mexica and Social Mexica, evaluating each of the generated stories, and then contrasted the evaluations against each of the metrics describing the internal structure of the knowledge bases utilized during the generation process.

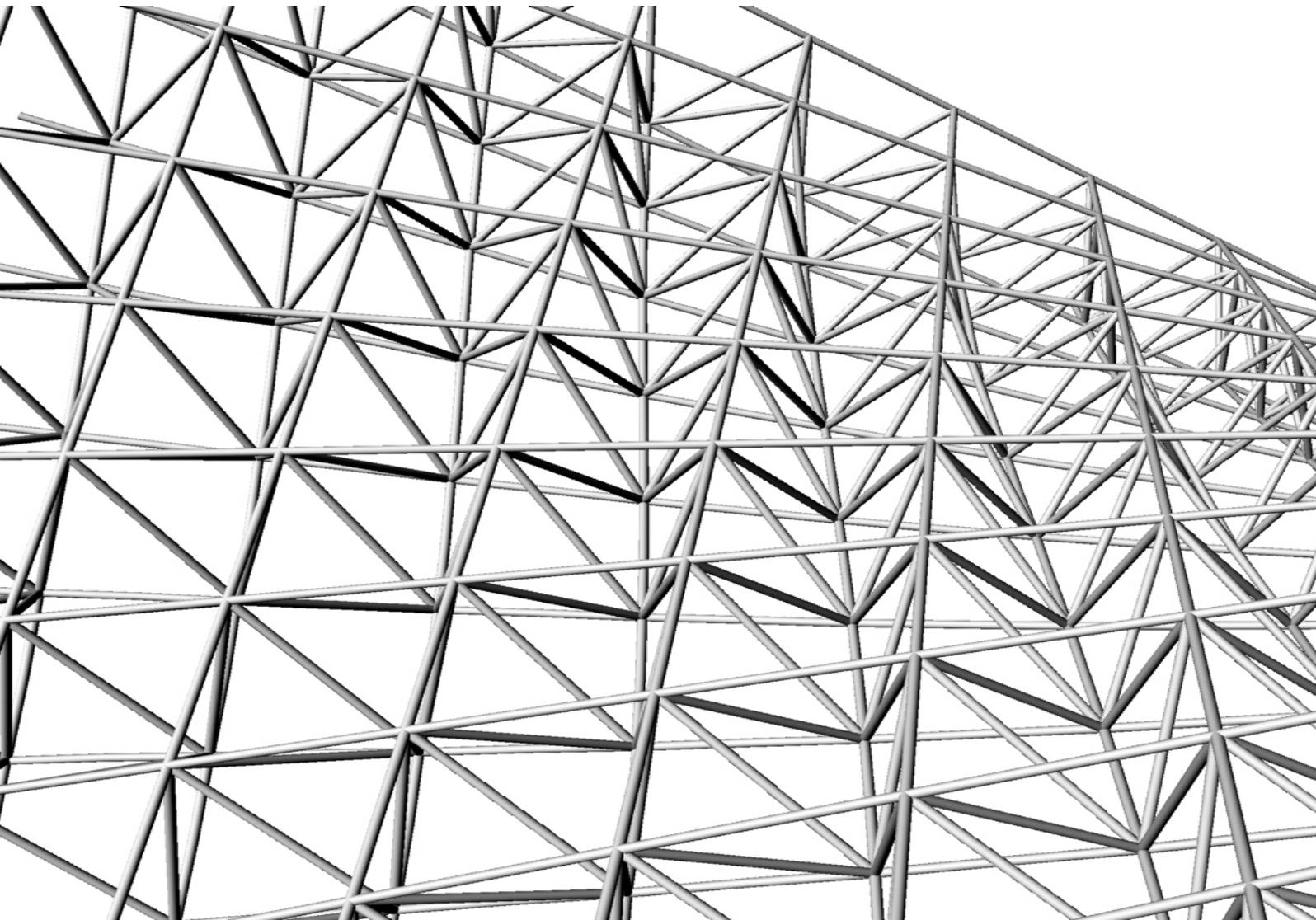
Acknowledgments

This research was sponsored by the National Council of Science and Technology in Mexico (CONACyT), project number: 181561.

References

- Freytag, G. 1896. *Technique of the drama. An exposition of dramatic composition and art*. S. C. Griggs and company.
- Guerrero, I., and Pérez y Pérez, R. 2014. Social mexica: A computer model for social norms in narratives. In *Proceedings of the Fifth International Conference on Computational Creativity*.
- Klein, S. 1965. Control of style with a generative grammar. *Language* 41:619–631.
- Massey, F. 1951. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association* 46:68–78.
- Pérez y Pérez, R., and Sharples, M. 2001. Mexica: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13(2):119–139.
- Pérez y Pérez, R. 1999. *Mexica, A computer model of creativity in writing*. Ph.D. Dissertation, University of Sussex, England.
- Pérez y Pérez, R. 2007. Employing emotions to drive plot generation in a computer-based storyteller. *Cognitive Systems Research* 8:89–109.
- Pérez y Pérez, R. 2014. The three layers evaluation model for computer-generated plots. In *Proceedings of the Fifth International Conference on Computational Creativity*.
- Pérez y Pérez, R. 2015. A computer-based model for collaborative narrative generation. *Cognitive Systems Research* 36-37:30–48.
- Riedl, M. 2004. *Narrative Planning: Balancing Plot and Character*. Ph.D. Dissertation, Department of Computer Science, North Carolina State University, Raleigh, NC.
- Turner, M. 1994. *The Creative Process: A Computer Model of Storytelling*. Hillsdale: Erlbaum (Lawrence).

GENERATING STRUCTURE



Flexible Generation of Musical Form: Beyond Mere Generation

Arne Eigenfeldt

School for the
Contemporary Arts
Simon Fraser University
Vancouver, Canada
arne_e@sfu.ca

Oliver Bown

Art and Design
University of
New South Wales
Sydney, Australia
o.bown@unsw.edu.au

Andrew R. Brown

Queensland
College of Art
Griffith University
Brisbane, Australia
andrew.r.brown@griffith.edu.au

Toby Gifford

Queensland Conservatorium
Griffith University
Brisbane, Australia
t.gifford@griffith.edu.au

Abstract

Despite a long history of generative practices in music, creation of large-scale form has tended to remain under direct control of the composer. In the field of musical metacreation, such interaction is generally perceived to be desirable, due to the complexity of generating form, and the necessary aesthetic decisions involved in its creation. However, the requirements for dynamic musical generation in games, installations, as well as in performance, point to a need for greater autonomy of creative systems in generating large-scale structure. This position paper surveys the complexity of musical form and existing approaches to its generation, and posits potential methods for more computationally creative procedures to address this open problem.

Introduction

Galanter's definition of generative art (2003) – “any art practice where the artist uses a system... which is set into motion with some degree of autonomy contributing to or resulting in a *completed work of art*” (italics ours) – assumes a fully finished artwork. Furthermore, implicit in this definition is that the system may involve human interaction, in that the system need only *contribute* to the final work. In practice, human involvement, whether through algorithm design, direct control by an operator or interaction with a live human performer, has remained an active presence in the dynamic generation of music.

One reason for this is that generating entire musical compositions entails the development of musical form, a highly complex task (Berry 1966). Form, discussed more fully below, involves the complex interaction of multiple musical structures in order to logically organise the work's progression in time. Strategies are required to organise these structures so as to “provide reference points for the listener to hold on to the piece, otherwise it may lose its sense of unity” (Miranda 2001).

Musical metacreation (MuMe) is an emerging term describing the body of research concerned with the automation of any or all aspects of musical creativity (Pasquier *et al.* 2016). It looks to bring together and build upon existing academic fields such as algorithmic composition (Nierhaus 2009), generative music (Dahlstedt and McBurney 2006), machine musicianship (Rowe 2004) and live algorithms (Blackwell *et al.* 2012). There have been many musically successful MuMe production systems that have generated

complete compositions, and therefore generated long-term musical structure. As MuMe does not exclude human-machine interaction, these systems have tended to rely upon human-machine partnerships.

Although MuMe considers itself to be a subfield of computational creativity (CC), some definitions of the latter exclude a large part of the former. Colton and Wiggins (2012) suggest that the degree of creative responsibility assigned to a CC system *may* include the “development and/or employment of aesthetic measures to assess the value of artefacts it produces” as well as “derivation of motivations, justifications and commentaries with which to frame their output”. Veale (2015) on the other hand declares that any works that do not meet these self-reflective criteria are to be deemed “mere generation”. For many creative practitioners, however, and perhaps musicians in particular, “merely” generative systems can still play a hugely important role in human-computer co-creativity, and despite this lack, generative software can still be considered actively creative (Compton and Mateas 2015). Much innovation has been achieved with MuMe systems that make no claim to be fully autonomous, and as noted in the reflections on the first Musical Metacreation Weekend (Bown *et al.* 2013), the delegation of large-scale musical structure to a system is challenging, to the point that many composers felt the need to remain “in the loop” and in order to maintain control over form interactively.

In some instances, interaction with the generative system is significantly restricted by design; for example, composition may entail managing a surfeit of parameters that constrain the system's choices (e.g. Bown and Britton 2013). Alternatively, the presentation may require decisions in absence of a human, such as a continuously running installation (e.g. Schedel and Rootberg 2009) or interactive media (e.g. Collins 2008). For these very practical reasons, along with intellectual and aesthetic reasons, the need to automate long-term structure is a pressing issue in the MuMe community. This paper will describe musical form and the difficulties in its creation, present some existing methodologies for its creation and outline what we feel are some novel approaches to the problem, with particular attention to adapting generative techniques for formal design to dynamic situations.

Defining musical form

Musical form is “the result of the deployment of particular materials and processes” (Whittal 2016). Our immediate perception of music is its *surface*: the relationship between individual events: for example, a melodic phrase, a given chord, a drum beat, the signal processing on a sound recording. The selection and resulting relationships between these individual objects at a given point in time can be considered the music’s *design* (after Salzer 1962). These surface elements almost always undergo some sort of organisation over time, and such methods involve the creation of *musical structures*: for example, the combination of melodic phrases into a longer melody, the organisation of harmonies into a repeating progression, the combination of related drum beats into an eight-bar phrase, the control over time of a signal processing parameter, such as the slow opening of a filter. *Form* is the consequence of the structural relationships between the various musical elements.

Kramer (1988) suggests that one difficulty in conceptualising form is due to its inherent role in organising time. While theories exist dealing with rhythm and meter (e.g. Xenakis 1992; Lester 1986), “more difficult to discuss are motion, continuity, progression, pacing, proportion, duration, and tempo”, all aspects to do with musical form. Schoenberg expressed the complexity of form, stating that it requires “logic and coherence” in order for a musical composition to be perceived as being comprehensible, but that its elements also should function “like those of a living organism” (Schoenberg and Stein 1970). The dilemma for young composers, and MuMe practitioners, is achieving a balance between a strict structure that appears logical, with organic elements that engender surprise.

Forms that are not based upon functional tonality’s goal-directed nature – such as the non-developmental and non-teleological structures often found in ambient music, world music, and specifically Stockhausen’s *Momentform* (1963) – still require subtle and deft handling: “the order of moments must appear arbitrary for the work to conform to the spirit of moment form”, yet they must not *be* arbitrary (Kramer 1978). For example, many works of Stockhausen

depend upon *discontinuity* for their structural effect, but the points of division require careful selection:

Ending a permutational form is nearly always a matter of taste, not design. While the listener may be satisfied with a sensation of completion, the composer knows that though a series of permutations may eventually be exhausted, it does not automatically resolve. The ending’s essential arbitrariness has to be disguised (Maconie 1976).

Certain formal relationships have proven more successful than others, and these relationships became standardised: from simple procedures – such as ternary, rondo, and canon – to more complex relationships, such as sonata. All of these can be considered *architectural forms*, which pre-exist, and to which structures and surface features can be “poured into”: in other words, a *top-down* approach.

The opposite method has been to allow the material to define its use: form from material (Boulez *et al.* 1964). Such organic procedures have found great success in much twentieth century art-music, including improvisation, and can be considered a *bottom-up* approach. Narmour (1991) provides a useful discussion on the interaction and opposition of these two approaches in musical composition.

While recent research on form in pre-20th century tonal music has provided new insights (Caplin 1998), none has appeared with the same depth and scope for contemporary non-tonal music. Kramer’s examination of *Momentform* (1988) offers compelling views on non-teleological music, while also noting that 20th century composers’ rejection of traditional (i.e., top-down) approaches to form based upon expectation have forced new non-universal formulations of large-scale organisation: “continuity is no longer part of musical syntax, but rather it is an optional procedure. It must be created or denied anew in each piece, and thus it is the material and not the language of the music” (Kramer 1978). Another theme in late 20th Century musicology has been an increasing respect for non-Western, or non-art-music structures. Consideration of long-term form in Indian, Indonesian and African music, for example, or in contemporary electronic dance music, broadens the scope of this enquiry.

	H	L			I	L	I		I	L	I			H
	1	2	3		6	9	12		17	22	23			31
<i>meter</i>	x				x						x			x
<i>tempo</i>	x										x			x
<i>attack</i>	x	x	x		x	x	x		x		x			x
<i>density</i>	x	x	x			x	x		x		x			x
<i>harmony</i>	x				x		x		x		x			x
<i>motivic</i>	x	x	x		x	x	x				x	x		x
<i>repetition</i>	x								x		x			x
<i>texture</i>	x	x	x		x		x		x		x	x		x
<i>orchestration</i>	x	x	x		x	x	x		x		x	x		x
<i>register</i>	x	x	x		x	x	x		x		x	x		x
<i>loudness</i>	x				x	x			x		x			x

Fig. 1. Park’s analysis of discontinuities within the first 31 measures of Debussy’s *De l’aube à midi sur la mer*. At left are the form delimiting parameters, above are the four architectonic levels: High, Intermediate, Lowest, (none), and the measures in which alterations occur. X indicates a discontinuity in the music for that parameter.

Music theorist Richard Parks has produced an interesting analysis of Debussy's music (1989) that provides a clue to the composer's unique organisational methods regarding structure. Parks suggests that form-defining parameters include meter, tempo, successive-attack activity, sonorous density, harmonic resources, thematic/motivic resources, repetition/recurrence, quality of texture, orchestration, register, and loudness. He then examines a variety of compositions by Debussy, and partitions the works based upon the locations of simultaneous alterations of these parameters: the discontinuities (see Fig. 1). The resulting delimiters demonstrate how a multiplicity of structural boundaries interact to create larger formal structures.

Although post-structuralist thinkers challenged how form might be considered in music (Nattiez 1990; Dahlhaus 1989), their contribution mainly concerns the potential *meaning* and *reception* found within a work's form. The former has no bearing in our discussion here; the latter has allowed new viewpoints, specifically those involving cognition and musical perception (e.g. Meyer 1956; Sloboda 1991; Deutsch 2013) to become considerations in generative music system design and use. Consistent between pre-20th century, modernist, and postmodernist concepts has been the role of modulating tension as a means of structural and formal design. Perceptual models of musical tension have recently been formulated (Farbood 2012), and, more generally, cognitive models for composition have themselves been evaluated (Pearce and Wiggins 2007); however, the latter have only been used to generate short musical excerpts, and not complete works.

Musical composition requires the organisation of material in such a way that it provides enough surface variation to maintain the listener's interest, while providing enough structural repetition in order to avoid overwhelming the listener with new material. This continuum can be compared to information theory's compressibility (Shannon and Weaver 1949), or Galanter's complexism theory (Galanter 2008), with the "sweet spot" being the formally-balanced, aesthetically pleasing musical work. Structures to control surface features in music can be generated – by humans or computationally – without too much difficulty, as shown in the wealth of interactive music production systems; knowing *when* to apply and alter such structures takes a great deal more sophistication and contextual knowledge and understanding.

Generative Music

Algorithmically generated music has a long and rich history: from Mozart's musical dice game (Hedges 1978), aleatoric compositions of Cage, Cowell and Stockhausen (Nyman 1999), through to compositions done in part by a computer program (Hiller 1970). These have been variously described as algorithmic composition (Cope 2000), generative music (Eno 1996), procedural music (Collins 2008) and, more recently, as musical metacreation (Bown *et al.*

2013). Previous computational models of musical structure include *Cypher* (Rowe 1992), *Experiments in Musical Intelligence* (Cope 1996), *GESMI* (Eigenfeldt 2013), the use of statistical prediction (e.g. Conklin 2003), the use of machine learning techniques (e.g. Smith and Garnett 2012), and agent negotiation (e.g. Eigenfeldt 2014). Sorensen and Brown (2008) explored human-guided parametric control over structure in the *MetaScore* system.

MuMe for interactive media faces the challenge of adapting to an *a priori* unknown and unfolding dramatic structure. As Karen Collins notes, in the context of video-game composition:

procedural music composers are faced with a particular difficulty when creating for video games: the sound in a game must accompany an image as part of a narrative, meaning sound must fulfill particular functions in games. These functions include anticipating action, drawing attention, serving as leitmotif, creating emotion, representing a sense of time and place, signaling reward, and so on (Collins 2008).

Musical descriptions of drama are often connected with temporal structure – indeed for some music theorists "structural and dramatic factors are fundamentally inseparable" (Suurpää 2006). Other emotive descriptors for music such as tension, relaxation, anticipation and surprise are variously described as operating in the moment (Hindemith 1970; Huron 2006), across phrasal structures (Narmour 1990; Huron 2006; Negretto 2012) or across the structure of sections, movements, and entire pieces (Schenker 1972; Lerdahl and Jackendoff 1983).

MuMe for interactive media has focused, up to now, on reactivity (e.g. Eigenfeldt 2006) or on generative techniques operating over short timescales, to suit an externally supplied dramatic contour (e.g. Hoover *et al.* 2014). Current techniques include selecting from pre-composed content (Collins 2008) or algorithmic manipulation of symbolic scores (Livingstone *et al.* 2010) on receipt of a signal from the host system. What remains conspicuously absent is the dynamic generation of longer-term temporal structures.

As Nick Collins notes, "it is rare to see engagement from algorithmic composition research with larger-scale hierarchical and associative structure, directedness of transition, and interactions of content and container" (Collins 2009). Perhaps in response to this perceived dearth of fully formed generative works, Collins pursued a brief research direction involving a multi-agent generative acousmatic system, *Autocousmatic*, which created complete electroacoustic works (Collins 2012), discussed more thoroughly later.

Example Practices in Generating Form

Despite these difficulties, designers of MuMe systems have made attempts to control structure through generative

means. We outline some of these approaches, both bottom-up and top-down.

Bottom-up: Perceived Structure through Self-Organisation

Many MuMe systems have relied upon the human performer, whether an improvising musician or the designer operating the machine directly, to “move the system along” (e.g. Lewis 1999; Pachet 2004). The complex interactions between human and machine can give rise to an organic self-organisation (Blackwell and Young 2004; Beyls 2007).

Some systems have attempted to impart a musical form upon the improvisation. Within *The Indifference Engine* (Eigenfeldt 2014), agents generate individual formal structures upon initiation that provide density and activity goals over the course of the work. These structures are continuously adapted based upon how they perceive the evolving environment, which includes a human performer. Within the *JamBot* (Gifford and Brown 2011) target complexity levels can be managed to vary or maintain sectional characteristics that dynamically balance the texture of human and generated parts.

Musebots (Bown *et al.* 2015) are autonomous musical agents that interact in performance, messaging their current states in order to allow other musebot to adapt. Recent musebots have been developed that broadcast their intentions, and not just their current state, thereby allowing other musebots to modify their own plans¹.

Top-down: Architectural models of structure

Adopting a more architectural approach within generative music has required pre-generation of formal structures in varying degrees. *GESMI* (Eigenfeldt 2013) creates complete electronic dance music tracks, using structural rules derived from a supplied corpus. Formal repetition is the first structural element generated, using a Markov-model learned from the example music, with surface features later filled in. Due to the clear repetitive phrases found within the original styles, *GESMI*'s forms are entirely believable.

Lerdahl and Jackendoff's much discussed Generative Theory of Tonal Music (1983) offers a tantalising model for top-down generation using musical grammars; unfortunately, it has never been successfully implemented in a production system – most likely due to its dependence upon 19th century functional tonality – and only a limited number of times as an automated analysis system (Hamana *et al.* 2006).

Cope (2000) models musical tension at several hierarchical levels through SPEAC: statement, preparation, extension, antecedent, consequent. Cope hand tags his corpus with these labels based upon harmonic tension, and uses

these tags when selecting from the corpus in his recombinant methodology, enabling the generation of high-level templates that can be filled in later.

It is also possible to impart formal structures upon self-organising material. *Coming Together: Notomoton* (Eigenfeldt 2014) uses a multi-agent system exploring such organisation through agent negotiation. While the surface variation resulting from the agent interaction provides surface interest, variety in macrostructure is ensured through the use of an algorithm initiated at the beginning of the performance that segments the requested performance's duration into sections, replete with varying goals for the defining musical parameters.

We note that the potential to influence self-organisation is an active research area within computer science: guided self-organisation (Prokopenko 2009).

Towards Organic Top-down / Bottom-up Form Generation

For purposes of dynamic generation – for example, music for online games, generative video, or more structured musical improvisation – architectural form's inflexibility provides little attraction or utility; conversely, the more organic self-organisation model is extremely difficult to control. Instances of dynamic musical generation in acoustic situations have tended to involve improvisation (Hill 2011), although some efforts involving generative methods have recently appeared (d'Esquivan 2014). As such, there are no existing models available for computational dynamic generation of which we are aware.

Whether approaching the problem of dynamic generation from a top-down or bottom-up perspective, human interaction has remained conspicuously present. In order to design generative musical systems that can produce flexible long-term temporal structures that adjust to the dynamic situations of gaming and generative multimedia, it is necessary to remove human interaction, and provide more autonomy to the system. Such a solution is necessary for more powerful MuMe systems, while at the same time approaching a true computationally creative system that will no longer be merely generative.

Because music has a large rule set – albeit rules that tend to have been agreed upon *after* the creative acts – some initial success has been achieved by directly codifying rules (e.g. Ebcioğlu 1988), or learning them through analysis (Conklin and Witten 1995). MuMe researchers do have access to large databases of symbolic music representations² which may produce further success in this direction; however, the material as it is provides potential use for melodic, harmonic, or rhythmic generation, but little use for structural generation, as such analysis has not yet been

¹ <http://musicalmetacreation.org/musebots/videos/>

² see <http://metacreation.net/corpus-1/> for a list of such corpora.

automated, despite promising beginnings (Kuhl and Jensen 2008).

The use of aesthetic agents within music has been proposed previously (Spector and Alpern 1994; Pearce and Wiggins 2001; Collins 2006, Galanter 2012), and their complexities noted. However, the higher one rises in the musical hierarchy (i.e. toward generation of complete musical compositions), the more one relies upon aesthetic judgment: it becomes more difficult, if not impossible, to evaluate creative output, since there are no optimal solutions in these cases (Pasquier *et al.* 2016) rendering even the judgment of relative suitability awkward.

Collins' *Autocousmatic* (2012) uses critical agents within an algorithmic compositional system. Complete fixed media works are generated based upon rules derived using machine-learning algorithms trained on exemplar works of acousmatic music. Formal aspects are derived from the database of works, using a top-down method. Several versions of a work are generated with varying surface details; the agents then analyse the candidate generations, comparing them to a single exemplar work, and the best version is selected.

When the completed generated works were evaluated by human composers, a recurring criticism centered around "problems of structure", "structural designs", and "issues... to do with larger forms". Firstly, one must acknowledge that the critical agents are unable to derive enough high-level knowledge from low-level feature analysis, so the perceived formal limitations are understandable; however, there are more fundamental issues in play. The agents are only able to compare the generative material to existing examples, rather than any intentionality of the compositional system at any point in the compositional process: as such, musical context for the decisions as they are being made is completely lost. In addition, the top-level architectural structures learned by the system are dislocated from lower level organisations and so the interdependence between hierarchical levels in creating a well-formed musical structure is absent.

Autocousmatic is not a real-time system, so the opportunity for selection – albeit completely automated selection – from a pool of extant generations exists. Performance systems, and those concerned with dynamic situations, eliminate such possibilities. Even when "big data" approaches, such as those of *Autocousmatic* and Collins' more recent work (2016), become conceivable in real-time, it is doubtful that they will solve the issue of dynamic generative structure for musical CC.

While we argue for the continued necessity of bringing artistic domain-specific knowledge to bear on any successful generative system – especially those that attempt to generate formal structures – we acknowledge the open problem of how such structures can be created dynamically through computational means.

Beyond Mere Generation - New Directions

We recognise the potential for the use of machine-learning to build aesthetic-agents in the real-time evaluation of generative music, with the understanding that they will require domain-specific knowledge in their assignment. We propose building on Collins' approach, with agents trained on specific corpora of exemplar music, Kramer's notion of discontinuities as form-defining elements, and recent research in musebot communications to express intentions and goals.

Musebots, described earlier, allow designers to create autonomous musical agents that interact in a collaborative ensemble with other musebots, potentially created by other designers. A particularly exciting aspect involves the notion that developers must decide *how* the musebots should interact, and *what* information is necessary to produce meaningful musical interaction. Musebots offer the potential to create complex musical surfaces and structures in which the organisation is emergent rather than attributed to a single clever programmer. Concepts of formal design have been raised already: initial musebot ensembles followed either a self-organising model, or a reactive model in which one musebot "took the lead" in determining sectional change. They have thus far avoided the requirement of large-scale formal structures by limiting their performances to five to seven minute compositions.

Musebots communicate their current states and, potentially, their intentions; however, as with all creative acts, intentions are not always achieved. Having dedicated musebots actively *listening* to music as it is being generated would allow for aesthetic decisions to be made as to when formal changes will need to be made, thus exemplifying a bottom-up perspective informed by high-level knowledge. These agents could be trained on specific styles, using standard MIR feature analysis (Tzanetakis and Cook 2000), having learned *why* formal changes occurred in the corpus. The example music would be hand-annotated by experts – rather than relying upon inexact machine analysis – at points of structural and formal change. The agents could learn to recognise a discontinuity – using models proposed by Parks (1989), for example – as well as examining the musical features prior to this break. How long is unvarying continuity acceptable until change is required? Or, at what point is boredom about to be felt by the listening agent (Eigenfeldt 2014)? This knowledge could then be used during generation, allowing the musebots to produce material using current methodologies (e.g. Eigenfeldt, Bown, and Carey 2015), while the listening agent could suggest when structural changes need to occur.

However, this model requires careful selection in determining the specific corpus, and locations within that corpus, on which the listening agents would be trained for the specific generation desired. There is no universal standard pertaining to musical form; how much repetition and variation is preferable in electronic dance music is significantly different than in free improvised music, or Debussy, for example.

Large-scale structure could be instantiated through the use of shape-negotiation; negotiation has already proven to be a useful method of organising musical agents (Eigenfeldt 2010). These shapes could be applied to a variety of musical structures over the course of a performance. Musebots have already been created that react to high-level attributes of valence and arousal (Eigenfeldt *et al.* 2015). Individual agents can ignore, agree, or offer alternatives to the formal contours; rather than having a single agent issuing predetermined orders (or following the directions of a human operator), these shapes can be proposed, accepted, and altered by prescient musebots.

An important aspect in terms of the dynamic generation of longer-term temporal structures would be the potential for the shapes themselves to be modified by the agents in real-time. Rather than interpreting these shapes directly, breakpoints could be assumed to be individual goals at proposed formal divisions, with the agents determining individual trajectories toward these agreed upon goals. These divisions could be considered suggestions, and the bottom-up listening agents could provoke revisions to these breakpoints. We recognise this as being a form of dynamic time warping (Keogh and Ratanamahatana 2005).

This multi-agent approach also allows for the maintenance of alternative interpretations and corresponding generative options. It provides flexibility to changing circumstances required by dynamic and interactive systems such as interactive games or improvised music performance systems. This approach has been used for generative rhythms and formalised as the Chimera Architecture that simultaneously tracks a collection of viable scenarios for musical continuation (Gifford and Brown 2009).

To summarise, convincing musical intelligence involves coordinating and solving many micro-problems in order to achieve musical *coherence*. Generating sequences of events – whether low-level melodic shapes or high-level formal outlines – is a more elementary task of simple generation. We can use corpus-based strategies involving statistical or rule-based learning and endlessly generate content that sounds similar to other content, but when we attempt to insert originality – for example, in combinatorial creativity, by combining one kind of melodic style with an unrelated song structure, and trying to make these things ‘fit’ – we encounter problems of coherence that aren’t necessarily answered by looking at the corpus. We posit that these problems may inherently require forms of evaluation that take them into a domain beyond Veale’s “mere generation” and into computational creativity proper, where the only way to determine the value of an output is through its analysis.

Much MuMe research has already looked at whether this is indeed the case; for example, Blackwell and Young’s swarm/self-organisation approach (2004) looks to see how far non-evaluative structuring processes can be taken. The problem remains open, and we suggest that MuMe researchers should continue to pursue generative approaches to complex structure, following either the scientific tradition of attempting to create autonomous systems that im-

plement a theoretical hypothesis, or in the artistic tradition building interactive systems that attempt musical coherence with a human performer.

We feel that the musicology of Parks and others provides a strong starting point to these investigations and is a productive way forward. It is grounded in a level of analysis that is sufficiently abstract to apply to all music. What is potentially of great interest here is that it affords a tie-in with the kinds of linguistic reasoning that is present in other areas of computational creativity (Perez and Sharples 2001; Veale 2012). If we begin to think of systems that form their own concepts of musical structure, then we can imagine them building a language from which a logic emerges. This logic would define the coherence of the music, and could have generative potential through metaphors and other linguistic constructs. It would be a mid-level language, meaning not at the musical surface, but also not necessarily at the level of our actual use of language (i.e., a mentalesse representation). It would also be highly subjective, adaptive to the individual’s own experience, just as statistical learning approaches are, but very different to statistical learning in terms of generative process – it would involve analytical problem solving in an iterative generate-and-test cycle. This implies an approach where we would ask, for any given musical form, or corpus of musical forms: can a non-trivial conceptual language be constructed for which this music is coherent? Or given a set of such solutions, what are the generative properties from which coherent music can emerge?

References

- Berry, W. 1966. *Form in Music* (Vol. 1). Prentice-Hall.
- Beyls, P. 2007. Interaction and Self-organisation in a Society of Musical Agents. *ECAL Workshop on Music and Artificial Life*, Lisbon.
- Blackwell, T., Young, M. 2004. Self-organised music. *Organised Sound*, 9(02), 123–136.
- Blackwell, T., Bown, O., Young, M. 2012. Live Algorithms: towards autonomous computer improvisers. *Computers and Creativity*, Springer Berlin, 147–174.
- Boulez, P., Noakes, D., Jacobs, P. 1964. Alea. *Perspectives of New Music* 3(1), 42–53.
- Bown, O., Britton, S. 2013. Methods for the Flexible Parameterisation of Musical Material in Ableton Live. *Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, Boston, 24–26.
- Bown, O., Eigenfeldt, A., Pasquier, P., Martin, A., Carey, B. 2013. The Musical Metacreation Weekend: Challenges Arising from the Live Presentation of Musically Metacreative Systems. *AIIDE*, Boston, 27–34.
- Bown, O., Carey, B., Eigenfeldt, A. 2015. Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents. *International Symposium of Electronic Art*, Vancouver.

- Caplin, W. 1998. *Classical Form: A Theory of Formal Functions for the Instrumental Music of Haydn, Mozart, and Beethoven*. New York: Oxford University Press.
- Collins, K. 2008. *Game Sound: An Introduction to the History, Theory and Practice of Video Game Music and Sound Design*. MIT Press.
- Collins, N. 2006. Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems. *Doctoral dissertation*, Cambridge.
- Collins, N. 2009. Musical Form and Algorithmic Composition. *Contemporary Music Review*, 28(1), 103–114.
- Collins, N. 2012. Automatic Composition of Electroacoustic Art Music Utilizing Machine Listening. *Computer Music Journal*, 36(3), 8–23.
- Collins, N. 2016. Towards Machine Musicians Who Have Listened to More Music Than Us. *Musical Metacreation*, ACM Computers In Entertainment, forthcoming.
- Colton, S., Wiggins, G. 2012. Computational creativity: The final frontier? *European Conference on Artificial Intelligence*, Montpellier, 21–26.
- Compton, K., Mateas, M., 2015. Casual Creators. *Proceedings of the Sixth International Conference on Computational Creativity*, Park City, 228.
- Conklin, D., Witten, I. 1995. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1), 51–73.
- Conklin, D. 2003. Music Generation from Statistical Models. *AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, Aberystwyth, 30–35.
- Cope, D. 1996. *Experiments in Musical Intelligence*. A-R Editions.
- Cope, D. 2000. *The Algorithmic Composer*. A-R Editions.
- d'Escrivan, J. 2014. Videogames as digital audiovisual performance. *Conference on Computation, Communication, Aesthetics and X*, Porto, 248–259.
- Dahlhaus, C. 1989. *Schoenberg and the New Music: Essays by Carl Dahlhaus*. Cambridge University Press.
- Dahlstedt, P., McBurney, P. 2006. Musical agents: toward computer-aided music composition using autonomous software agents. *Leonardo*, 39(5), 469–470.
- Deutsch, D. 2013. *Psychology of music*. Elsevier.
- Ebcioğlu, K. 1988. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12(3), 43–51.
- Eigenfeldt, A. 2006. Kinetic Engine: Toward an Intelligent Improvising Instrument. *Sound and Music Computing Conference (SMC)*, Marseille, 97–100.
- Eigenfeldt, A. 2010. Coming Together - Negotiated Content by Multi-agents. *ACM Multimedia International Conference*, Firenze, 1433–1436.
- Eigenfeldt, A. 2013. Generating Electronica - A Virtual Producer and Virtual DJ. *ACM Creativity and Cognition*, Sydney.
- Eigenfeldt, A. 2014. Generating Structure – Towards Large-scale Formal Generation. *AIIDE*, Raleigh, 40–47.
- Eigenfeldt, A., Bown, O., Carey, B. 2015. Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble. *Proceedings of the Sixth International Conference on Computational Creativity*, 134–141.
- Eigenfeldt, A., Bizzocchi, J., Thorogood, M., Bizzocchi, J. 2015. Applying Valence and Arousal Values to a Unified Video, Music, and Sound Generative Multimedia Work. *Generative Art Conference (GA)*, Venice.
- Eno, B. 1996. Generative Music. <http://www.inmotionmagazine.com/enol.html>, accessed Feb. 1, 2016.
- Farbood, M. 2012. A parametric, temporal model of musical tension. *Music Perception*, 29(4), 387–428.
- Galanter, P. 2003. What is Generative Art? Complexity theory as a context for art theory. *GA*, Milan.
- Galanter, P. 2008. Complexism and the role of evolutionary art. *The Art of Artificial Evolution*, Springer, 311–332.
- Galanter, P. 2012. Computational aesthetic evaluation: past and future. *Computers and Creativity*, Springer, 255–293.
- Gifford, T., Brown, A. 2009. Do Androids Dream of Electric Chimera? *Improvise: The Australasian Computer Music Conference*, Brisbane, 56–63.
- Gifford, T., Brown, A. 2011. Beyond Reflexivity: Mediating between Imitative and Intelligent Action in an Interactive Music System. *BCS Conference on Human-Computer Interaction*, Newcastle Upon Tyne.
- Hamanaka, M., Hirata, K., Tojo, S. 2006. Implementing “A Generative Theory of Tonal Music”. *Journal of New Music Research*, 35(4), 249–277.
- Hedges, S. 1978. Dice music in the eighteenth century. *Music & Letters*, 180–187.
- Hill, M. 2011. Beneath Clouds and The Boys: jazz artists making film music. *Screen Sound* n2, 27–47.
- Hiller, L. 1970. Music Composed with Computers: An Historical Survey. H. B. Lincoln, ed. *The Computer and Music*. Cornell University Press, 42–96.
- Hindemith, P. 1970. *The Craft of Musical Composition: Theoretical Part I*. Schott.
- Hoover, A., Szerlip, P., Stanley, K. 2014. Functional Scaffolding for Composing Additional Musical Voices. *Computer Music Journal*, 38(4), 80–99.
- Huron, D. 2006. *Sweet anticipation: Music and the psychology of expectation*. MIT Press.
- Keogh, E., Ratanamahatana, C. 2005. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3), 358–386.
- Kramer, J. 1978. Moment form in twentieth century music. *The Musical Quarterly*, 64(2), 177–194.
- Kramer, J. 1988. *The Time of Music New Meanings, New Temporalities, New Listening Strategies*. Schirmer Books.

- Kühl, O., Jensen, K. 2007. Retrieving and Recreating Musical Form. *Computer Music Modeling and Retrieval. Sense of Sounds*. Springer Berlin, 263–275.
- Lerdahl, F., Jackendoff, R. 1983. *A Generative Theory of Tonal Music*. MIT Press.
- Lester, J. 1986. Notated and Heard Meter. *Perspectives of New Music*, 24(2), 116–128.
- Lewis, G. 1999. Interacting with Latter-Day Musical Automata. *Contemporary Music Review* 18(3), 99–112.
- Livingstone, S., Muhlberger, R., Brown, A., Thompson, W. 2010. Changing Musical Emotion: A Computational Rule System for Modifying Score and Performance. *Computer Music Journal*, 34(1), 41–64.
- Maconie, R. 1976. *The Works of Karlheinz Stockhausen*. Oxford University Press.
- Meyer, L. 1956. *Emotion and Meaning In Music*. University of Chicago Press.
- Miranda, E. 2001. *Composing music with computers*. CRC Press.
- Narmour, E. 1990. *The Analysis and Cognition of Basic Melodic Structures*. University of Chicago.
- Narmour, E. 1991. The Top-down and Bottom-up Systems of Musical Implication: Building on Meyer's Theory of Emotional Syntax. *Music Perception: An Interdisciplinary Journal*, 9(1), 1–26.
- Nattiez, J. 1990. *Music and discourse: Toward a semiology of music*. Princeton University Press.
- Negretto, E. 2012. *Expectation and anticipation as key elements for the constitution of meaning in music*. *Teorema* 31(3), 149–163.
- Nierhaus, G. 2009. *Algorithmic composition: paradigms of automated music generation*. Springer Science & Business Media.
- Nyman, M. 1999. *Experimental music: Cage and beyond*. Cambridge University Press.
- Pachet, F. 2004. Beyond the cybernetic jam fantasy: The continuator. *Computer Graphics and Applications*, IEEE, 24(1), 31–35.
- Parks, R. 1989. *The Music of Claude Debussy*. Yale University Press.
- Pasquier, P., Eigenfeldt, A., Bown, O., Dubnov, S. 2016. An Introduction to Musical Metacreation. *Special Issue: Musical Metacreation*, ACM Computers In Entertainment, forthcoming.
- Pearce, M., Wiggins, G. 2001. Towards a framework for the evaluation of machine compositions. *Artificial Intelligence and Creativity in the Arts and Sciences*, 22–32.
- Pearce, M., Wiggins, G. 2007. Evaluating cognitive models of musical composition. *International Joint Workshop on Computational Creativity*. University of London, 73–80.
- Pérez, R., Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2), 119–139.
- Prokopenko, M. (2009). Guided Self-Organization. *HFSP Journal*, 3(5), 287–289.
- Rowe, R. 1992. *Interactive music systems: machine listening and composing*. MIT Press.
- Rowe, R. 2004. *Machine musicianship*. MIT press.
- Salzer, F. 1962. *Structural hearing: Tonal Coherence in Music*. Courier Corporation.
- Schedel, M., Rootberg, A. 2009. Generative Techniques in Hypermedia Performance. *Contemporary Music Review*, 28(1), 57–73.
- Schenker, H. 1972. *Harmony*. University of Chicago Press.
- Schoenberg, A., Stein, L. 1970. *Fundamentals of musical composition*. Faber & Faber.
- Shannon, C., Weaver, W. 1949. *The mathematical theory of communication*. University of Illinois Press.
- Sloboda, J. 1991. Music structure and emotional response. *Psychology of Music*, 19, 110–120.
- Smith, B., Garnett, G. 2012. Improvising Musical Structure with Hierarchical Neural Nets. *AIIDE*, Palo Alto, 63–67.
- Sorensen, A., Brown, A. 2008. A Computational Model For The Generation Of Orchestral Music In The Germanic Symphonic Tradition: A progress report. *Australasian Computer Music Conference*, Sydney, 78–84.
- Spector, L., Alpern, A. 1994. Criticism, culture, and the automatic generation of artworks. *National Conference on Artificial Intelligence*, Seattle, 3–8.
- Stockhausen, K. 1963. Momentform: Neue Beziehungen zwischen Aufführungsdauer, Werkdauer und Moment. *Texte zur Musik I*, 189–210.
- Suurpää, L. 2006. Form, structure, and musical drama in two Mozart expositions. *Journal of Music Theory* 50(2), 181–210.
- Tzanetakis, G., Cook, P. 2000. Marsyas: A framework for audio analysis. *Organised Sound*, 4(03), 169–175.
- Veale, T. 2012. *Exploding the creativity myth: The computational foundations of linguistic creativity*. A&C Black.
- Veale, T. 2015. Scoffing at Mere Generation. <http://prosecco-network.eu/blog/scoffing-mere-generation>, accessed Feb. 1, 2016.
- Whittall, A. 2016. Form. *Grove Music Online. Oxford Music Online*. Oxford University Press, accessed Feb. 1, 2016.
- Xenakis, I. 1992. *Formalized music: thought and mathematics in composition* (No. 6). Pendragon Press.

Generative Choreography using Deep Learning

Luka Crnkovic-Friis
Peltarion
luka@peltarion.com

Louise Crnkovic-Friis
The Lulu Art Group
louise@theluluartgroup.com

Abstract

Recent advances in deep learning have enabled the extraction of high-level features from raw sensor data which has opened up new possibilities in many different fields, including computer generated choreography. In this paper we present a system *chor-rnn* for generating novel choreographic material in the nuanced choreographic language and style of an individual choreographer. It also shows promising results in producing a higher level compositional cohesion, rather than just generating sequences of movement. At the core of *chor-rnn* is a deep recurrent neural network trained on raw motion capture data and that can generate new dance sequences for a solo dancer. *Chor-rnn* can be used for collaborative human-machine choreography or as a creative catalyst, serving as inspiration for a choreographer.

Introduction

Can a computer create meaningful choreographies? With its potential to expand and facilitate artistic expression, this question has been explored since the start of the computer age. To answer it, a good starting point is to identify the different levels that go into a choreographic work.

A choreography can be said to contain three basic levels of abstraction, *style* (the dynamic execution and expression of movement by the dancer), *syntax* (the choreographic language of the work and choreographer) and *semantics* (the overall meaning or theme that binds the work into a coherent unit) (Blacking & Kealiinohomoko, 1979). All three levels present unique practical and theoretical challenges to computer generated choreography.

As syntax is the easiest to formalize in the form of a notation system, it has been the logical starting point for creating generative choreography (Calvert, Wilke, Ryman, & Fox, 2005). However, unlike music or literature dance lacks a universally accepted notation system. Although systems, such as Benesh movement notation and Labanotation have been proposed they have not been universally adopted mostly because of their steep learning curve (Guest, 1998). They also cannot capture *style* - the nuanced dynamics of movement that emerges as a collaboration between choreographer and dancer (Blom & Chaplin, 1982). The alternative of building computational models from raw data (video, motion capture) is alluring as it contains much more information. It has however until recently not been feasible both because of combination of lack of computing power, algorithms and available data (LeCun, Bengio, & Hinton, 2015). With the advent of GPU powered deep learning that has changed and we can now start building entirely data driven end-to-end generative models that are capable of capturing both *style* and *syntax*. Furthermore, as deep neural networks

are capable of extracting multiple layers of abstraction, they can begin to model the *semantic* level as well. In this paper we describe such a system, *chor-rnn* that we have developed and discuss related work, show how the raw data is collected and present a deep recurrent neural network model. Finally, we also detail the training and discuss results, possible use and future work.

Related work

Earlier work in this field has included various programmatic approaches with parametrized skeleton systems (Noll, 2013) as well as using simplified movement models combined with genetic algorithms to explore the parameter space (Lapointe, 2005). Several systems have been developed as a combination of a visualization system with a choice of pre-defined movement material that could be sequenced into longer compositions by the choreographer. Fully autonomous sequence generation has mostly been limited to sequencing a combination of snippets of movement material. Several proposed systems have been interactive, requiring a choreographer to make a number of selections during the generation phase (Carlson, Schiphorst, & Pasquier, 2011). Artificial neural networks have been used in generative systems in the past (McCormick, 2015). They have however not involved deep learning and the neural network presented in this paper is using tens of millions of model parameters rather than thousands.

Recording movement

A choreography is the purposeful arrangement of sequences of motion. The basic building block is the change of position in a 3D space (Maletic, 1987). Techniques for recording the movement of human body in space are called “motion capture” and while there are various technical solutions at the time of writing, the most simple to use and cost effective was the Microsoft Kinect v2 sensor (Berger et al., 2011).

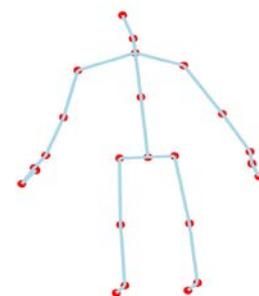


Figure 1 The red dots are joints tracked by the Kinect sensor.

It consists of a 3D camera augmented by an infrared camera and software that can automatically perform efficient and accurate joint tracking. The sensor records the movements of 25 joints (see Figure 1) at up to 30 frames per second. Each joint position is represented by a 3D coordinate for each frame. The sensor can in theory track up to 6 bodies simultaneously but multiple bodies can occlude each other relative to the field of view of the sensor. It has no way of tracking occluded joints (Fürntratt & Neuschmied, 2014). Multiple sensors can be used to overcome that limitation, but it requires more complex software to combine the results (Kwon et al., 2015). Our work was done with one sensor and one body.

Generative model

Recurrent neural networks (RNNs) have been used to get state-of-the-art results for complex time series modeling tasks such as speech recognition and translation (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2015). Since the motion capture data is a multidimensional time series we use a deep RNN model.

Long Short-Term Memory

Standard RNNs are difficult to train in a stable way (due to the vanishing/exploding gradient problem) so we use a Long Short-Term Memory (LSTM) type of RNN. LSTMs are stable over long training runs and can be stacked to form deeper networks without loss of stability (Schmidhuber, 2015).

Contrary to a regular RNN which uses simple recurrent neurons, the central unit in an LSTM is a memory cell that holds a state over time, regulated by gates that control the signal flow in and out of the cell.

As the signal flow is tightly controlled, the risk is minimized of overloading the cell through positive feedback or extinguishing it through negative feedback.

The following equations show the relations in an LSTM cell (see Figure 2):

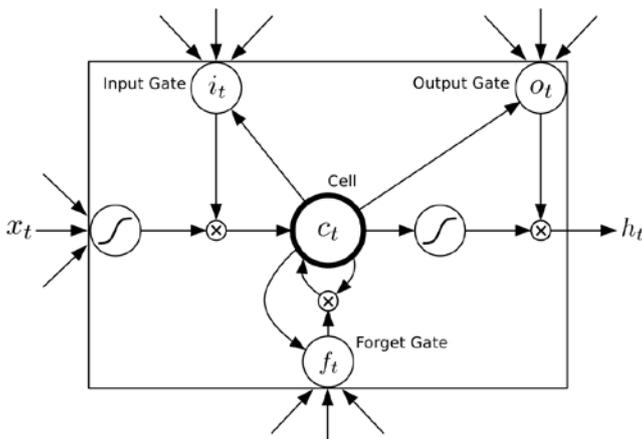


Figure 2 LSTM neuron

$$i_t = \delta(W^i x_t + R^i h_{t-1} + p^i \odot c_{t-1} + b^i) \quad (1)$$

$$f_t = \delta(W^f x_t + R^f h_{t-1} + p^f \odot c_{t-1} + b^f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} \odot g(W^c x_t + R^c h_{t-1} + b^c) \quad (3)$$

$$o_t = \delta(W^o x_t + R^o h_{t-1} + p^o \odot c_{t-1} + b^o) \quad (4)$$

$$h_t = \delta(W^h x_t + R^h h_{t-1} + p^h \odot c_{t-1} + b^h) \quad (5)$$

Here i_t , f_t , c_t , o_t and h_t are the input gate, forget gate, memory cell and output gate at time step t ; x_t is the input while $\delta()$ and $g()$ are the sigmoid and tangent activation functions; W and R are the weight matrices applied to input and recurrent units; p and b are the peep-hole connection and biases while \odot denotes dot product. Typically, when training a generative system, target output data would be the same as the input data but shifted with one sample:

$$t_t = x_{t+1} \quad (6)$$

This works well when the input and targets are discrete and the last layer is a softmax function (such as in the case of words or characters). For continuous functions as in this case there is a fundamental problem. When sampling dance movements 30 times/second x_t is trivial to predict if x_{t-1} and x_{t-2} are known. It is just a continuation of the previous vector. A very simple model will produce very low errors during training, validation and testing:

$$x_t = x_{t-1} + (x_{t-1} - x_{t-2}) \quad (7)$$

However, when using it in a generative fashion where the output of the LSTM is used as the next input

$$y_t = LSTM_t(x_{t-1}) \quad (8)$$

it will fail completely. In cases where the data is discrete, a softmax introduces a controlled random element that can force the trajectory of the network into a new but controlled direction. In the case of continuous data, it is not possible as we do not have a controlled statistical distribution of the output so adding random noise will not help. In general, it can be shown that when using a mean square error metric, the output will stagnate and converge to an average output (Bishop, 1994).

Mixture Density LSTMs

To counteract the issue of stagnating output we attach a mixture density network (MDN) to the output of the LSTM. This technique has been used successfully among other things for robotic arm control (Bishop, 1994) as well as handwriting generation (Graves, 2013).

Instead of just outputting a single position tensor, we output a probability density function for each dimension in the tensor. The output of the LSTM consists of a layer of linear

output units that provide parameters for a mixture model defined as the probability of a target \mathbf{t} given an input \mathbf{x} :

$$p(\mathbf{t}|\mathbf{x}) = \sum_{i=1}^m \alpha_i(\mathbf{x})\varphi_i(\mathbf{t}|\mathbf{x}) \quad (9)$$

where m is the number of components in the mixture with α_i being the mixing coefficients as a function of the inputs (\mathbf{x}). The function φ_i is the conditional density of the target tensor \mathbf{t} for the i :th kernel. We use a Gaussian kernel, defined as:

$$\varphi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{\frac{c}{2}}\sigma_i(\mathbf{x})^c} e^{-\frac{\|\mathbf{t}-\mu_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2}} \quad (10)$$

The neural network part that feeds into the mixture density model hence provides a set of values for the mixture coefficients, a set of means and a set of variances. The total number of output variables will be $m(c+2)$ where m is the number of mixture components and c the number of output variables (a regular LSTM would have c outputs).

The outputs from the neural network will consist of a tensor

$$\mathbf{z} = [z_1^\alpha, \dots, z_m^\alpha, z_{m+1}^\mu, \dots, z_{m+c+1}^\mu, z_{m+c+2}^\sigma, \dots, z_{m(c+2)}^\sigma] \quad (11)$$

containing all the necessary parameters to construct a mixture model. The number of mixture components, m , is arbitrary and can be interpreted as the number of different choices the network can pick at each time step. With the parametrized output, the whole MDN part can be encoded as a simple error metric where the error function becomes a negative log likelihood function (for the q :th sample):

$$E^q = -\log \left[\sum_{i=1}^m \alpha_i(\mathbf{x}^q)\varphi_i(\mathbf{t}^q|\mathbf{x}^q) \right] \quad (12)$$

Where

$$\alpha_i = \frac{e^{z_i^\alpha}}{\sum_{j=1}^m e^{z_j^\alpha}}, \quad \sigma_i = e^{z_i^\sigma}, \quad \mu_i = z_{ik}^\mu \quad (13)$$

and the derivatives needed for the training can be expressed as:

$$\frac{\partial E^q}{\partial z_k^\alpha} = \alpha_k - \frac{\alpha_k \varphi_k}{\sum_{j=1}^m \alpha_j \varphi_j} \quad (14)$$

$$\frac{\partial E^q}{\partial z_{ik}^\mu} = \frac{\alpha_i \varphi_i}{\sum_{j=1}^m \alpha_j \varphi_j} \frac{\mu_{ik} - t_k}{\sigma_i^2} \quad (15)$$

$$\frac{\partial E^q}{\partial z_i^\sigma} = -\frac{\alpha_i \varphi_i}{\sum_{j=1}^m \alpha_j \varphi_j} \left\{ \frac{\|\mathbf{t} - \mu_i\|^2}{\sigma_i^2} - c \right\} \quad (16)$$

The derivatives of the error function can be used with any standard gradient based optimization algorithm together with backpropagation.

Training

The data collected consisted of five hours of contemporary dance motion capture material created and performed by a choreographer. The resulting data set consisted of 13.5 million spatiotemporal joint positions. We used multiple deep configurations but the final neural network topology consists of 3 hidden layers with 1024 neurons in each (a total of ~21M weights). The input data was a 75-dimensional tensor (25 joints x 3 dimensions).

The model was trained for ~48h on a GPU computation server with 4 x Nvidia Titan X GPUs (a total of 27 teraflops capacity). A batch size of 512 sequence parts (128/GPU) was used with a sequence length of 1024 samples. The sequence length corresponds to how many steps the system is unrolled in time and in effect the number of layers becomes $1024*3 = 3072$ during training.

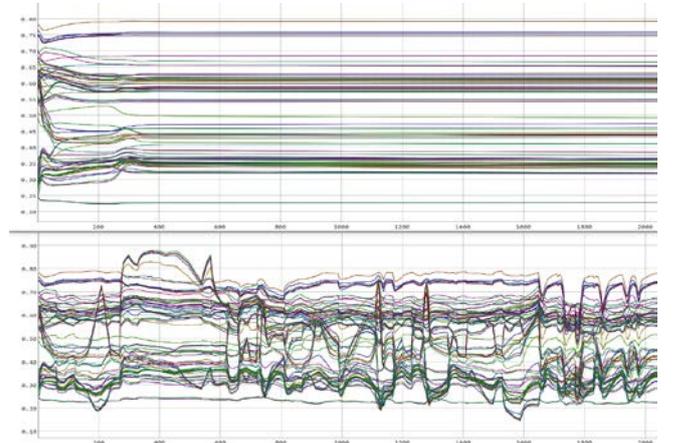


Figure 3 Output of a minute of generated joint positions over time: Without mixture density (top) and with mixture density (bottom)

The number of layers, and their effect in a recurrent neural network requires a far more complex interpretation than standard feed forward/convolutional neural networks as a signal can take an indeterminate number of spatiotemporal paths through the network. (Greff et al., 2015)

The neural network was trained with RMS Prop using Back Propagation Through-Time. The software was implemented in lua/Torch7 using the Peltarion Cortex platform. A comparison of a network trained with MDN and without can be seen in Figure 3.

In generation mode the MDN distributions were sampled at each time step to get a new set of coordinates for the joints. For this experiment we used unbiased sampling.

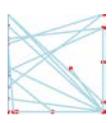
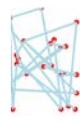
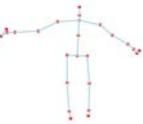
Training Time	Sample frames from generated animation			Description
~10 min				Nearly untrained system. Joint positions are almost random. https://www.youtube.com/watch?v=QnaKyc1Mpmc
~6h				Understands relative joint positions and very basic movement. https://www.youtube.com/watch?v=c9h9zc7uPWQ
~48h				Understands joint relations well, understand syntax and style well, understands basic semantics https://www.youtube.com/watch?v=Q4_XSMqN8w0 https://www.youtube.com/watch?v=W1oRgDPxEkc

Table 1 Example results over time

Results

Our *chor-rnn* system can produce novel choreographies in the general style represented in the training data. Over the training interval it passes through several stages: basic joint relations (understanding the anatomy of human joints), basic movement style and syntax and at last the composition of several movements into a meaningful composition (semantics). See Table 1 for examples of results and Figure 4 for a visualization of example generated trajectories.

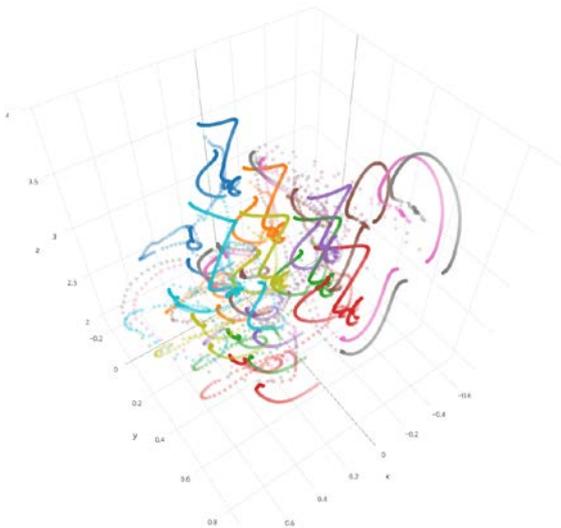


Figure 4 Spatial visualization of 30 seconds of generated trajectories for 10 joints. Each color represents a joint.

Discussion

The generated material, presented as an animated “stick figure”, was evaluated qualitatively by the choreographer. As choreography is an art largely based on physical expression and embodied knowledge (Blom & Chaplin, 1982), the choreographer also learned and executed the generated material. The conclusion was that the *chor-rnn* system produces novel, anatomically feasible movements in the specific choreographic language of the choreographer whose work it was trained on. If you generated an hours’ worth of new choreography, it would have significantly less semantic meaning than the work it was trained on.

Generally speaking, the semantic level is the most difficult to quantify, especially when it comes to avant-garde art as it does not follow an established form (Foster, 1986). It is also the most complex one from the point of view of the neural network.

As with text or image generation (Hinton, 2014), the semantic level is the last one to emerge from the training.

There will of course be significant limitations when it comes to generating novel semantic levels as an artificial neural network can’t draw on the human choreographer’s life experience.

Use as an artist’s tool

While there are interesting philosophical questions regarding machine creativity especially in a longer perspective, it is also interesting to see how current results can be used as a practical tool for a working choreographer. The *chor-rnn* system in its current state can be used to facilitate a human choreographer’s creative process in several ways. Two examples are collaborative choreography and mutually interpreted choreography.

In the first case the artist and *chor-rnn* can collaborate in creating a sequence by alternating between them as shown in Figure 5.

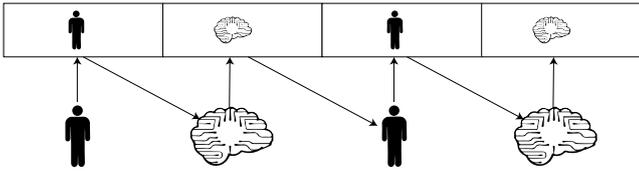


Figure 5 Alternating artist /chor-rnn choreography

1. The artist choreographs a sequence A_1
2. Chor-rnn takes sequence A_1 as input and produces a new sequence B_1 as a continuation of A_1
3. The artist looks at sequence B_1 and choreographs a new sequence A_2 as a continuation of B_1
4. Steps 2-3 are repeated

The resulting sequence will be $A_1B_1A_2B_2..A_NB_N$ – an alternation between human and computer choreography.

In the second case, the artist can start a sequence, let the *chor-rnn* generate a new sequence. The human can then reinterpret the output of the *chor-rnn* and input the interpretation into the system.

1. The artist choreographs a sequence A_1
2. Chor-rnn takes sequence A_1 as input and produces a new sequence (or set of sequences) B_1 as a continuation of A_1
3. The artist looks at B_1 and choreographs a reinterpretation of B_1 as a new sequence, A_2
4. Steps 2-3 are repeated

The resulting sequence will be $A_1A_2A_3..A_N$ – a computer inspired human made choreography. Due to the symmetry of the process, a secondary sequence is created as well, $B_1B_2B_3..B_N$ – a human inspired computer made choreography.

When a choreographer works with a dancer to develop a choreography, the latter will inevitably influence the end result. While this may be desirable, it also dilutes the distinctive style (and possibly syntax) that is unique to the choreographer.

With *chor-rnn*, the choreographer works with a virtual companion using the same choreographic language. At the same time as it is capable of producing novel work, it can provide creative inspiration. As the level of machine involvement is variable and can be chosen by the choreographer, the results can be an interesting starting point of philosophical discussions on authenticity and computer generated art.

Future work

Collect a larger corpus of data The five hours of motion capture data was enough to build a proof of concept system but ideally the corpus should be larger – especially if multiple choreographers are involved. For comparison state of the art speech recognition models use 100+ hours of data (and

it is considered to be a major bottleneck in that field of research) (Graves & Jaitly, 2014).

Derive a choreographic symbolic language One of the most intriguing features of deep neural networks is that they internally build up multiple levels of abstraction (Hinton, 2014). Using a recurrent variational autoencoder would allow us to compress meaningful higher order information into a fixed size tensor (encoding) (Sutskever, Vinyals, & Le, 2014). This in turn would allow a derivation of a symbolic language and by mapping it to feature detectors that operate on that encoding.

A general symbolic encoding could provide an alternative to the existing notation systems and simplify the creation of computer created choreography. It could also provide a convenient method of recording a choreographic work in a compact, human readable format. As multiple mobile phone makers are now integrating 3D cameras (comparable to the Kinect) into their devices, an easy way of transforming recorded material to a symbolic encoding may be of significant practical use for documentation/archiving purposes (Kadambi, Bhandari, & Raskar, 2014).

Multiple bodies The Kinect sensor cannot directly handle occluded body parts. This is problematic even with one dancer and makes it nearly impossible to capture interactions between multiple dancers. The solution is to use multiple Kinect sensors and combine their data (Kwon et al., 2015). This would allow us to record choreographies with up to 6 dancers and allow the system to learn about interactions between dancers.

Multi-modal input The input data could be extended to beyond motion capture data also include sound (and even images and video). One could for instance build a system that in the generated choreography relates to a musical composition.

Conclusions

This paper details a system, *chor-rnn* that is trained using a corpus of motion captured contemporary dance. The system can produce novel choreographic sequences in the choreographic style represented in the corpus. Using a deep recurrent neural network, it is capable of understanding and generating choreography *style*, *syntax* and to some extent *semantics*. Although it is currently limited to generating choreographies for a solo dancer there are a number of interesting paths to explore for future work. This includes the possibility of tracking multiple dancers and experimenting with variational autoencoders that would allow the automatic construction of a symbolic language for movement that goes beyond simple syntax. Apart from fully autonomous operation, *chor-rnn* can be used by a choreographer as a creativity catalyst or choreographic partner.

We asked if a computer could create meaningful choreographies and with tools like *chor-rnn* we think we can get one step closer to answering that question or at least to discover new relevant questions.

References

- Berger, K., Ruhl, K., Schroeder, Y., Bruemmer, C., Scholz, A., & Magnor, M. A. (2011). *Markerless Motion Capture using multiple Color-Depth Sensors*. Paper presented at the VMV.
- Bishop, C. M. (1994). Mixture density networks.
- Blacking, J., & Kealiinohomoko, J. W. (1979). *The Performing arts: music and dance*: Walter de Gruyter.
- Blom, L. A., & Chaplin, L. T. (1982). *The intimate act of choreography*: University of Pittsburgh Pre.
- Calvert, T., Wilke, L., Ryman, R., & Fox, I. (2005). Applications of computers to dance. *Computer Graphics and Applications, IEEE, 25*(2), 6-12.
- Carlson, K., Schiphorst, T., & Pasquier, P. (2011). *Scuddle: Generating movement catalysts for computer-aided choreography*. Paper presented at the Proceedings of the Second International Conference on Computational Creativity.
- Foster, S. L. (1986). *Reading dancing: Bodies and subjects in contemporary American dance*: Univ of California Press.
- Fürntratt, H., & Neuschmied, H. (2014). *Evaluating pointing accuracy on Kinect V2 sensor*. Paper presented at the International Conference on Multimedia and Human-Computer Interaction (MHCI).
- Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Graves, A., & Jaitly, N. (2014). *Towards end-to-end speech recognition with recurrent neural networks*. Paper presented at the Proceedings of the 31st International Conference on Machine Learning (ICML-14).
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2015). LSTM: A search space odyssey. *arXiv preprint arXiv:1503.04069*.
- Guest, A. H. (1998). *Choreo-graphics: a comparison of dance notation systems from the fifteenth century to the present*: Psychology Press.
- Hinton, G. (2014). Where do features come from? *Cognitive science, 38*(6), 1078-1101.
- Kadambi, A., Bhandari, A., & Raskar, R. (2014). 3d depth cameras in vision: Benefits and limitations of the hardware *Computer Vision and Machine Learning with RGB-D Sensors* (pp. 3-26): Springer.
- Kwon, B., Kim, D., Kim, J., Lee, I., Kim, J., Oh, H., . . . Lee, S. (2015). Implementation of Human Action Recognition System Using Multiple Kinect Sensors *Advances in Multimedia Information Processing--PCM 2015* (pp. 334-343): Springer.
- Lapointe, F.-J. (2005). *Choreogenetics: The generation of choreographic variants through genetic mutations and selection*. Paper presented at the Proceedings of the 7th annual workshop on Genetic and evolutionary computation.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.
- Maletic, V. (1987). *Body-space-expression: The development of Rudolf Laban's movement and dance concepts* (Vol. 75): Walter de Gruyter.
- McCormick, J. H. S., Vincs, Kim, Vincent, Jordan Beth (2015). *Emergent Behaviour: Learning From An Artificially Intelligent Performing Software Agent*. Paper presented at the ISEA 2115.
- Noll, A. M. (2013). EARLY DIGITAL COMPUTER ART & ANIMATION AT BELL TELEPHONE LABORATORIES, INC.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85-117.
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to sequence learning with neural networks*. Paper presented at the Advances in neural information processing systems.

Investigating the Musical Affordances of Continuous Time Recurrent Neural Networks

Steffan Ianigro

Architecture, Design and Planning,
University of Sydney,
Darlington NSW 2008, Australia
steffanianigro@gmail.com

Oliver Bown

Art and Design,
University of New South Wales,
Paddington NSW 2021, Australia
o.bown@unsw.edu.au

Abstract

This paper investigates the musical affordances of Continuous Time Recurrent Neural Networks (CTRNNs) as an evolvable low-level algorithm for the exploration of sound. Our research will be divided into two parts. Firstly, we will conduct various studies that provide insight into CTRNN behaviours, identifying aspects that could prove creatively valuable to musicians. We expect to find that this system will exhibit musical behaviours reminiscent of conventional audio processing methods such as amplitude modulation and additive synthesis, as well as producing interesting temporal structures. In the second part of this paper, we will discuss how these interesting behaviours can be harnessed by musicians. Specifically we investigate how evolutionary search can be used to exploit the compact low-level structure of CTRNNs and explore their potential for audio diversity beyond the capabilities of more traditional methods of audio exploration.

Introduction

Evolutionary Algorithms (EAs) have been widely explored as tools for musical composition, demonstrated in the survey by Husbands et al. (2007). EAs are highly abstract biological models and provide an effective search heuristic for solving complex problems. Of particular interest to the authors are EAs used as creative tools for the exploration of audio synthesis algorithms such as described in (Yee-King and Roth 2008; Dahlstedt 2007). Despite the success of these systems, the question of what audio representations maximise both exploitability and variety is still a debated issue. For example, McCormack (2008) emphasises the potential for creativity afforded by searching low-level structures for creative artefacts, such as manipulating pixels of an image in search of interesting artworks. However, McCormack also identifies the futility of searching these low-level representations, as although they may be capable of extensive diversity, artefacts of any interest will take an impractical amount of time to find. This example refers to brute force random search, but the same notion is true when evolving low-level audio representations, such as manipulating individual samples of an audio waveform to synthesise interesting sounds. We could evolve almost any audio possibility, but if the genetic representation is too broad, the vastness of the system's

search space would render many of these creative prospects unreachable.

In pursuit of more explorable audio representations, many authors of EAs embed higher-level software structures within their creations that they think will yield interesting results (McCormack 2008), constraining the system's creative search space within manageable limits. For example, when evolving the parameters of a commercial synthesiser, a high-level of abstraction or a meta relationship exists between the parameters that are being manipulated by the EA (genotype) and the resulting audio produced by the structure of the device (phenotype). As a result, the system's output is constrained by the capabilities of its components as individuals produced by the system will exhibit strong traits of the underlying formalised structures that created them. This means that the outputs of the system will be of a specific 'class', defined by the audio representation or parameterisations selected by the system's creator (McCormack 2008). This reduction of the audio search space means that the system is more manageable to explore and thus creatively useful, a solution that may prove sufficient if a user just wants to explore permutations of an existing system, but what if a user seeks audio with greater spectral complexity or variety beyond the capabilities of the plethora of music-making devices at their disposal?

Within this paper, we propose evolving Continuous Time Recurrent Neural Networks (CTRNNs) as an alternative, providing a low-level audio representation with a compact explorable genotype structure, capable of exhibiting complex dynamics that could afford interesting sonic possibilities that are otherwise hard to achieve using more conventional synthesis approaches. However, discovery of these complex dynamics can be problematic, as although there is much research on CTRNNs covering a range of domains, little is known about their behaviours when used as audio synthesis mechanisms, raising questions such as: how do CTRNN parameter changes translate to the audio domain?; do CTRNNs behave similarly to more conventional audio synthesis mechanisms?; and how can users effectively discover their scope of audio possibilities?

We aim to address these questions through two empirical investigations. In the first part of this paper, we will conduct various CTRNN studies in an attempt to discover and understand behaviours that may prove creatively valu-

able to a musician. We have identified four interesting dynamics: the introduction of temporal and pitch structures; a strong relationship between CTRNN inputs and outputs; amplitude modulation characteristics; and additive synthesis capabilities. An online interactive appendix of selected figures from these studies can be found at www.plecto.io/ICCC2016appendix. In the second part of this paper, we discuss how evolution can be used to discover and shape CTRNN behaviours, allowing musicians to harness their idiosyncratic dynamics. Specifically, we ask questions about what types of EA structures will afford effective creative search of their parametric space and propose future research directions for implementing CTRNNs as evolvable structures for audio exploration.

Background

Related Work

Artificial Neural Networks (ANNs) have been used for many different functions in music, from beat tracking algorithms (Lambert, Weyde, and Armstrong 2015) to artificial composers that can extract stylistic regularities from existing pieces and produce novel musical compositions based on these learnt musical structures (Mozer 1994). Bown and Lexer (2006) offer another application, proposing the use of CTRNNs to create autonomous software agents that exhibit musicality. Bown and Lexer also outline the possibility of using CTRNNs as audio synthesis algorithms, a prospect which inspired this research.

A notable example of similar work is discussed by Ohya (1995), who describes a system that trains a Recurrent Neural Network to match an existing piece of audio. The network structure can then be manipulated to synthesise variants of the original sound. Eldridge (2005) provides another example, exploring the use of Continuous Time Neural Models for audio synthesis. In previous work (Ianigro and Bown 2016), we propose a system that allows users to interactively evolve CTRNNs to produce aesthetically desirable outputs for use in their artistic practices. In this paper we build on our system, *Plecto*, and further investigate the behaviour of CTRNNs within the audio signal domain.

CTRNNs

CTRNNs are nonlinear continuous dynamical systems that can exhibit complex temporal behaviours (Beer 1995). They are well suited to produce audio output as various configurations result in smooth oscillations that resemble audio waveforms. They are an interconnected network of computer-modelled neurons, typically of a type called the *leaky integrator*. The internal state of each neuron is determined by the differential equation (1),

$$\tau_i(dy_i/dt) = -y_i + \sum W_{ij}\sigma(g_j(y_j - b_j)) + I_i \quad (1)$$

where τ_i is the time constant, g_i is the gain and b_i is the bias for neuron i . I_i is any external input for neuron i and W_{ij} is the weight of the connection between neuron i and neuron j . σ is a *tanh* non-linear transfer function (Bown and Lexer 2006).

The behaviour of a neuron is defined by three parameters - gain, bias and time constant - and each neuron input has a weight parameter that governs its strength over the neuron's other inputs (Bown and Lexer 2006). CTRNNs are continuous, recurrent, and due to their complex dynamics, they are often trained using an EA. For this research, we adopt a fully connected CTRNN, meaning that the neurons in the hidden layer are all connected and the input layer has a full set of connections to the hidden layer. Each hidden neuron also has a self connection, enhancing its behavioural complexity. The output or activation of each neuron is calculated using a *tanh* transfer function, providing outputs between -1 and 1 for use as samples in an audio wavetable (the CTRNN output is the activation of a selected hidden neuron).

Evolutionary Search

Many EAs are based on Darwinian theory, with evolutionary change a result of the fittest of each generation surviving and passing on the traits that made them fit (Husbands et al. 2007). These algorithms provide a powerful method for searching a problem space, optimising candidates until the best solution is found. There are two main type of EAs: target based EAs which evaluate individuals according to a criterion that is encoded into the system, and interactive EAs that incorporate human evaluation as their selective pressure. The latter is often used for creative applications, with the user assuming the role of a 'pigeon breeder', acting as a selective pressure in an artificial environment (Bown 2009). This is an appealing prospect as it is difficult to define explicit fitness functions for audio phenotypes that can identify subjective creatively desirable traits (Tokui and Iba 2000). There are also many other types of EAs for creative exploration, such as the ecosystem model described in (McCormack 2001) and the use of artificial immune systems such as discussed in (Abreu, Caetano, and Penha 2016).

Evolution of Neural Networks

The growing area of neuroevolution refers to the optimisation of neural networks using EAs (Stanley and Miikkulainen 2002). This is an effective approach when training CTRNNs; unlike methods such as back-propagation in which network weights are adjusted to minimise the network error, multiple features of the network can be evolved at one time. The definition of an EA's performance criterion is also more flexible than the definition of an energy or error function (Floreano, Dürr, and Mattiussi 2008). There is a variety of work in this area, such as (Jónsson, Hoover, and Risi 2015; Hoover and Stanley 2007), describing systems that evolve neural networks in pursuit of creative artefacts. In this paper, we adopt a similar method of network optimisation, using an EA to manipulate gain, bias, time constant and weight parameters of the CTRNN that provide a compact genotype capable of producing extensive diversity. We will use an Artificial Immune System (AIS), a type of evolutionary optimisation algorithm called *opt-aiNet* (Timmis and Edmonds 2004) to achieve this.

Musical Behaviours of CTRNNs

In this section, we will conduct CTRNN studies by randomly generating network configurations, feeding various types of audio inputs into these networks and observing the results through audio analysis. This process will provide an insight into behaviours that are common within the search spaces of CTRNNs, such as if CTRNNs have dynamics similar to more conventional audio generation algorithms and how consistent these behaviours are, informing our expectations of the evolvability of audio CTRNNs.

Generation of Temporal and Pitch Structures

A contributing factor to the complexity that CTRNNs are able to produce is the presence of neuron self connections (Beer 1995). A strong self connection can dominate the neuron input, saturating the neuron by locking it in a certain state (emitting a constant output). This behaviour is analogous to an internal switch that can influence the behaviour of the rest of the network, creating interesting temporal dynamics that afford many creative possibilities such as described in (Bown and Lexer 2006). To investigate the audio synthesis implications of this behaviour, we adjusted the hidden neuron self connection weight of a CTRNN with one input neuron and one hidden neuron whilst feeding in the audio sample notated in Figure 1. This experiment produced some interesting results, as once the hidden neuron weight passed a certain threshold, the CTRNN alternated between states of saturation (outputs a constant value of -1 or 1) and oscillation. This is evident in Figure 2, showing the original unprocessed audio which was used as the input for the CTRNN (top) and the processed CTRNN output (bottom). In the case of this exploration, the CTRNN primarily responded to the amplitude fluctuations caused by the kick drum. However, as its hidden neuron self connection is adjusted, the degree of saturation changes, exhibiting graceful degradation of the behaviour, almost like tuning the sensitivity of a conventional signal gate used in audio production.



Figure 1: Notated version of the CTRNN input used to produce Figures 2, 3 and 4. The Drum Kit consists of a bass drum (open note head) and a hi-hat (cross note head). The synthesiser has a timbre very similar to a sine wave.

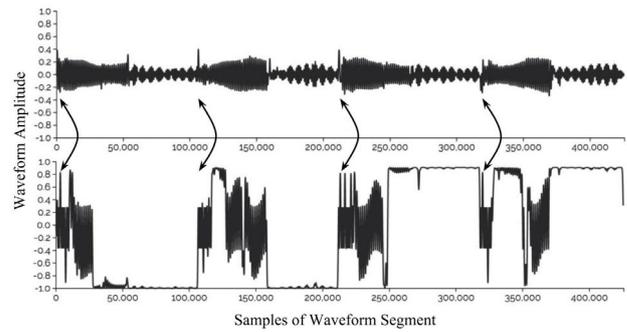


Figure 2: Comparison of unprocessed input notated in Fig. 1 (top) to its processed counterpart (bottom), showing how certain input values saturate the CTRNN's hidden neuron and others do not.

We further tested the consistency of this behaviour by adopting another input for the CTRNN that exhibits a more sporadic melody absent of any percussion. We were able to achieve a similar result by tweaking the CTRNN's hidden neuron self connection weight parameter until we observed similar saturation fluctuations. Furthermore, we found that if sinusoid inputs are adopted instead of more complex audio samples, the results are less interesting as the neuron saturates and remains so, emphasising that the continuous flipping of neurons is caused by amplitude fluctuations in the neuron input. But what are the musical implications of this behaviour in larger CTRNN structures?

In Figure 3, we can see the output of a larger CTRNN configuration (one input neuron and three hidden neurons) with an input of a small audio sample consisting of a two note melody and rhythmic accompaniment (notated in Fig. 1). The CTRNN's original input melody is evident in part one (each bar consisting of notes D and G), however, in part two we can see the introduction of new melodic and rhythmic content (notes D, C, A, G and F). The timbre of the CTRNN's input also varies and the synthesiser's percussive accompaniment is removed. Through further randomisation of the CTRNN's parameters, we found another CTRNN configuration that produces similar behaviour (output notated in Fig. 4), identifying that this introduction of temporal and pitch structures is not a one off occurrence but can occur in various forms within the CTRNN search space. These larger CTRNN outputs have similarities to the neuron saturation behaviour described earlier. For example, if we compare both Figures 3 and 4 to their original input material (Fig. 1), we can see that these introduced temporal and pitch structures coincide with the rhythmic events in the CTRNN's input material, such as when the hi-hat symbol is struck. Therefore, it appears that CTRNN input amplitude fluctuations are flipping neuron states within the network, shifting the musical structure of the CTRNN's output. This is a more complex manifestation of the behaviour seen in Figure 2 and has many creative implications, affording a means to generate temporal and pitch structures. Furthermore, the complex neuron interactions within CTRNNs can produce unexpected outputs, exhibiting agency or Musical Metacreativity (Eigenfeldt et al. 2013) during the composi-

tional process.

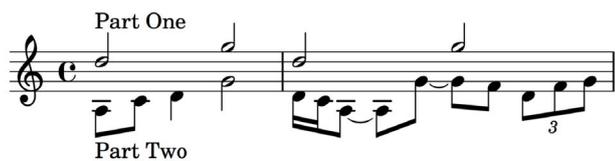


Figure 3: Simplified notation of CTRNN's output (input notated in Fig. 1). We transposed the melody up one octave for legibility and the frequencies produced by the CTRNN are converted to their closest equal tempered note values.



Figure 4: Simplified notation of CTRNN's output (input notated in Fig. 1). We transposed the melody up one octave for legibility and the frequencies produced by the CTRNN are converted to their closest equal tempered note values.

Strong Input/Output Relationship

The strong relationship we have observed between CTRNN inputs and outputs highlight that CTRNN behaviour can be similar to that of a modular synthesiser or digital signal processing (DSP) effects module, altering their input structure towards creatively exciting directions. Further evidence of this dynamic can be seen in the pitch structure of both Figures 3 and 4, with the additional note values exhibiting similar pitch structures to sections of a harmonic series based the CTRNN's input melody. For example, in Figure 3, when the note D of the input melody is playing, we also hear a counter melody consisting of A, C and D, which are the 6th, 7rd and 8th overtones of a harmonic series with a fundamental of D. This DSP effect-like dynamic could afford some interesting possibilities for the use of CTRNNs as building blocks within a larger, modular system, a possibility we will further discuss in the final section of this paper.

Amplitude Modulation

Through further analysis of the CTRNN configurations that produced Figures 3 and 4, we noticed some CTRNN behaviour similar to that produced by an amplitude modulation synthesis algorithm. This form of audio modulation follows a general rule that if two signals are multiplied, two partials result (called sidebands), one at the sum of the two frequencies and one at the difference (Puckette 2007). We can see evidence of this behaviour in both Figures 5 and 6, displaying spectrogram outputs of the same CTRNNs that produced Figures 3 and 4, except a sinusoid waveform oscillating at 523Hz was used as their inputs instead of the more complex input notated in Figure 1. Sideband structures are evident around the CTRNN input frequencies in both figures

at ratios typical of amplitude modulation. This behaviour is interesting as we can see CTRNNs are not only an evolvable structure capable of generating interesting rhythmic and pitch structures, but afford possibilities for timbral variation. Puckette (2007) also identifies that amplitude modulation can be used as an octave divider, offering a possible explanation for the overtone structures that appear below their fundamental frequencies in Figures 3 and 4.

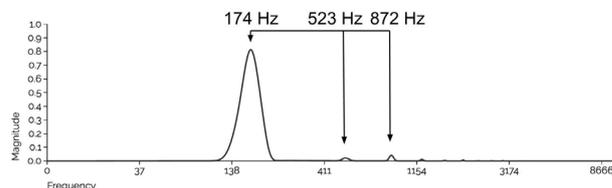


Figure 5: Sidebands that correspond to the multiplication of the sinusoid CTRNN input oscillating at 523Hz and a frequency of 349Hz. Frequency values are approximate (+-3Hz).

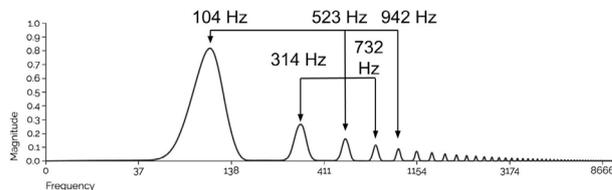


Figure 6: Sidebands that correspond to the multiplication of the sinusoid CTRNN input oscillating at 523Hz and a frequency of 419Hz as well as multiplication with a frequency of 209Hz. Frequency values are approximate (+-3Hz).

Additive Synthesis

Neurons within CTRNNs can also oscillate at fixed frequencies independent of their input. In larger CTRNNs, a dynamic analogous to additive synthesis (Puckette 2007) can result in which neuron oscillations are summed with either other neuron or CTRNN input oscillations to produce a more complex audio waveform. Figure 7 shows this relationship, exhibiting the CTRNN's sinusoid input oscillating at 523Hz (top) which appears to be summed with a lower frequency (caused by neuron oscillations at 86Hz (+-3Hz) within the CTRNN), producing a multi-phonetic CTRNN output (bottom). These summed frequencies in the CTRNN's output can also change independently of each other, evident in Figure 8. At the top, we can see a spectrogram produced by the same CTRNN that produced Figure 7 when fed a sinusoid input oscillating at 523Hz. The bottom also shows a spectrogram produced by this CTRNN except we used a sinusoid oscillating at 1000Hz as its input. We can see the presence of the same neuron oscillations at about 86Hz in both spectrograms, however the other dominant oscillations present in the CTRNN outputs vary in regard to the CTRNN's input frequency. It is worth noting that if the CTRNN's sinusoid input oscillates at a rate below about 375Hz, we lose this

additive synthesis behaviour, demonstrating the non-linear dynamics of CTRNNs.

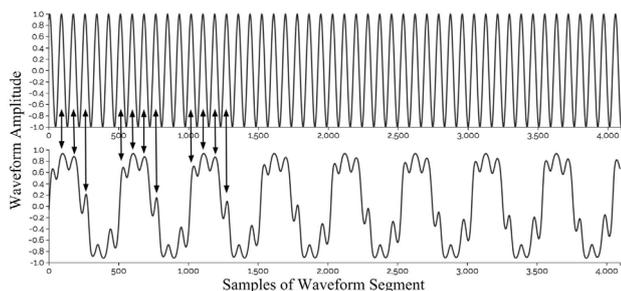


Figure 7: Comparison of a sinusoid oscillating at 523Hz (top) to the CTRNN's output it produced (bottom), demonstrating CTRNN additive synthesis capabilities.

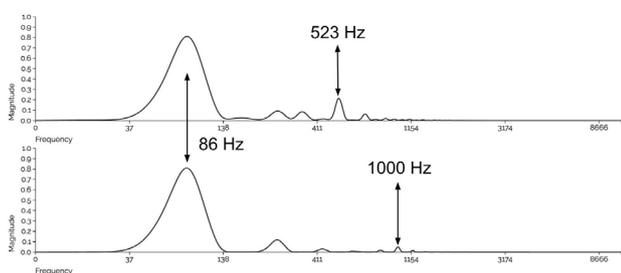


Figure 8: Comparison of spectrograms produced by a CTRNN with different sinusoid inputs (top: 523Hz, bottom: 1000Hz) showing the independent relationship between the CTRNN neuron oscillations (at about 86Hz) and the CTRNN's input.

Through these studies, we have found the occurrence of multiple musically interesting audio synthesis characteristics of CTRNNs. The variety of behaviours we have observed also hint at the generality or scope for audio variety that CTRNNs are capable of producing. However, their complex dynamics raise many questions about how to discover and utilise these creative possibilities within the workflows of musicians. In the next section, we will discuss a means to explore the creative possibilities of CTRNNs using an EA.

Evolution of CTRNNs

Many different EA designs exist that have been used for creative search. Within this section, we adopt a model based on the *opt-aiNet* algorithm conceived by de Castro and Timmis (de Castro and Timmis 2002), a multimodal optimisation algorithm inspired by some of the evolutionary properties of the human immune system (de Franca, Von Zuben, and de Castro 2005). This is an appealing model for our purpose as it can maintain many candidate solutions to a problem, providing not only the global optimum but also many of the local optima in a search space (Timmis and Edmonds 2004). This method has also shown promise for use as a

sound matching utility (Abreu, Caetano, and Penha 2016), a use case we adopt within this section.

In order to both measure how effective the *opt-aiNet* EA is for searching the creative possibilities of CTRNNs, as well as further understanding the creative capabilities of CTRNNs, we conduct a sound matching experiment in which CTRNNs (with one input neuron, ten hidden neurons and a constant input value of 0) are evolved towards five different drone-like audio targets. These selected audio sample targets cover a range of timbral profiles, which if successfully matched, will identify that the low-level functionality of CTRNNs affords a varied creative search space of audio possibilities that is explorable by an EA. Additionally, we will discuss another use for the *opt-aiNet* EA structure as a Novelty Search (NS) algorithm which rewards candidates that are unique in some way compared to existing individuals (Lehman and Stanley 2008). This is an interesting prospect for exploring the sound possibilities of CTRNNs without needing a predefined target.

opt-aiNet EA Design

The *opt-aiNet* algorithm follows a general structure outlined below. Differing from more conventional EA structures, this model incorporates sub-populations, each locally optimised with the fittest individual of each sub-population added to the main population for global evaluation during each algorithm iteration. These sub-populations are generated by cloning and mutating each member of the global population, with mutation rates inversely proportionate to the parent individual's fitness. This EA model also discourages convergence on a specific area of the search space using a population suppression mechanism. Once the population stagnates (the difference between average fitness errors over time is below a predefined threshold), individuals of the global population are compared using a distance metric and individuals with a close similarity are removed (higher fitness individuals are maintained). A number of randomly generated individuals are then introduced into the population (its size can vary dynamically) to facilitate thorough exploration of the EA's search space (Timmis and Edmonds 2004).

1. Randomly initialise the population.
2. While the stopping criterion is not met, continue, else save the global population to a database.
 - I Calculate the fitness of each individual in the global population.
 - II Generate a number of clones for each individual, creating sub populations.
 - III Mutate each clone inversely proportionate to its parent's fitness (fitter individuals are mutated less).
 - IV Determine the fitness of individuals within each sub population including the parent individual and remove all but the fittest, which replaces the parent cell in the global population.
 - V Calculate the average distance from the algorithm target and if the population stagnates, continue to steps 3 else go back to step 2.
3. Re-calculate the fitness of each individual in the global population after the fittest individuals of the sub-populations replace their parents.
4. Determine the highest affinity individuals (similar phenotype) and perform population suppression to avoid redundancy whilst maintaining the fittest individuals.

- Introduce a number of randomly generated individuals and go back to step 2.

The global population is initiated with 10 individuals and 10 clones are produced for each individual. The threshold dictating the chance of mutation for each parameter is calculated according to (2)

$$a = (1/\beta) \exp(-f^*) \quad (2)$$

where β is a parameter that controls the decay of the inverse exponential function and f^* is the fitness of the parent individual normalised within the interval of $[0..1]$. The mutated parameter value is calculated according to (3)

$$C' = c + aN(0, 1) \quad (3)$$

where c is a parameter value of a parent cell, C' is the mutated parameter value, a is calculated according to (2) and $N(0, 1)$ is a Gaussian random variable with a mean of 0 and standard deviation of 1.

Fitness Function

Within this experiment, we use a multi-objective fitness function to compare CTRNN audio outputs with the EA's target audio sample. Much work exists on reducing timbral profiles to comparable dimensions for the measurement of timbral similarity such as (Carpentier et al. 2010; Abreu, Caetano, and Penha 2016), with spectral features like *Spectral Centroid* and *Spectral Spread* being commonly used metrics. Another method for measuring timbre similarly is by comparing Mel-Frequency Cepstral Coefficients (MFCCs) of two audio samples, such as discussed in (Yee-King 2011; Aucouturier and Pachet 2004). Extracting MFCCs is a single, tested descriptor for musical timbre, therefore we have adopted this measure as one of the objectives in the EA's fitness function. MFCCs are pitch independent therefore we also use the dominant frequency present in the audio spectrum as the other fitness objective. These measures are calculated from a frequency domain description of the audio being analysed, produced by applying a Fast Fourier Transform (FFT) to small windows of the audio (4096 frames with an overlap of 2048 samples) after a Hamming windowing function is applied. As we are dealing with drone-like audio samples that do not change much over time, the amplitudes of the frequency bins produced are averaged to reduce noise in the spectrum, providing a spectral description of the most consistent frequencies in the analysed audio. The dominant frequency of the audio is calculated by identifying the frequency bin with the highest magnitude and the MFCCs are calculated as described in (Yee-King 2011): the FFT magnitudes are passed through a 42 component Mel filter bank spaced in the range of 20 to 22,050Hz, the 42 outputs of which are then transformed using a Discrete Cosine Transform and all 42 coefficients are kept. The similarity error between dominant frequencies is the absolute value of their difference. The similarity error between MFCCs is calculated using a Dynamic Time Warping (DTW) Algorithm (Muda, Begam, and Elamvazuthi 2010) with a Euclidean distance metric. Individuals are ranked according to each

fitness objective and these individual ranks are summed to measure the overall fitness of the individual.

Results

For each of the five different EA targets, we ran the algorithm for 100 iterations and as seen in Figure 9, the population commonly converges before 95 iterations. The six fittest individuals within the EA's population are then saved to a database once the algorithm stopping criterion is met. The best of these candidate CTRNNs can be heard and compared with their targets in the online appendix for this paper at www.plecto.io/ICCC2016appendix.

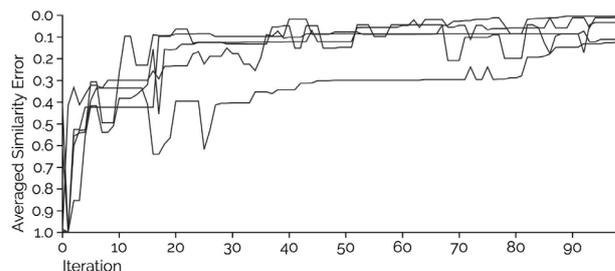


Figure 9: Graph of averaged MFCC and dominant frequency similarity errors (normalised within the interval of $[0..1]$) for each of the five algorithm runs. The best ranked individual of each algorithm iteration is displayed.

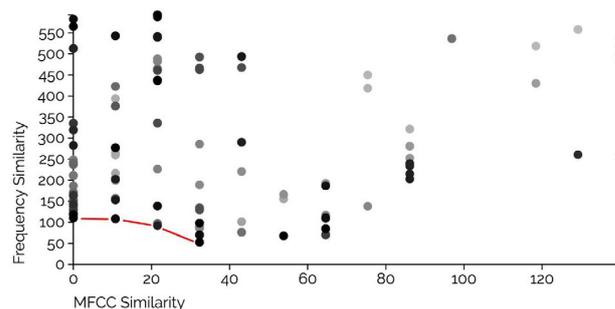


Figure 10: Pareto front of EA (bottom left hand corner) when evolving CTRNNs towards the 'Glass' audio sample. The colour of each individual communicates the EA's iteration in which the individual was produced (darkest are the final iterations).

Among these saved individuals are Pareto optimal candidates, meaning the performance of one of the individual's objectives cannot be improved without adversely affecting another objective (Van Veldhuizen and Lamont 1998). For example, Figure 10 depicts a zoomed in view of the EA's population phenotype space produced when evolving CTRNNs towards the 'Glass' audio sample. In the bottom left hand corner, we can see four Pareto optimal individuals which could all be considered to have an optimal similarity error, forming the EA's Pareto front. There are however often only between one and three individuals from each algorithm run that can be considered Pareto optimal, as there

is often a high correlation between the MFCC and the dominant frequency similarity errors when comparing CTRNN outputs with the EA's target audio.

After previewing the EA's outputs, we found that the individuals that sounded most similar to their targets were Pareto optimal solutions with the smallest similarity error between MFCCs. Furthermore, when listening and comparing the various candidate CTRNN outputs to their targets, it is evident that this EA structure is effective when evolving CTRNNs to match simple audio samples but has difficulty replicating more complex sounds, such as multi-phonetic spectral profiles. For example, when simple audio targets are used such as the 'Glass' or 'Clarinet' audio samples, the resulting CTRNN outputs exhibit strong aural similarities to their target. This contrasts to the CTRNN outputs produced when using more spectrally complex EA targets such as denser, multi-phonetic timbres. The attempt to match the 'Cello' audio sample is an example, as spectral aspects of the original recording were lost in the CTRNN outputs even though some of the pitch and general timbral characteristics were present. Matching the 'Complex Synth' audio sample resulted in similar behaviour, with the CTRNN outputs exhibiting only certain aspects of the original audio's spectral structure.

These results highlight that this EA still needs further work. For instance, it may be interesting to adopt a NEAT (NeuroEvolution of Augmented Topologies) method (Stanley and Miikkulainen 2002), meaning that the topology or structure of the CTRNN is manipulated by the EA as well as its parameters as opposed to just the gain, bias, time constant and weight parameters which formulate the genotype for the EA used in this experiment. This approach could provide a means to dynamically increase the complexity of the CTRNN's output by growing the network, removing CTRNN structural limitations when matching complex sounds. Furthermore, additional fitness objectives could be added to the EA's multi-objective fitness function to capture a greater variety of audio characteristics such as information about the change of audio over time, allowing the EA to match more dynamically varied targets such as percussive sounds. Additionally, when dealing with more complex targets, the EA's similarity errors seldom align with aural comparisons of candidate CTRNN outputs and their targets. This suggests that the extraction of MFCCs as a timbral measure either needs to be further refined or supported by other timbral comparison metrics. Nevertheless, these experiments have shown that a simple CTRNN structure can produce a range of timbres and although we have not been able to fully replicate complex sounds, we feel the EA is a good starting point in constructing an effective algorithm for the discovery of CTRNN behaviours.

Future Directions

From our observations within this paper, we believe that CTRNNs could prove valuable as a compositional aid for the discovery of interesting sounds, with their low-level functionality and compact genotype structure affording an exploratory algorithm capable of extensive audio diversity. One goal of this research is to achieve a system that enables rapid

user exploration of CTRNN audio possibilities. However, although evolving CTRNNs using the *opt-aiNet* algorithm showed promise, the process is time-consuming and will not be feasible in the creation of an engaging system that allows rapid user exploration of audio CTRNNs.

As discussed earlier, another interesting use case for the *opt-aiNet* algorithm could be for NS as we now have tested metrics for audio comparison which can be used to differentiate potential novel CTRNN candidates from existing individuals. This approach removes the need to define an explicit objective for the algorithm, simply rewarding novel finds. Therefore, an interesting design possibility could be to create a large population of small unique CTRNN modules using this method, which can be rapidly assembled by users to build more complex audio structures. This process will take advantage of the DSP effect-like dynamic that CTRNNs possess, with each CTRNN module imparting its various characteristics at each stage of a larger modular system's audio chain.

Additionally, in (Ianigro and Bown 2016), a system is described that evolves CTRNNs using an interactive EA, allowing users to select and evolve CTRNN configurations they find interesting for use within their artistic practices. The paper also identifies difficulties that arise when interactively evolving CTRNN structures, with their vast search spaces creating user fatigue and ineffective discovery of the CTRNN search space. However, if this interactive evolutionary approach is instead used to evolve combinations of higher-level CTRNN modules, a more effective system for the discovery of sound may be achieved. We aim to explore this possibility through the development of *Plecto*, a distributed composition tool that allows users to explore the creative potential of CTRNNs. The progress of this system can be monitored by visiting www.plecto.io.

Conclusion

Through this research, we conclude that CTRNNs are an effective evolvable synthesis mechanism, affording a compact genotype structure which can be manipulated to achieve vast audio diversity. We have conducted various CTRNN experiments and identified four basic musical dynamics that we believe could be conducive to interesting musical discovery: the introduction of temporal and pitch structures; a strong relationship between CTRNN inputs and outputs; amplitude modulation characteristics; and additive synthesis capabilities. We have also discussed how neuroevolution can be used to manipulate CTRNNs as a means to navigate their creative search space. However, as our current EA design can be slow when discovering ideal candidates to a creative problem, we also discuss future system designs that facilitate flexible, open ended discovery of CTRNN behaviours. Specifically, we discuss a hierarchical system, which at its base level adopts a NS adaption of the *opt-aiNet* EA to discover many small CTRNN modules, each exhibiting unique behaviours that exist within the CTRNN creative search space. At its top level, users can interactively evolve combinations of these CTRNN modules to discover audio complexity that is specific to their creative needs. In our next phase of research, we will implement this system and

conduct user studies to further investigate how the low-level dynamics of CTRNNs can be utilised as an effective creative tool that fits into the creative workflows of musicians.

References

- Abreu, J.; Caetano, M.; and Penha, R. 2016. Computer-aided musical orchestration using an artificial immune system. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer. 1–16.
- Aucouturier, J.-J., and Pachet, F. 2004. Tools and architecture for the evaluation of similarity measures: Case study of timbre similarity. In *ISMIR*.
- Beer, R. D. 1995. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior* 3(4):469–509.
- Bown, O., and Lexer, S. 2006. Continuous-time recurrent neural networks for generative and interactive musical performance. In *Applications of Evolutionary Computing*. Springer. 652–663.
- Bown, O. 2009. Ecosystem models for real-time generative music: A methodology and framework. In *International Computer Music Conference (Gary Scavone 16 August 2009 to 21 August 2009)*, 537–540. The International Computer Music Association.
- Carpentier, G.; Tardieu, D.; Harvey, J.; Assayag, G.; and Saint-James, E. 2010. Predicting timbre features of instrument sound combinations: Application to automatic orchestration. *Journal of New Music Research* 39(1):47–61.
- Dahlstedt, P. 2007. Evolution in creative sound design. In *Evolutionary computer music*. Springer. 79–99.
- de Castro, L. N., and Timmis, J. 2002. An artificial immune network for multimodal function optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, 699–704. IEEE.
- de Franca, F. O.; Von Zuben, F. J.; and de Castro, L. N. 2005. An artificial immune network for multimodal function optimization on dynamic environments. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 289–296. ACM.
- Eigenfeldt, A.; Bown, O.; Pasquier, P.; and Martin, A. 2013. Towards a taxonomy of musical metacreation: Reflections on the first musical metacreation weekend. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'13) Conference, Boston*.
- Eldridge, A. 2005. Neural oscillator synthesis: Generating adaptive signals with a continuous-time neural model.
- Floreano, D.; Dürr, P.; and Mattiussi, C. 2008. Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 1(1):47–62.
- Hoover, A. K., and Stanley, K. O. 2007. Neat drummer: Interactive evolutionary computation for drum pattern generation. Technical report, Technical Report TR-2007-03.
- Husbands, P.; Copley, P.; Eldridge, A.; and Mandelis, J. 2007. An introduction to evolutionary computing for musicians. In *Evolutionary computer music*. Springer. 1–27.
- Ianigro, S., and Bown, O. 2016. Plecto: A low-level interactive genetic algorithm for the evolution of audio. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer. 63–78.
- Jónsson, B. .; Hoover, A. K.; and Risi, S. 2015. Interactively evolving compositional sound synthesis networks. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, 321–328. ACM.
- Lambert, A. J.; Weyde, T.; and Armstrong, N. 2015. Perceiving and predicting expressive rhythm with recurrent neural networks.
- Lehman, J., and Stanley, K. O. 2008. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, 329–336.
- McCormack, J. 2001. Eden: An evolutionary sonic ecosystem. In *Advances in Artificial Life*. Springer. 133–142.
- McCormack, J. 2008. Facing the future: Evolutionary possibilities for human-machine creativity. In *The Art of Artificial Evolution*. Springer. 417–451.
- Mozer, M. C. 1994. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. *Connection Science* 6(2-3):247–280.
- Muda, L.; Begam, M.; and Elamvazuthi, I. 2010. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*.
- Ohya, K. 1995. A sound synthesis by recurrent neural network. In *Proceedings of the 1995 International Computer Music Conference*, 420–423.
- Puckette, M. 2007. The theory and technique of electronic music.
- Stanley, K. O., and Miikkulainen, R. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10(2):99–127.
- Timmis, J., and Edmonds, C. 2004. A comment on opt-ainet: An immune network algorithm for optimisation. In *Genetic and Evolutionary Computation—GECCO 2004*, 308–317. Springer.
- Tokui, N., and Iba, H. 2000. Music composition with interactive evolutionary computation. In *Proceedings of the 3rd international conference on generative art*, volume 17, 215–226.
- Van Veldhuizen, D. A., and Lamont, G. B. 1998. Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, 221–228. Citeseer.
- Yee-King, M., and Roth, M. 2008. Synthbot: An unsupervised software synthesizer programmer. In *Proceedings of the International Computer Music Conference, Ireland*.
- Yee-King, M. J. 2011. *Automatic sound synthesizer programming: techniques and applications*. University of Sussex.

How Blue Can You Get? Learning Structural Relationships for Microtones via Continuous Stochastic Transduction Grammars

Dekai Wu

HKUST

Human Language Technology Center
Department of Computer Science and Engineering
Hong Kong University of Science and Technology

dekai@cs.ust.hk

Abstract

We describe a new approach to probabilistic modeling of structural inter-part relationships between continuous-valued musical events such as microtones, through a novel class of *continuous* stochastic transduction grammars. Linguistic and grammar oriented models for music commonly approximate features like pitch using discrete symbols to represent ‘clean’ notes on scales. In many musical genres, however, contextual relationships between continuous values are essential to improvisational and accompaniment decisions—as with the ‘bent notes’ that blues rely heavily upon. In this paper, we study how stochastic transduction grammars or STGs, which have until now only been able to handle discrete symbols, can be generalized to model continuous valued features for such applications. STGs are interesting for modeling the learning of musical improvisation and accompaniment where parallel musical sequences interact hierarchically (compositionally) at many overlapping levels of granularity. Each part influences decisions made by other parts while at the same time satisfying contextual preferences across multiple dimensions; applications to flamenco and hip hop have recently been shown using discrete STGs. We propose to use a formulation of continuous STGs in which musical signals are finely represented as continuous values without crude quantization into discrete symbols, yet still retaining the ability to model probabilistic structural relations between multiple musical languages. We instantiate this approach for the specific class of stochastic inversion transduction grammars (SITGs), which has proven useful in many applications, via a polynomial time algorithm for expectation-maximization training of continuous SITGs.

Introduction

Musical improvisation is the creative activity of spontaneous, on-the-fly musical composition without prior planning, in response to a novel context (typically provided by other musicians, who are often also improvising), in contextually relevant ways that adhere to stylistic conventions, yet are not constrained by *a priori* written scores. Throughout most of history, creative improvisation has been the norm in many, if not most, traditional and folk forms of music (unlike Western music in recent centuries, where written music has been a historically recent artifact). Musical improvisation is a uniquely human behavior, exhibiting creative expression that has not been found in other “singing” species.

It can be relatively easy to construct automatic music generation algorithms that can be parametrized by various con-

ditions and constraints. On one hand, some approaches rely on manually constructed rules; these approaches can represent fairly complex kinds of structures and patterns, but the improvisation is limited to the rules that have been imagined by experts and hand coded in advance, which can only crudely be matched to true human improvisation. On the other hand, other approaches employ machine learning; these approaches attempt to match their performance more finely to human improvisation by training contextual predictors on actual music data, but improvisation tends to be restricted to what can be modeled via fairly simple representations such as HMMs to limit the complexity of the learning.

The problem is that real musical improvisation at human levels requires both complex structures and patterns, and also contextual prediction that is finely tuned to human performance data. Improvisational and accompaniment decisions in one part can be influenced strongly, or subtly, by decisions made in other parts, interacting hierarchically or *compositionally* at many overlapping levels of granularity. Improvisation and accompaniment decisions are not merely random; rather, participants understand how to communicate with each other within accepted conventions and frameworks—witness, for example, flamenco *palos*, Indian *ragas*, jazz and blues. Conventions in widespread use include tonal systems, metrical constraints, chord progressions, verse structures, rhythmic patterns, and melodic phrases that are re-used or swapped into different positions within the structures. Making improvisation decisions that integrate interacting contextual factors over many levels of granularity requires a representation that can encode such sophisticated phenomena, yet will not blow up machine learning complexity exponentially.

To attack this challenge, we are engaged in a long-term program to develop a general mathematical framework for creative improvisation, that is capable of representing a realistically broad range of the many different complex interactions among factors that should influence the improvisation, and yet which can still support efficient polynomial-time training and improvisation algorithms—so that ultimately, we should be able to build more realistic models of learning to improvise. A full solution to the representation, learning, and improvisation problems will obviously require many advances, but we have already begun to show how various aspects of these tasks can be accomplished, via bilingual **stochastic transduction grammar** or **STG** models that can

simultaneously capture contextual preferences across a wide variety of dimensions. In our work on hip hop learning models (Wu *et al.*, 2013), we showed how **stochastic inversion transduction grammars** or **SITGs** can be used to learn how to improvise responses in freestyle rap battling when confronted with arbitrary challenge raps, by learning complex relationships between challenges and responses. In our work on flamenco learning models (Wu, 2013), we showed how SITGs can be used to learn how to improvise complementary lines in, for example, *palmas* percussion in the context of perceiving *cajón* percussion, by learning complex hypermeter and rhythm biases in the relationship between the languages of different percussion instruments.

Applications like these have demonstrated how STGs (a) have the expressiveness to represent compositionally interacting factors between two different parts or instruments at many overlapping levels of granularity, (b) can be efficiently induced via the polynomial-time learning algorithms that exploit the combinatorial structure of SITGs, and (c) can then use the learned knowledge representation to creatively perform real-time improvisational expression. For capturing the complexity of hierarchical structural relationships between different musical languages, the linguistic bilingual approaches of STGs have many appealing properties. They allow idiomatic constructs of significant complexity to be encoded. They allow biasing of probabilities from many different contextual features. They allow idiomatic constructs to be combined in creative new ways inspired by the unplanned contextual factors. They accommodate correlations that are not necessarily aligned in time, which make them significantly more expressive than context-free grammars (CFGs); this is why the basic time complexities for stochastic CFG recognition and training are $O(n^3)$, in contrast to $O(n^6)$ for SITGs. Musical improvisation modeling approaches based on SITGs benefit from leveraging several decades of advances in the field of statistical machine translation, which exhibits very analogous challenges.

However, all SITG based models to date over the past two decades have exhibited a glaring weakness when it comes to learning creative improvisation knowledge in the domain of music: they are only capable of representing sequences of discrete symbolic events. This was not an obstacle in the rap battle improvisation domain, where words and phrases were modeled by discrete symbols. Likewise, it was not an obstacle in the flamenco improvisation domain, where each percussive event was modeled by a discrete symbol. However, this is a major limitation in the music domain in general, where the overwhelming majority of events are continuous values like pitch, timbre, or volume.

This paper proposes for the first time a formulation of SITGs that (a) have the expressiveness to represent compositionally interacting factors between different *continuous valued* parts or instruments at many overlapping levels of granularity, and yet (b) can still preserve all the aforementioned advantages of SITGs, including efficiently induction via the polynomial-time learning algorithms.

The motivation for this is that if continuous-valued probabilistic structured associations can be learned, then creative improvisation algorithms can be developed along analogous

designs to those previously developed for discrete events. Our new approach bridges the gap between (a) computational models that leverage linguistic approaches to describing the complex structural relationships between different musical parts or languages, and (b) computational models that realistically describe truly continuous valued musical events such as pitch or volume. This gap is presently one of the impediments in modeling many creative decisions necessary in live musical improvisation and accompaniment.

Continuous stochastic transduction grammars represent perhaps the first completely integrated models that are capable of finely representing musical events as continuous values while modeling probabilistic structural compositional relations between multiple musical languages. Crude quantization into discrete symbols is no longer necessarily needed in STG modeling.

We illustrate (a) the new representational approach, and (b) the new EM training algorithm for continuous STGs. To illustrate how the new formulation works, we consider an example inspired by that fact that the same traditional and folk genres in which improvisation plays an important role also very often make heavy use of microtonal pitches, as opposed to ‘clean’ notes on a discrete scale. The degree to which notes are ‘bent’ may depend on a host of contextual factors both within and between parts, at various different granularities of musical structure. To approach human levels of improvisation quality, or to advance musicological studies, truly integrated computational modeling must natively handle not only discretized symbols but also continuous values.

We show how the relationship in blues music between microtonal melody pitches (‘bent notes’) and bass pitches can be modeled, by instantiating the idea of continuous STGs for a particularly useful kind of STGs known as a **stochastic inversion transduction grammar** or **SITG**. This is motivated by the fact that SITGs have been empirically shown over decades to exhibit an excellent balance of expressiveness and inductive biases, while maintaining practical polynomial computational complexity characteristics (including statistical machine translation, as well as the hip hop and flamenco models mentioned above). This enables an efficient polynomial time algorithm for expectation-maximization training of continuous SITGs.

Stochastic transduction grammars

Transduction grammars can be seen in the generative modeling paradigm of GTTM (Lerdahl and Jackendoff, 1983) and Steedman Steedman (1984) or Steedman (1996) in using formal grammars to model musical sequences—but instead of monolingual modeling of a single musical language, transduction grammars represent bilingual modeling of the *relationship* between two musical languages.

This makes sense because music is not primarily about a single sequence. Rather, what makes music *musical* more often than not concerns the loosely coupled relationships between parallel strands of different kinds of sequences. Transduction grammars are by nature *bilingual*, which renders them ideally suited for modeling the complex structural relationships between different musical sequences.

Just like natural language, music is highly nondeterministic. As with language, stochastic versions of transduction grammars must be used for any but the most trivial models of music. Much of the previous work on stochastic grammatical modeling of music has been based on flat Markov models and/or hidden Markov models (HMMs). The Continuator model of Pachet (2003) and the Factor Oracle models of Assayag *et al.* (2006) and Assayag and Dubnov (2004) both learned music improvisation conventions using Markov models, later further explored by François *et al.* (2007) and François *et al.* (2010). Jazz grammars were induced by Gillick *et al.* (2010) also under Markovian assumptions.

Much less has been done on modeling of musical structure via stochastic context-free grammars Lari and Young (1990). Unsupervised learning of CCMs (a variant of SCFGs) for musical grammars was described by Swanson *et al.* (2007) and in the DOP approach originally proposed by Bod (2001).

The work on machine learning of stochastic transduction grammars originated largely in the statistical natural language processing community. Stochastic transduction grammars generalize stochastic grammars to model two streams instead of one. As transduction grammars are strictly more powerful than their corresponding monolingual grammars, they are capable of modeling anything that stochastic grammars can model. **Inversion transduction grammars** or ITGs (Wu, 1997) are a subclass of syntax directed transduction grammars or SDTGs (Lewis and Stearns, 1968) that generalize context-free grammars to the bilingual case. Stochastic ITGs, or SITGs, are the bilingual generalization of stochastic CFGs and have proven extremely effective in machine translation as well as other NLP applications.

Whereas the production rules in CFGs probabilistically generate a monolingual subtree, the transduction rules in STGs probabilistically generate both input and output language subtrees. Just as in CFGs, subtrees are generated by recursively combining smaller subtrees (which describe the compositional structure of aligned input and output chunks) into larger subtrees. But unlike in monolingual CFGs, each leaf of a parse tree is a preterminal representing a bilingual *pair* of atoms, as opposed to simply a monolingual atom.

ITGs restrict the alignment between the children of any internal node to be only straight or inverted, rather than arbitrary permutations. This ITG restriction empirically (and somewhat surprisingly) provides sufficient alignment flexibility between the input and output language atoms across virtually every pair of natural languages (Zens and Ney, 2003; Saers *et al.*, 2009; Addanki *et al.*, 2012), but unlike general SDTGs, yields tractable polynomial time training and translation algorithms.

Stochastic transduction grammars appear quite promising for learning of probabilistic structural relations between musical languages. Wu (2013) learned a SITG that discovered structural relationships between flamenco *cajón* and *palmas* languages via transduction grammar induction driven by a Bayesian MAP (maximum *a posteriori*) criterion, in which metrical relations, hypermetrical relations, and probabilistic transduction relations were simultaneously integrated. Wu *et al.* (2013) used SITG induction to automatically learn hip hop freestyling by discovering structural relationships

between challenge and response rap languages. However, these models suffer from the weakness mentioned above of only being able to model non-continuous musical information that can be represented in terms of discrete symbols.

Continuous STGs

We now describe how continuous stochastic transduction grammars represent continuous-valued musical information at various levels of structural granularity within an integrated model, by generalizing a step at a time from context-free grammars. For greater detail on the formal properties of STGs, the reader is referred to (Wu, 1997) and (Wu, 2010).

In the well-known **twelve-bar blues** form, verses consist of three lines: a first four bars, a second four, and a third four called a turnaround. The following syntactic rules, in a conventional context-free grammar, describe a twelve-bar blues in its typical ‘quick to four’ variant:

S	\rightarrow	VERSE
S	\rightarrow	[VERSE S]
VERSE	\rightarrow	[FIRST8 TURNAROUND]
FIRST8	\rightarrow	[FIRST4 SECOND4]
FIRST4	\rightarrow	[AD AA]
SECOND4	\rightarrow	[DD AA]
TURNAROUND	\rightarrow	[ED AA]
AA	\rightarrow	[$A A$]
AD	\rightarrow	[$A D$]
DD	\rightarrow	[$D D$]
ED	\rightarrow	[$E D$]

We can generalize this to a bilingual transduction grammar that expresses the relationship between, for example, a bassline language and a vocal melody language. Ordinary grammars have preterminal symbols corresponding to the monolingual lexical atoms of a single language. On the other hand, transduction grammars have bilingual preterminal symbols corresponding to a relation between two lexical atoms from two *different* languages, which is called a **biterminal**. Let us further decompose the nonterminal symbol A , which represents a single bar in the tonic, into a finer grained series of frames—we’ll use eighth note durations for simplicity’s sake here, though we could also use much finer granularities:

A	\rightarrow	[AT BU CV DW EX FY GZ H0]
AT	\rightarrow	a/t
BU	\rightarrow	b/u
CV	\rightarrow	c/v
DW	\rightarrow	d/w
EX	\rightarrow	e/x
FY	\rightarrow	f/y
GZ	\rightarrow	g/z
H0	\rightarrow	h/ϵ

The preterminal AT, for instance, generates the biterminal a/t which stands for a bassline language atom a , representing some bass note, that is associated with a melody

language atom t , representing some melodic note. The special empty symbol ϵ , represents an absence or silence—for example, the preterminal H0 generates the **singleton** biterminal h/ϵ which represents a standalone bassline note h against which no melodic note occurs. Thus, the nonterminal A simultaneously generates *both* the bassline $abcdefgh$, and the melody $tuvwxyz\epsilon$. We use the convention of referring to the languages to the left and right of the slash as **language 0** and **language 1**, respectively.

Positional variation in musical phrases

Blues are a good example of an improvisational form in which often melodic phrases are re-used or swapped into different positions within the verses. Melodies from the first four are often re-used or swapped into the second four instead, and vice versa.

We can easily model such phenomena using inversion transduction grammars, since ITGs naturally model the possibility of such swapping of positions of various chunks (a constant phenomenon in natural language translation). Consider the ordinary **straight** rule for FIRST8 from above. If we also add a corresponding **inverted** rule, then we now have two alternatives, where the angle brackets signify that the order for language 1 is inverted:

$$\begin{aligned} \text{FIRST8} &\rightarrow [\text{FIRST4 SECOND4}] \\ \text{FIRST8} &\rightarrow \langle \text{FIRST4 SECOND4} \rangle \end{aligned}$$

This says that for the same language 0 bassline generated by the sequence of two constituents FIRST4 and SECOND4, the language 1 melodic phrase that was played against the bassline of the FIRST4 could also be played against the language 0 bassline of the SECOND4, and vice versa.

As a result, now the melody $tuvwxyz\epsilon$ (generated in language 1 by the nonterminal A , which leads off FIRST8) can not only be played against the bassline $abcdefgh$ (generated in language 0 again by the nonterminal A), but can also possibly be played against whatever bassline is generated in language 0 by the nonterminal D , which leads off SECOND8.

Probabilistic biases and preferences

Just as with monolingual stochastic CFGs, a stochastic transduction grammar is parameterized by associating a probability with each transduction rule. This imposes a probability distribution over the space of possible distributions.

Denoting the model being learned as Φ , the lexical rule $\text{AT} \rightarrow a/t$ for example has the probability $b_{\text{AT}}(a/t) \equiv P(\text{AT} \rightarrow a/t \mid \Phi)$. Likewise, the syntactic rule $\text{FIRST4} \rightarrow [\text{AD AA}]$ has the probability $a_{\text{FIRST4} \rightarrow [\text{AD AA}]} \equiv P(\text{FIRST4} \rightarrow [\text{AD AA}] \mid \Phi)$, and this could be used to bias the nondeterministic choice between the ‘quick to four’ and basic variants of twelve-bar blues:

$$\begin{aligned} \text{FIRST4} &\rightarrow [\text{AD AA}] \\ \text{FIRST4} &\rightarrow [\text{AA AA}] \end{aligned}$$

Continuous values

In conventional STG models, it is necessary to assign melodic symbols like a and x to ‘clean’ notes like F# and

C# in Western classical scales. This of course does not come close to adequately describing the microtonal pitch values of the characteristic ‘bent notes’ that are pervasive in blues. Pitches can be bent a little, or a lot, creating significantly different musical effects. Many other non-Western genres, such as flamenco or Indian genres, are even more sensitive to the microtones. A native approach to modeling such continuous values is needed if integrated STG modeling is to be realistically applied to music in general.

In continuous STGs, we replace biterminals that consisted of a pair of discrete symbols, like a/t , with biterminals that instead consist of a pair of continuous values. This means the probability of lexical rules in which preterminals generate biterminals, for example $b_{\text{AT}}(a/t) \equiv P(\text{AT} \rightarrow a/t \mid \Phi)$ which formerly had a scalar value, must be replaced by probability density functions. Using independent Gaussians, with x and y as real values:

$$b_{\text{AT}}(x/y) \equiv \frac{1}{\sqrt{2\pi\sigma_{\text{AT},0}^2}} e^{-\frac{(x-\mu_{\text{AT},0})^2}{2\sigma_{\text{AT},0}^2}} + \frac{1}{\sqrt{2\pi\sigma_{\text{AT},1}^2}} e^{-\frac{(y-\mu_{\text{AT},1})^2}{2\sigma_{\text{AT},1}^2}}$$

With this generalization, x can be used to represent a microtonal melodic pitch, while y can be used to represent an exact bass pitch.

EM training of continuous STGs

Applications

There are numerous applications for automatic simultaneous estimation of both the probabilities for syntactic transduction rules and the pdfs for lexical transduction rules.

In cases where full or partial knowledge of the high-level structure of musical forms is available, as with twelve-bar blues, we can estimate probabilities for the syntactic transduction rules from data. Note that it is not necessary for the training set to be parsed or annotated.

In cases where no high-level structure is known in advance, as in Wu (2013), estimation of transduction rule probabilities is a basic building block in transduction grammar induction algorithms that automatically analyze and extract the high-level structure.

In either case, simultaneously estimating the pdfs for lexical transduction rules is both important for (a) anchoring estimation of the syntactic transduction rule probabilities from continuous data, and (b) automatically improving the modeling of phenomena like microtonal pitches and volumes.

Algorithm

Estimation of probabilities for both syntactic and lexical transduction rules in continuous SITGs like those in the previous section can be accomplished in $O(n^6)$ time via an expectation-maximization algorithm for iteratively improving the transduction rule parameters, driven by a maximum likelihood objective. As all ITGs can be normalized into an equivalent 2-normal form (Wu, 1997), we can simplify the description of the algorithm by assuming the SITG to be in 2-normal form, although EM can also readily be implemented for SITGs in arbitrary form. Unlike the inside-outside algorithm for estimating parameters of monolingual SCFGs



Figure 1: Example contour for a blues vocal melodic phrase that occurs repeatedly in verses at alternate positional variants, showing heavy use of microtonal ‘bent’ notes.

(Baker, 1979; Lari and Young, 1990), this algorithm handles bilingual SITGs allowing positional variance and pdfs over pairs of continuous-valued musical properties on two musical language streams.

Each iteration of EM first computes generalized inside and outside probabilities, as shown in Figure 2. These quantities are used in reestimating the model parameters Φ employing the procedure derived in Figure 3. We use the shorthand $e_{s..t}$ to denote the language 0 subsequence of continuous values in the span from s to t , or more precisely $e_s, e_{s+1}, \dots, e_{t-1}$. Likewise, $f_{u..v}$ denotes a subsequence in language 1. We use the notation q_{stuv} to denote the nonterminal label on a bilingual span or bispan $s..t, u..v$.

How blue can you get?

Microtonal blues notes can be ‘bent’ to a larger or smaller degree; the musical effect is altered by the degree to which they are ‘bent’. An accurate model of blues should be capable of learning what degree of microtonal ‘bending’ goes well with what other parts and in what contexts, so as to reflect biases and preferences in accompaniment and improvisation.

To test this, we trained a continuous SITG using data extracted from the twelve-bar blues ‘Give Me One Reason’, as recorded by Tracy Chapman. This ‘quick to four’ blues consisted of seven vocal verses (plus one instrumental verse), over the course of which all the phenomena described in the foregoing sections are exhibited.

The vocal melody and bassline were extracted using the Tony system (Mauch *et al.*, 2015), and then converted into a sequence of frames in language 0 and language 1 streams. Figure 1 shows the melody’s heavy use of bent notes.

The transduction grammar encapsulated prior knowledge of the basic twelve-bar blues structures, including the syntactic rules discussed earlier. For the preterminal rules’ Gaussian pdfs, on the other hand, the means were randomly ini-

tialized rather than trying to predefine microtonal values by hand, and the variances simply initialized to constants. For each nonterminal that directly dominated preterminals, two alternate versions were ‘cloned’, with separate randomly initialized preterminals allocated to each frame. This strategy provided exploration space to the continuous SITG to self-learn microtonal melodies, basslines, and their interrelationships.

EM training discovered the two main melodic phrases—assigning them to different nonterminals by allocating the ‘clones’.

Because the SITG permits positional variation, EM training pays attention to similar melodic phrases, whether they occurred in the first four or the second four bars. For the numerous occurrences of melodic phrases similar to Figure 1, we left it to EM training to determine whether a better fitting model could be learned by (a) grouping them all into the same melodic nonterminal category, thereby generalizing over the positional variation, versus (b) associating them with separate nonterminal categories for the first four versus the second four, due to systematic biases. Both possibilities are considered by EM, and they both influence generalization since the probabilities under both alternatives are aggregated when computing the expectations.

In this case EM decided in favor of the latter, despite the fact that the melodic phrases appear essentially the same when aggressively quantized into ‘clean’ notes on the scale. By instead modeling the continuous microtonal pitches, a correlation that previously would have been overlooked emerges, between the degree of melodic bending and the bassline pitch. For the ‘same’ melodic phrase, greater bending is associated with the tonic that introduces the first four, compared with the subdominant that introduces the second four. (The preference might be ascribed to greater dissonance in the latter case.)

It could well be that the preferences learned here were idiosyncratic to a particular performer. The EM technique can be used to adapt to mimic styles of particular individuals (as in this case), or alternatively it can be trained on data aggregated from many performers, in order to gain insight on general tendencies in a genre.

After the model parameters have been trained, it becomes possible to use the trained SITG for accompaniment or improvisation. This is accomplished via a transduction algorithm similar to that used in tree-based machine translation Wu and Wong (1998), but again generalized to handle continuous values instead of discrete symbols in analogous fashion to the EM algorithm. Either we can designate the melody (language 0) as the ‘output’ part to be improvised against a human ‘input’ bassline (language 1), or we can designate the bassline (language 1) as the ‘output’ part to accompany a human ‘input’ melody (language 0). In order to find the most likely improvisation or accompaniment (which we can think of as finding the best translation of the ‘input’), we use dynamic programming based parsing to apply the ‘input’ half of the trained SITG rules to the ‘input’ language. Once the most likely parse is found, reading the ‘output’ half of the rules forming that parse yields the best translation.

Conclusion

We have discussed a new strategy for learning complex structural relationships between microtones, and other continuous valued musical features, that simultaneously models contextual influences both within and between different musical languages (players or parts) at many hierarchical or compositional levels of granularity, in improvisational and accompaniment settings. Using continuous stochastic transduction grammars, we bridge the computational modeling gap between (a) fully integrating structural, hierarchical inter-part factors, and (b) finely represented continuous valued signals, overcoming what has until now been one of the major weaknesses in realistically modeling of music based on STGs. Because continuous STGs natively handle continuous valued biterminals, phenomena like microtonal pitch can be modeled without crude quantization to ‘clean’ notes.

The degree to which melody notes in blues should be ‘bent’ in the context of decisions made by other players, such as that of the bassline, can be learned via a practical polynomial-time EM algorithm for the continuous instantiation of stochastic inversion transduction grammars—empirically one of the most useful subclasses of stochastic transduction grammars. Syntactic and preterminal probabilities are automatically learned, to model patterns at different contextual granularities between two different continuous valued parts while allowing positional variance.

We are currently exploring whether neural networks, which employ inherently continuous valued representations, could be used to augment continuous STGs. The recursive neural network implementation of STGs described by Wu and Addanki (2015) still only use continuous valued vectors to represent discrete symbols. We believe such neural networks may be directly useful for true continuous valued musical signals, but perhaps in combination with the approach discussed in this paper because the neural models are significantly more lossy and noisy, and difficult to analyze in terms of what musical knowledge they encode.

Acknowledgements

This work is supported in part by the Hong Kong Research Grants Council (RGC) research grants GRF16210714, GRF16214315, GRF620811 and GRF621008; by the Defense Advanced Research Projects Agency (DARPA) under LORELEI contract HR0011-15-C-0114, BOLT contracts HR0011-12-C-0014 and HR0011-12-C-0016, and GALE contracts HR0011-06-C-0022 and HR0011-06-C-0023; and by the European Union under the Horizon 2020 grant agreement 645452 (QT21) and FP7 grant agreement 287658. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

References

Karteek Addanki, Chi-Kiu Lo, Markus Saers, and Dekai Wu. LTG vs. ITG coverage of cross lingual verb frame alternations. In *16th Annual Conference of the European Association for Machine Translation (EAMT-2012)*, pages 295–302, Trento, Italy, May 2012.

- Gérard Assayag and Shlomo Dubnov. Using factor oracles for machine improvisation. *Soft Computing*, 8:1432–7643, 2004.
- Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. OMax Brothers: A dynamic topology of agents for improvisation learning. In *First ACM Workshop on Audio and Music Computing Multimedia*, pages 125–132, 2006.
- James K. Baker. Trainable grammars for speech recognition. In D. H. Klatt and J. J. Wolf, editor, *Speech Communication Papers for the 97th Meeting of the Acoustic Society of America*, pages 547–550, 1979.
- Rens Bod. Stochastic models of melodic analysis: Challenging the gestalt principles. *Journal of New Music Research*, 30(3), 2001.
- Alexandre R.J. François, Elaine Chew, and Dennis Thurmond. Mimi - a musical improvisation system that provides visual feedback to the performer. Technical Report 07-889, USC Computer Science Department, Apr 2007.
- Alexandre R.J. François, Isaac Schankler, and Elaine Chew. Mimi4x: An interactive audio-visual installation for high-level structural improvisation. In *IEEE International Conference on Multimedia and Expo (ICME 2010)*, pages 1618–1623, 2010.
- Jon Gillick, Kevin Tang, and Robert M. Keller. Machine learning of jazz grammars. *Computer Music Journal*, 34(3):56–66, Fall 2010.
- Karim Lari and Steve J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.
- Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.
- Philip M. Lewis and Richard E. Stearns. Syntax-directed transduction. *Journal of the Association for Computing Machinery*, 15(3):465–488, 1968.
- Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justin Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. In *First International Conference on Technologies for Music Notation and Representation (TENOR 2015)*, 2015.
- François Pachet. The Continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):33–341, 2003.
- Markus Saers, Joakim Nivre, and Dekai Wu. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithms. In *11th International Conference on Parsing Technologies (IWPT’09)*, pages 29–32, Paris, Oct 2009.
- Mark J. Steedman. The formal description of musical perception. *Music Perception*, 2:52–77, 1984.
- Mark J. Steedman. The blues and the abstract truth: Music and mental models. In A. Garnham and J. Oakhill, editors, *Mental Models in Cognitive Science*, pages 305–318. Erlbaum, 1996.

1. Recursive computation of generalized inside probabilities $\beta_{stuv}(i) \equiv P[i \xrightarrow{*} e_{s..t}/f_{u..v} | q_{stuv} = i, \Phi]$

(a) Basis

$$\begin{aligned} \beta_{ttvv}(i) &= 0 & 0 \leq t \leq T, 0 \leq v \leq V \\ \beta_{stuv}^0(i) &= \begin{cases} b_i(e_s/f_u) & \text{if } s+1=t, u+1=v, 0 \leq s < t \leq T, 0 \leq u < v \leq V \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

(b) Recursion

$$\begin{aligned} \beta_{stuv}(i) &= \beta_{stuv}^{[1]}(i) + \beta_{stuv}^{(\cdot)}(i) + \beta_{stuv}^0(i) \\ \beta_{stuv}^{[1]}(i) &= \sum_{\substack{1 \leq j \leq N \\ 1 \leq k \leq N \\ s \leq S \leq t \\ u \leq U \leq v \\ (S-s)(t-S)+(U-u)(v-U) \neq 0}} a_{i \rightarrow [jk]} \beta_{sSuU}(j) \beta_{StUv}(k) \\ \beta_{stuv}^{(\cdot)}(i) &= \sum_{\substack{1 \leq j \leq N \\ 1 \leq k \leq N \\ s \leq S \leq t \\ u \leq U \leq v \\ (S-s)(t-S)+(U-u)(v-U) \neq 0}} a_{i \rightarrow \langle jk \rangle} \beta_{sSuU}(j) \beta_{StUv}(k) \end{aligned}$$

2. Recursive computation of generalized outside probabilities $\alpha_{stuv}(i) \equiv P[S \xrightarrow{*} e_{0..s} i e_{t..T} / f_{0..u} i f_{v..V}, q_{stuv} = i | \Phi]$

(a) Basis

$$\begin{aligned} \alpha_{0,T,0,V}(i) &= \begin{cases} 1 & \text{if } i = S \\ 0 & \text{otherwise} \end{cases} \\ \alpha_{ttvv}(i) &= 0 & 0 \leq t \leq T, 0 \leq v \leq V \end{aligned}$$

(b) Recursion

$$\begin{aligned} \alpha_{stuv}(i) &= \alpha_{stuv}^{[1]}(i) + \alpha_{stuv}^{(\cdot)}(i) \\ \alpha_{stuv}^{[1]}(i) &= \sum_{\substack{1 \leq j \leq N \\ 1 \leq k \leq N \\ 0 \leq S \leq s \\ 0 \leq U \leq u \\ (s-S)(u-U) \neq 0}} \alpha_{StUv}(j) a_{j \rightarrow [ki]} \beta_{sSuU}(k) + \sum_{\substack{1 \leq j \leq N \\ 1 \leq k \leq N \\ t \leq S \leq T \\ v \leq U \leq V \\ (S-t)(U-v) \neq 0}} \alpha_{sSuU}(j) a_{j \rightarrow [ik]} \beta_{tSvU}(k) \\ \alpha_{stuv}^{(\cdot)}(i) &= \sum_{\substack{1 \leq j \leq N \\ 1 \leq k \leq N \\ 0 \leq S \leq s \\ v \leq U \leq v \\ (s-S)(U-v) \neq 0}} \alpha_{StUv}(j) a_{j \rightarrow \langle ki \rangle} \beta_{sSuU}(k) + \sum_{\substack{1 \leq j \leq N \\ 1 \leq k \leq N \\ t \leq S \leq T \\ 0 \leq U \leq u \\ (S-t)(u-U) \neq 0}} \alpha_{sSuU}(j) a_{j \rightarrow \langle ik \rangle} \beta_{tSvU}(k) \end{aligned}$$

Figure 2: Dynamic programming for computing generalized inside and outside probabilities for continuous SITGs.

Reid Swanson, Elaine Chew, and Andrew S. Gordon. Supporting musical creativity with unsupervised syntactic parsing. In *AAAI Spring Symposium on Creative Intelligent Systems*, 2007.

Dekai Wu and Karteek Addanki. Learning to rap battle with bilingual recursive neural networks. In *24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 2524–2530, Buenos Aires, Jul 2015.

Dekai Wu and Hongsing Wong. Machine translation with a stochastic grammatical channel. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL '98)*, Montreal, Aug 1998.

Dekai Wu, Karteek Addanki, Markus Saers, and Meriem Be-

loucif. Learning to freestyle: Hip hop challenge-response induction via transduction rule segmentation. In *2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 102–112, Seattle, Oct 2013.

Dekai Wu. Stochastic Inversion Transduction Grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, Sep 1997.

Dekai Wu. Alignment. In Nitin Indurkha and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, pages 367–408. Chapman and Hall / CRC, second edition, 2010.

Dekai Wu. Simultaneous unsupervised learning of flamenco metrical structure, hypermetrical structure, and multipart

1. Probability of using each nonterminal in a derivation of the observed training pair:

$$P[i \text{ used} \mid S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi] = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid q_{stuv} = i, \Phi]}{P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid \Phi]} = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}{P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid \Phi]}$$

2. Probability of using each straight or inverted transduction rule in a derivation of the observed training pair:

$$P[i \rightarrow [jk] \text{ used} \mid S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi] = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V P[i \Rightarrow [jk] \xrightarrow{*} e_{s..t}/f_{u..v} \mid S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi]}{P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid \Phi]} = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \sum_{S=s}^t \sum_{U=u}^v a_{i \rightarrow [jk]} \alpha_{stuv}(i) \beta_{sSuU}(j) \beta_{StUv}(k)}{P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid \Phi]}$$

$$P[i \rightarrow \langle jk \rangle \text{ used} \mid S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi] = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V P[i \Rightarrow \langle jk \rangle \xrightarrow{*} e_{s..t}/f_{u..v} \mid S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi]}{P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid \Phi]} = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \sum_{S=s}^t \sum_{U=u}^v a_{i \rightarrow \langle jk \rangle} \alpha_{stuv}(i) \beta_{sSuU}(j) \beta_{StUv}(k)}{P[S \xrightarrow{*} \mathbf{e}/\mathbf{f} \mid \Phi]}$$

3. Transduction rule probabilities (by definition):

$$a_{i \rightarrow [jk]} \equiv P[i \rightarrow [jk] \text{ used} \mid i \text{ used}, S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi]$$

$$a_{i \rightarrow \langle jk \rangle} \equiv P[i \rightarrow \langle jk \rangle \text{ used} \mid i \text{ used}, S \xrightarrow{*} \mathbf{e}/\mathbf{f}, \Phi]$$

4. Re-estimation procedure for transduction rule probabilities \hat{a} (by substitution):

$$\hat{a}_{i \rightarrow [jk]} = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \sum_{S=s}^t \sum_{U=u}^v a_{i \rightarrow [jk]} \alpha_{stuv}(i) \beta_{sSuU}(j) \beta_{StUv}(k)}{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}$$

$$\hat{a}_{i \rightarrow \langle jk \rangle} = \frac{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \sum_{S=s}^t \sum_{U=u}^v a_{i \rightarrow \langle jk \rangle} \alpha_{stuv}(i) \beta_{sSuU}(j) \beta_{StUv}(k)}{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}$$

5. Re-estimation procedure for preterminal rules' Gaussian means $\hat{\mu}$ and variances $\hat{\sigma}$:

$$\hat{\mu}_{i,0} = \frac{\sum_{s=0}^{T-1} \sum_{u=0}^{V-1} e_s \alpha_{s,s+1,u,u+1}(i) \beta_{s,s+1,u,u+1}(i)}{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}$$

$$\hat{\sigma}_{i,0}^2 = \frac{\sum_{s=0}^{T-1} \sum_{u=0}^{V-1} (e_s - \mu_{i,0})^2 \alpha_{s,s+1,u,u+1}(i) \beta_{s,s+1,u,u+1}(i)}{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}$$

$$\hat{\mu}_{i,1} = \frac{\sum_{s=0}^{T-1} \sum_{u=0}^{V-1} f_u \alpha_{s,s+1,u,u+1}(i) \beta_{s,s+1,u,u+1}(i)}{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}$$

$$\hat{\sigma}_{i,1}^2 = \frac{\sum_{s=0}^{T-1} \sum_{u=0}^{V-1} (f_u - \mu_{i,1})^2 \alpha_{s,s+1,u,u+1}(i) \beta_{s,s+1,u,u+1}(i)}{\sum_{s=0}^T \sum_{t=s}^T \sum_{u=0}^V \sum_{v=u}^V \alpha_{stuv}(i) \beta_{stuv}(i)}$$

Figure 3: Derivation of EM reestimation of model parameters Φ for continuous SITGs, using inside and outside probabilities.

structural relations. In *14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, Curitiba, Brazil, Nov 2013.

In *41st Annual Meeting of the Association for Computational Linguistics (ACL-2003)*, pages 192–202, Sapporo, Aug 2003.

Richard Zens and Hermann Ney. A comparative study on reordering constraints in statistical machine translation.

A Music-generating System Based on Network Theory

Shawn Bell

Dawson College
Arts, Literature & Communication Program
Interactive Media Arts profile
Montreal QC, H3Z 1A4, Canada
sbell@dawsoncollege.qc.ca

Liane Gabora

Psychology Department
University of British Columbia (Okanagan Campus)
Kelowna BC, V1V 1V7, Canada
liane.gabora@ubc.ca

Abstract

This paper is the first scholarly presentation of NetWorks (NW), an interactive music-generation system that uses a hierarchically clustered scale-free network to generate music that ranges from orderly to chaotic. NW was inspired by the Honing Theory of creativity, according to which human-like creativity hinges on (1) ability to self-organize and maintain dynamics at the ‘edge of chaos’ using something akin to ‘psychological entropy’, and (2) the capacity to shift between analytic and associative processing modes. At the ‘edge of chaos’ it generates patterns that exhibit emergent complexity through coherent development at low, mid, and high levels of musical organization, and often suggests goal seeking behavior. The architecture consists of four 16-node modules: one each for pitch, velocity, duration, and entry delay. The *Core* allows users to define how nodes are connected, and rules that determine when and how nodes respond to their inputs. The *Mapping Layer* allows users to map node output values to MIDI data that is routed to software instruments in a digital audio workstation. By shifting between bottom-up and top-down it shifts between analytic and associative processing modes.

Introduction

This paper introduces NetWorks (NW), a music-generating program inspired by the view that (1) the human mind is a complex adaptive system (CAS), and thus (2) human-like computational creativity is best achieved by drawing on the science of complex systems. NW uses scale-free networks and the concept of the ‘edge of chaos’ to generate music that is aesthetically pleasing and that maintains interest. The approach has origins that date back to a CD of emergent, self-organizing computer music based on cellular automata and asynchronous genetic networks titled, “Voices From The Edge of Chaos” (Bell 1998), and more generally to the application of artificial life models to computer-assisted composition, generative music and sound synthesis (Beyls 1989, 1990, 1991; Bowcott 1989; Chareyron 1990; Horner and Goldberg 1991; Horowitz 1994; Millen 1992; Miranda 1995; Todd and Loy 1991).

We first summarize key elements of a CAS-inspired theory of creativity, and discuss the relevance for computational creativity. Next we outline the architecture of NW,

evaluate its outputs, and highlight some of its achievements. We then summarize how the NW architecture adheres to principles of honing theory and CAS, and how this contributes to the appealing musicality of its output.

Honing Theory: Creativity as a Complex Adaptive System

The honing theory (HT) of creativity (Gabora 2010, in press) has its roots in the question of what kind of structure could evolve novel, creative forms effectively and strategically (as opposed to at random). We now summarize the elements of the theory most relevant to NetWorks.

Self-Organization

Humans possess two levels of complex, adaptive, structure: an organismic level and a psychological level, i.e., a mind (Pribram 1994). Like a body, a mind is *self-organizing*: a new stable global organization can emerge through interactions amongst the parts (Ashby 1947; Carver and Scheier 2002; Prigogine and Nicolis 1977). The capacity to self-organize into a new patterned structure of relationships is critical for the generation of creative outcomes (Abraham 1996; Goertzel 1997; Guastello 1998). The mind’s self-organizing capacity originates in a memory that is distributed, content addressable, and sufficiently densely packed that for any one item there is a reasonable probability it is similar enough to some other item to evoke a reminding of it, thereby enabling the redescription and refinement of ideas and actions in a stream of thought (Gabora, 2010). Mental representations are distributed across neural cell assemblies that encode for primitive stimulus features such as particular tones or timbres. Mental representations are both constrained and enabled by the strengths of connections between neurons they activate.

Just as a body mends itself when injured, a mind is on the lookout for ‘gaps’—arenas of incompleteness or inconsistency or pent-up emotion—and explores the gap from different perspectives until a new understanding has been achieved. We can use the term *self-mending* to refer to the capacity to reduce psychological entropy in response to a perturbation (Gabora, in press), i.e., it is a form of self-organization involving reprocessing of arousal-provoking

material. Creative thinking induces *restructuring* of representations, which may involve re-encoding the problem such that new elements are perceived to be relevant, or relaxing goal constraints. However, according to HT, the transformative impact of immersion in the creative process can bring about sweeping changes to that second (psychological) level of complex, adaptive structure that alter one's self-concept and view of the world.

Edge of Chaos

Self-organized criticality (SOC) is a phenomenon wherein, through simple local interactions, complex systems find a critical state poised at the transition between order and chaos—the proverbial *edge of chaos*—from which a small perturbation can exert a disproportionately large effect (Bak, Tang, and Wiesenfeld 1988). It has been suggested that insight is a self-organized critical event (Gabora 1998; Schilling 2005). SOC gives rise to structure that exhibits sparse connectivity, short average path lengths, and strong local clustering. Other indications of SOC include long-range correlations in space and time, and rapid reconfiguration in response to external inputs. There is evidence of SOC in the human brain, e.g., with respect to phase synchronization of large-scale functional networks (Kitzbichler, Smith, Christensen, and Bullmore 2009). There is also evidence of SOC at the cognitive level; word association studies show that concepts are clustered and sparsely connected, with some having many associates and others few (Nelson, McEvoy, and Schreiber 2004). Cognitive networks exhibit the sparse connectivity, short average path lengths, and strong local clustering characteristic of self-organized complexity and in particular 'small world' structure (Steyvers and Tenenbaum 2005).

Like other SOC systems, a creative mind may function within a regime midway between order (systematic progression of thoughts), and chaos (everything reminds one of everything else). Much as most perturbations in SOC systems have little effect but the occasional perturbation has a dramatic effect, most thoughts have little effect on one's worldview, but occasionally one thought triggers another, which triggers another, and so forth in a chain reaction of conceptual change. This is consistent with findings that large-scale creative conceptual change often follows a series of small conceptual changes (Ward, Smith, and Vaid 1997), and with evidence that power laws and catastrophe models are applicable to the diffusion of innovations (Jacobsen and Guastello 2011).

Contextual Focus

Psychological theories of creativity typically involve a divergent stage that predominates during idea generation and a convergent stage that predominates during the refinement, implementation, and testing of an idea (for a review see Runco 2010; for comparison between divergent / convergent creative processes and dual process models of cognition see Sowden, Pringle, and Gabora 2015). *Diver-*

gent thought is characterized as intuitive and reflective; it involves the generation of multiple discrete, often unconventional possibilities. It is contrasted with *convergent thought*, which is critical and evaluative; it involves tweaking of the most promising possibilities. There is empirical evidence for oscillations in convergent and divergent thinking, with a relationship between divergent thinking and chaos (Guastello 1998). It is widely believed that divergent thought involves defocused attention and associative processing, and this is consistent with the literal meaning of divergent as "spreading out" (as in a divergence of a beam of light). However, the term divergent thinking has come to refer to the kind of thought that occurs during creative tasks that involve the generation of multiple solutions, which may or may not involve defocused attention and associative memory. Moreover, in divergent thought, the associative horizons simply widen generically instead of in a way that is tailored to the situation or context (Fig. 2). Therefore, we will use the term *associative thought* to refer to creative thinking that involves defocused attention and context-sensitive associative processes, and *analytic thought* to refer to creative thinking that involves focused attention and executive processes. The capacity to shift between these modes of thought has been referred to as *contextual focus* (CF) (Gabora 2010). While some dual processing theories (e.g., Evans 2003) make the split between automatic and deliberate processes, CF makes the split between an associative mode conducive to detecting relationships of correlation and an analytic mode conducive to detecting relationships of causation. Defocusing attention facilitates associative thought by diffusely activating a broad region of memory, enabling obscure (though potentially relevant) aspects of a situation to come to mind. Focusing attention facilitates analytic thought by constraining activation such that items are considered in a compact form amenable to complex mental operations.

According to HT, because of the architecture of associative memory, creativity involves not searching and selecting amongst well-formed idea candidates, but amalgamating and honing initially ill-formed possibilities from multiple sources. As a creative idea is honed, its representation changes through interaction with an internally or externally generated contexts, until psychological entropy is acceptably low. The unborn idea is said to be in a 'state of potentiality' because it could actualize different ways depending on the contextual cues taken into account as it takes shape.

The NetWorks System

NW consists of a music-generating system and the music it has produced. The goal of NW is to generate "emergent music," i.e., self-organizing, emergent dynamics from simple rules of interaction, expressed in musical forms. In terms of creative agency, NW has been designed as a closed, autonomous system while generating MIDI data. In selecting the network architecture and interaction rules, the

artist-user may be viewed as the system's mentor. The MIDI data generated by the system is orchestrated and mixed by the artist-user, who may be viewed in this role as a collaborator (McCormack and d'Inverno 2014).

Network theory, as it pertains to the study of complex adaptive systems (Mitchell 2006) was used in the design of the NW system. NW is currently configured to explore the expressive potential of hierarchical scale-free networks, as the properties of such networks underlies the interesting dynamics of many real world networks, from the cell to the World Wide Web (Barabási 2002). Moreover, a variety of musical genres exhibit a scale-free structure (Liu, Tse & Small 2009). Assuming that constraints define genre, the generation of "emergent music" is primarily a search for new genres. Given the ubiquity of hierarchical scale-free topology and dynamics found in CAS it is not surprising that such architecture have creative potential. In NW, the components mutually constrain and enable one another: *nodes* represent the components of a system, and *links* represent the couplings between them. Connected nodes interact through an exchange of values, which change the states of the nodes as well as the state of a network as a whole.

Since in complex systems science the term "hierarchical" often indicates top-down control, the use of hierarchical networks might appear to be at odds with the goal of generating complex emergent behaviour. However, in this model, communication and control flow both top-down and bottom-up between connected nodes. The architecture consists of *clusters* of interconnected nodes, connected by *hubs*, such that the nodes within a hub are more interconnected than nodes between hubs. A hub contributes input to—and thus co-determines—the next state (output value) of the nodes to which it is connected. Likewise, these connected nodes contribute input to—and thus co-determine—the next state of the hub.

Since NW MIDI data is computer-generated, sampled acoustic instruments are often used to give the music a "human feel", and to help the listener to relate and compare the self-organizing output patterns of NW to known genres. In general, the sounds chosen to manifest the musical patterns discovered by the network attempt to reflect the mystery and wonder that virtually unlimited diversity can come from such simple interactive models. When mapping patterns to sound, an effort is made to preserve the integrity of the patterns rather than obfuscate them with complex synthetic textures or other effects (such as echo effect) that are readily available during mixing. NW is composed of two layers:

1. The *Core*, which allows users to define how nodes are connected, as well as the rules that determine when and how nodes respond to their inputs, and
2. The *Mapping Layer*, which allows the user to map node output values to MIDI data that are routed to software instruments in a Digital Audio Workstation (DAW).

We now discuss these two layers in more detail.

The Core

There are several aspects to the core: the relationship between the architecture and the functions, the rules, and the relationship between the architecture and the rules. We now discuss each of these in turn.

Relationship between Architecture and Functions Networks consists of 64 nodes linked together in a scale-free architecture (see diagram below). There are four, 16 node modules: one for pitch, velocity, duration and entry delay (ED). The nodes that comprise the pitch module, are responsible for producing "notes". A note has five basic attributes: pitch, loudness (usually corresponding to MIDI "velocity"), duration, timing (or entry delay), and timbre. Pitch nodes output values for pitch, but require values for velocity and duration to produce a note. These values are provided by the nodes that comprise the velocity and duration modules. The pitch module is unique in that it includes the largest hub, which sends its output to, and receives inputs from, 40 nodes: 12 pitch nodes, 9 nodes each from the duration, velocity, and ED modules, and itself. The pitch node, as well as all other nodes in the network, receive their own outputs to participate in, and trigger the calculation of, its next output.

The ED module is responsible for keeping corresponding nodes of the four modules synchronized (see the diagram below). When a pitch node is activated, as determined by the delay value it receives from its ED module node, the corresponding velocity and duration module nodes are activated simultaneously to provide the values required to specify a "note". The function of the ED module is to determine timing, that is, *when* nodes produce an output, and therefore the pattern of activation across the network as a whole. In musical terms, the entry delay module generates rhythmic patterns, phrases and sections.

Note that nodes receive values from, and send values to, nodes in other modules. In this way, note attributes affect one another. For example, the output values of the nodes that comprise the ED module are determined by the values they receive from pitch, duration, and velocity module nodes. The output values of the nodes that comprise the pitch module are determined by the values they receive from ED, velocity and duration module nodes, and so on.

The timbral characteristics of the notes produced by the network can be partially controlled by mapping various network and module activities to selected synthesis parameters of the software instruments chosen by the user.

When the nodes are fully connected, that is, receiving values on all their inputs, the network architecture is scale-free; however users can prune the connectivity of the network by reducing the number of inputs to the nodes.

Rules When activated, nodes sum the last values received on their inputs and use a lookup table (LUT) to determine the value to output. The number of values (or states) that can be output by the nodes is determined by the user. Ex-

periments have used 13 and 25 values which allows for pitch to be mapped chromatically across one and two octaves respectively, and provide the same number of equivalent scale steps for velocity, duration and entry delay. However, the user can map the output values across any desired range.

NetWorks has been designed to allow:

1. each node to have its own LUT;
2. an LUT for each module;
3. one LUT for all the nodes of the network.

LUTs are generated using a variety of methods: random, random without repetition, ratios, etc.

Relationship between Architecture and Rules Two observations can be made regarding the relationship of rules and network architecture. First, when the network is scale-free, nodes have either 4, 5, 6, 15 or 40 inputs. Each input on a node can receive a range of values, determined by the user, which are then summed to determine an output value. This means the range of output values is always less than the range of summed values. For example, if the number of values that can be output by the nodes is set to 13 (e.g. 0–12), a node with four inputs will require an LUT with an index of 48 to store values for all possible sums, a node with 5 inputs will require a LUT with an index of 60, and so on. The largest hub with 40 inputs requires a LUT with an index of 480, but can only output 13 values (0–12), resulting in a loss of “information”. Put another way, the hub can distinguish between 480 inputs states, but can only respond with 13 different outputs.

Second, while hubs have a wider “sphere of influence” because their output is received by a greater number of nodes, hubs also receive input from the same nodes, which means they have an equal number of inputs that co-determine their outputs. However, the more connected the hub, the more inputs it sums, and the less able it is to respond with unique outputs. While less well-connected nodes have a smaller “sphere of influence”, their ability to distinguish between their inputs with unique outputs is significantly greater.

MIDI Mapping

The number of values (or states) of the nodes is determined by the user. The MIDI Mapping layer allows users to map these output values across appropriate MIDI ranges. For example, If nodes are set to output 12 values:

1. outputs from pitch nodes can be mapped to a chromatic scale (e.g. C4–B4);
2. velocity node outputs can be mapped to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120 MIDI values;
3. duration node outputs can be mapped to an arbitrarily chosen fixed range (e.g., 100, 150, 200, 250, 300 ... 650 milliseconds) or a duration based on a subdivision of the entry delay times between notes.
4. Entry delays values between notes are scaled to an appropriate musical range in milliseconds.

In addition to generating the basic attributes of notes, NetWorks provides for mapping network activity to MIDI cc control data to control various synthesis parameters such as filters, and so forth, chosen the user. However, currently these outputs do not feedback into the network. 64 nodes, organized into four 16 node module clusters allows for 16 channels of MIDI data (Figure 1).

Rules can be constructed to favour certain output values. At the extreme, a rule table could output the same value for any input. Unless a node has only one input (and that would have to be the one from itself, otherwise the node would never “fire”) they can be thought of as “funnels,” always reducing the specifics of their inputs. Nodes do not care which nodes send what values to their inputs; they simply sum the last values received and pass them on after an entry delay time. As feedback happens in time, where nodes may introduce previously stored values into the current stream of activations, the network dynamic as a whole must adjust (or “adapt”) to “old ideas.”

There are many ways inputs can sum to the same value. Nodes with rule tables that favour certain output values are less discriminating (lower resolution). Hubs are always less discriminating since they have more inputs, but the same vocabulary (number of possible output values). Nodes with fewer inputs and an equal distribution of output values across input sums are more discriminating (higher resolution). The interaction between nodes almost always results in an open-ended (endless) stream.

Evaluation of NW Output

To date, two albums have been produced using the NetWorks system: “NetWorks 1: Could-be Music” and “Networks 2: Phase Portraits”, which can be heard online:

- <https://shawnbell.bandcamp.com/album/networks-1-could-be-music>
- <https://shawnbell.bandcamp.com/album/networks-2-phase-portraits>

The most recent experiments can be found here:

- <https://soundcloud.com/zomes>

It is possible to modulate the output dynamics of NW from complete order (and thus repetition without change) to complete chaos (and thus no element of predictability). The musicality of the output is greatest when the system is tuned to an intermediate between these extremes, i.e., the proverbial ‘edge of chaos.’ At this point there is a pleasing balance between familiar, repeating patterns, and the desire for novelty and surprise. The system often generates motifs that repeat, vary, and develop into more complex melodies, as well as return to their original form.

The distribution of node LUTs output values is the determining factor in balancing uniformity and variety. Trivially, if all nodes output the same value, whatever the sum of their inputs, the MIDI output is uniformly repetitive. A random distribution of node LUT output values results in random MIDI output.

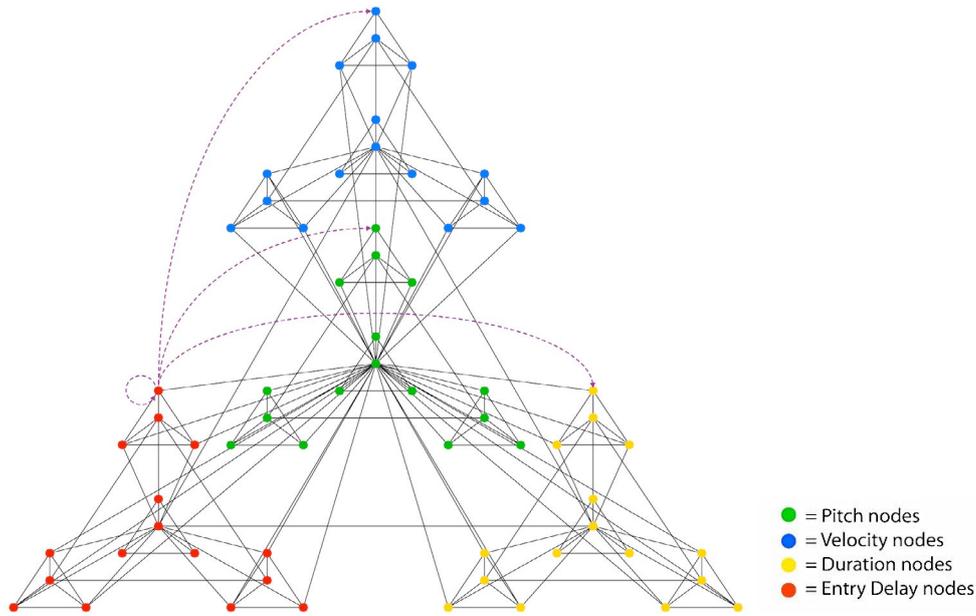


Figure 1. Schematic illustration of the different kinds of nodes and their interrelationships. Undirected edges (in black) indicate that values can be exchanged in both directions, i.e., nodes both send values to, and receive values from, nodes to which they are connected. Directed edges (purple) show the relationship between individual nodes of the Entry Delay module and the corresponding nodes of other modules. The ED module node determines *when* it will activate itself, and the corresponding node in the duration, velocity, and pitch modules. For clarity, only one of the 16 ED nodes and its four corresponding nodes are shown.

Shannon Entropy was used to compare NW MIDI data sequences generated with rules having a random distribution of output values with MIDI data generated using node LUTs that output (mostly) the same value when activated. Entropy was also used to compare NW pieces to other genres of music to confirm subjective comparisons. Entropy is a good measure of the unpredictability / complexity in data sequences. As a simplified data sequence, music has two features: the range of notes, i.e., pitch/duration pairs, and repetitiveness of notes. Entropy values capture the degree of variety and repetitiveness of note sequences in MIDI data. Roughly speaking, high entropy indicates surprising or unpredictable musical patterns while low entropy indicates predictable, repeating musical patterns (Ren 2015).

In this analysis, the entropy of a piece was calculated by counting the frequency of musical events, specifically the appearances of each note (pitch-duration pair), as well as pitch and duration separately to get the discrete distribution of those events. Equation 1 was used to calculate the information content of each note. The expectation value of the information content, defined as $-\log p(x_i)$, was used to obtain the entropy. The entropy is related to the frequency of musical events in a specific range. Differences in entropy values stem from differences of (1) the underlying possibility space size, i.e. how many different types of musical events there are, and (2) how repetitive they are. Although this does not take into account the order of events it provides a general characterization useful for comparing musical sequences (Ren 2015).

$$H(X) = - \sum_i p(x_i) \log p(x_i), i \in n = \text{outcomes} \quad (1)$$

In Figure 2, the entropy value of ten NW pieces ($x\text{-tick}=3$) is compared with Bach's chorales ($x\text{-tick}=1$) and with jazz tunes ($x\text{-tick}=2$). In terms of entropy, NW pieces are closer to jazz than to Bach, which confirms informal subjective evaluations of NW music. $x\text{-tick}=4$ shows the entropy value for three NW pieces generated using a random distribution of LUT output values and $x\text{-tick}=5$ shows the entropy values of three NWs pieces with near uniform LUTs. These values verify the relationship between NW MIDI outputs and the LUTs that generate them.

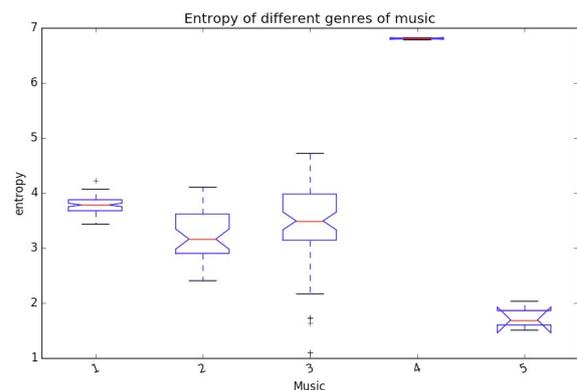


Figure 2. Comparison of entropy of ten NW pieces ($x\text{-tick}=3$) with Bach chorales ($x\text{-tick}=1$) and jazz tunes ($x\text{-tick}=2$).

Evaluation of NW music via social media (SoundCloud), shows an increasing interest in NW music from what is quite likely a diverse audience given the wide range of social-media groups to which NW music has been posted (e.g., classical, jazz, electronic, experimental, ambient, film music, algorithmic music, creative coding, complex systems, etc.). There has been a steady growth of “followers” over the two years (2014-2016) of posting NW pieces (28 tracks). As of the writing of this paper, NW has 307 followers, 7,418 listens, 796 downloads, 330 likes, 24 reposts, and 53 comments (all of which are positive).

As a search for “music-as-it-could-be,” (e.g., new genres) a comment from SoundCloud indicates this goal may have been attained: “What can I say except I think I like it?” This suggests that the person has heard something they cannot categorize, but that sounds like good music.

How NetWorks Implements Honing Theory

We now summarize how the NetWorks (NW) architecture and outputs adhere to and implement ideas from honing theory (HT), a theory of creativity inspired by chaos theory and the theory of complex adaptive systems (CAS).

NW as Creative, Self-Organizing Structure

NW is hardwired to exhibit the key properties of real-world complex systems through its modular, scale-free, small-world properties. NW architecture has a shallow, fractal, self-similar structure (4 node, 16 node, and 64 node modules) which allows multiple basins of attraction to form in parallel, over different timescales, and interact.

NW networks are not neural networks; they do not adapt or learn by tuning weights between nodes through experience or training, nor do they evolve; nodes simply accept input and respond. Their rules of interaction do not change, adapt, or self-organize over time, but their structure does.

Just like an experience or realization can provide the ‘seed incident’ that stimulates creative honing, the pseudo-randomly generated initial conditions provide ‘seed incidents’ that initiate NW processing. After NW receives its inputs it is a closed system that adapts to itself (self-organizes). Musical ideas sometimes unfold in an open-ended manner, producing novelty and surprise, both considered hallmarks of emergence. A diversity of asynchronous interactions (sometimes spread out in time) can push NW dynamics across different basins of attraction. Idea refinement occurs when users (1) generate and evaluate network architectures, rule-sets and mappings, and (2) orchestrate, mix, and master the most aesthetically pleasing instances of these outputs. The role of mental representation is played by notes—their basic attributes as well as attributes formed by their relationships to other notes.

Cellular Automata-like Behavioral Classes NW nodes have a significantly different topology from Cellular Automata (CA). While CA have a regular lattice geometry, NW has a hierarchical (modular), scale-free, small-world

structure. Moreover, unlike CAs, NW is updated asynchronously. However, similar to CA, NW exhibits Wolfram’s class one (homogenous), class two (periodic), class three (chaotic), and class four (complex) behaviour, and—rather than converging to a steady state—tends to oscillate between them. This is because the nested architecture of NW allow multiple basins of attraction to form in parallel and over different timescales. Pruning the scale-free architecture by reducing the inputs to hubs insulates clusters and modules from one another, reducing their interactions. Network dynamics within a basin of attraction can get pushed out of the basin by delayed values entering the system. In other words, because in the context of the current pattern an “old ideas” can push the dynamics to a different basin, the system exhibits “self-mending” behavior. This can result in musical transitions that lead to the emergence of new patterns and textures.

Representational Redescription The network “makes sense” of its present in terms of its past by adapting to delayed values or “old ideas” entering the current pattern of activations. NW nodes hone by integrating and simplifying inputs from multiple sources, and returning a particular value. In NW, a catalyst or “catalytic value” is one that needs to be received on the inputs of one or more nodes to maintain one or more periodic structure (perhaps playing a different role in each). As NW strings notes together (often in parallel) in a stream of music, its nodes act on and react to both the nodes in their cluster, and to other clusters, via their hubs. Bottom-up and top-down feedback and time-delayed interactions are essential for an open-ended communal evolution of creative novelty.

Periodic structures are often disrupted (stopped or modified) by the introduction of a new (delayed) value, although sometimes this does not affect output. As interactions between nodes occur through entry delays, periodic musical structures unfold at different timescales. Slowly evolving periodic structures can be difficult to hear (due to intervening events) but can have a “guiding” effect on the output stream, i.e., they affect what Bimbot, Deruty, Sargent, and Vincent (2011) refer to as the “semiotic” or high-level structure of the music emerging from long term regularities and relationships between its successive parts.

NW creates musical “ideas” that become the context for their further unfolding. Asynchrony, achieved by the (dynamically changing) values of the nodes in Entry Delay Module allow previously calculated node values (including their own) to be output later in time. NW outputs both manifests the dynamics of the network, and in turn generate the dynamics. As with the autopoietic structure of a creative mind, NW is a complex system composed of mutually interdependent parts.

Let us examine how this applies to the process by which the dynamics of a NW network could be said to like a creative mind, become autocatalytically closed. The nodes collectively act as a memory in the following sense. When a

node is activated, it sums the last values received on its inputs and uses the sum to output the stored value (which is then delayed before being sent to receiving nodes). Nodes are programmed so that their individual inputs can only store or “remember” the last value received. However, because nodes have 3, 4, 5, 14 and 39 inputs (excluding their own), and the network is asynchronous, a node (as a whole) can “remember” values spread out over time. How long a node can remember depends on its own ED value and the ED values of the nodes that participate in co-determining its output. It is important to note, however, that nodes can also “forget” much of the information they receive, if, for example, it receives a number of different values on the same inputs since only the last ones are used when the node is activated. Again, how much they forget depends on its own ED value and the ED values of the nodes to which it is linked.

These NW memory patterns are distributed across the network. They are self-organizing because they can recur with variation, such that the whole is constantly revising itself. NW chains items together into a stream of related notes / note attributes. As NW strings notes together in a stream of music, its nodes are acting on and reacting to (feeding-back and feeding-forward information) to and from both the nodes in their cluster and to other clusters via their hubs. It would seem that bottom-up, top-down and time-based interaction / feedback are essential for an open-ended communal evolution of creative novelty.

There are many ways inputs can sum to the same value. Nodes with rule tables that favour certain output values are less discriminating (lower resolution). Hubs are always less discriminating since they have more inputs, but the same vocabulary (number of possible output values). Nodes with fewer inputs and an equal distribution of output values across input sums are more discriminating (higher resolution). The interaction between nodes (individuals) almost always results in an open-ended (endless) stream.

Contextual Focus

Some of NW’s music sounds uninspired; it contains no surprising pattern development (e.g., a sudden transition or gradually modulated transition in texture, mood, or tempo), and/or the patterns do not elicit innovative variations. To minimize this problem, NW uses an architecture that, in its own way, implements contextual focus. Clusters of nodes that are more interlinked and share a similar rule tables process in a more analytic mode. Hubs, which connect clusters into a small-world network and merge more distantly related musical ideas, process in a more associative mode. Because clusters have fewer inputs than hubs they are more discriminating than hubs. Hubs act as funnels, summarizing or simplifying the information its receives from multiple sources. Thus NW is hardwired to shift between analytic and associative modes by modulating the relative influence of top-down and bottom up processing.

Edge of Chaos

NW structures transform as they propagate in time, and as mentioned above, all four behavior classes have been observed. Class one and two dynamics do not change unless disrupted. When NW exhibits analytic processing, output streams flows toward class two behaviour. When NW exhibits associative processing it flows toward Class three (deterministic chaos) which does not repeat if unbounded. Class four (edge of chaos) balances change and continuity.

Network dynamics often sound chaotic at the beginning of a piece—set in motion from an arbitrary, initial configuration (‘seed incident’). Repetition and development of motivic materials and/or melodic lines then moves the system toward one or more attractor(s) (or “grooves”), resulting in a more stable, organized musical texture. Nodes with different rules of interaction are apt to disturb the system, pushing it into another basin. If it returns to a basin, a similar texture returns. When tuned to midway between order and chaos, the global stable dynamics are repeatedly disturbed. This pushes it either (1) into another basin, creating a transition to contrasting musical material, or (2) further from the attractor, to which it tries to return. NW exhibits something akin to goal seeking behaviour in how it moves toward or away from an attractor by keeping within a range of “desirable” values. This is similar to the use of functional tonality in western music, in which a piece departs and returns to its tonal center. Quasi-periodic dynamics provide a sense of organization through cycling musical textures, or a loose theme and variation structure. These disturbances may be caused by nodes with different rules of interaction, or by delayed values entering the stream. One factor that affects the aesthetic quality of the output is the mapping of the node output values to a specific ED scale (mapped values are used to delay node outputs). This appears to produce a balance between current events and older ones that is at the proverbial edge of chaos.

Conclusions and Future Directions

NW’s unique architecture—in particular, its scale-free network architecture and transparent relationship between rules of interaction (LUTs) and MIDI output—was inspired by the science of complex adaptive systems as advocated by the honing theory of creativity. The number of possible LUTs that can generate “edge of chaos” dynamics, however, is extremely large and “by hand” rule design and “by ear” verification of the results should be augmented by evolutionary programming techniques guided by quantitative analyses. NW will also continue incorporating principles of HT. In turn, grounding the theory using NW is inspiring new developments in the understanding of creativity.

Acknowledgements

Many thanks to Iris Yuping Ren for providing the entropy analysis of NetWorks music, Bach chorales and jazz tunes.

This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

- Abraham, F. D. 1996. The dynamics of creativity and the courage to be. In W. Sulis & A. Combs (Eds.), *Nonlinear dynamics in human behavior*. River Edge, NJ: World Scientific, 364-400.
- Ashby, W. 1947. Principles of the self-organizing dynamic system. *Journal of General Psychology* 37:125-128.
- Bak, P., Tang, C., and Wiesenfeld, K. 1988. Self-organized criticality. *Physical Review A* 38:364.
- Barábasi, A. L. 2002. *Linked: The New Science of Networks*. New York: Perseus.
- Bell, S. 1998. *Emergent Music I: Voices From The Edge of Chaos*. Polycarpe Studios: EM001.
- Beyls, P. 1989. The musical universe of cellular automata. In Wells, T. & Butler, D., eds., *Proceedings of the International Computer Music Conference*, 34-41.
- Beyls, P. 1990. Musical morphologies from self-organizing systems. *Interface: Journal of New Music Research* 19(2-3): 205-218.
- Beyls, P. 1991. Self-organizing control structures using multiple cellular automata. *Proc Intl Computer Music Conf*. Montreal: Intl Computer Music Assoc. 254-257.
- Bimbot, F.; Deruty, E.; Sargent, G.; and Vincent, E. 2011. Methodology and resources for a structural segmentation of music pieces into autonomous and comparable blocks. *Proc Intl Soc Music Information Retrieval Conf* 287-292.
- Bowcott, P. 1989. Cellular Automata as a means of high level control of granular synthesis. In Wells, T. & Butler, D., eds., *Proc Intl Comp Music Conf*. San Francisco: ICMA. 55-57.
- Carver, C., and Scheier, M. 2002. Control processes and self organization as complementary principles underlying behavior. *Personality and Social Psych Review* 6:304-315.
- Chareyron, J. 1990. Digital synthesis of self-modifying waveforms by means of linear automata. *Computer Music Journal*, 14(4):25-41.
- Evans J. St. B. 2003. In two minds: dual process accounts of reasoning. *Trends in Cognitive Science* 7: 454-59.
- Gabora, L. 1998. Autocatalytic closure in a cognitive system: A tentative scenario for the origin of culture. *Psychology*, 9(67). [adap-org/9901002]
- Gabora, L. 2010. Revenge of the 'neurds': Characterizing creative thought in terms of the structure and dynamics of human memory. *Creativity Research Journal* 22:1-13.
- Gabora, L. (in press). Honing theory: A complex systems framework for creativity. [Nonlinear Dynamics, Psychology, and Life Sciences](#).
- Goertzel, B. 1997. *From complexity to creativity*. New York: Plenum Press.
- Guastello, S. J. 1998. Creative problem solving at the edge of chaos. *Journal of Creative Behavior* 32:38-57.
- Horner, A. & Goldberg, D. E. 1991. Computer-Assisted composition. *Fourth International Conference on Genetic Algorithms*. University of California, San Diego, CA.
- Horowitz, D. 1994. Generating rhythms with genetic algorithms. *Proceedings of the International Computer Music Conference*. Aarhus, Denmark.
- Kitzbichler, M. G., Smith, M. L., Christensen, S. R., and Bullmore, E. 2009. Broadband criticality of human brain network synchronization. *PLoS Computational Biology*, 5.
- Liu, X. F, Tse, C. K., and Small, M. 2010. Complex network structure of musical compositions: Algorithmic generation of appealing music. *Physica A* 38: 126-132.
- McCormack, J & d'Inverno, M. 2014. On the Future of Computers and Creativity. Symposium on Computational Creativity, AISB: Goldsmiths, University of London.
- Millen, D. 1992. Generation of formal patterns for music composition by means of cellular automata. In Strange, A., ed., *Proc Intl Comp Music Conf*. San Francisco: ICMA, 398-399.
- Miranda, E. R. 1995. Granular synthesis of sounds by means of a cellular automaton. *Leonardo* 28(4):297-300.
- Mitchell, M. 2006. Complex systems: Network thinking. *Artificial Intelligence*, 170(18): 1194-1212.
- Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. 2004. The University of South Florida free association, rhyme, and word fragment norms *Behavior Research Methods, Instruments, & Computers* 36(3):402-407.
- Pribram K.H. 1994. *Origins: brain and self-organization*. Hillsdale NJ: Lawrence Erlbaum.
- Prigogine, I., & Nicolis, G. 1977. *Self-organization in non-equilibrium systems*. New York: Wiley.
- Ren, I Y. 2014. *Complexity of Musical Patterns*. University of Warwick.
- Ren, I Y. 2015. Using Shannon Entropy to evaluate automatic music generation systems: A Case Study of Bach's Chorales. ECE Department, University of Rochester.
- Runco, M. 2010. Divergent thinking, creativity, and ideation. In J. Kaufman and R. Sternberg, (Eds.), *The Cambridge handbook of creativity*. Cambridge UK: Cambridge University Press, 414-446.
- Sowden, P., Pringle, A., and Gabora, L. 2015. The shifting sands of creative thinking: Connections to dual process theory. *Thinking & Reasoning* 21:40-60.
- Steyvers, M., and Tenenbaum, J. B. 2005. The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science* 29:4-78.
- Todd, P. M. and Loy, G., eds. 1991. *Music and Connectionism*. MIT Press.

BEYOND THE FENCE

A NEW MUSICAL



Has computational creativity successfully made it ‘Beyond the Fence’ in musical theatre?

Anna Jordanous

School of Computing, University of Kent,
Chatham Maritime, Medway, Kent, UK
a.k.jordanous@kent.ac.uk

Abstract

Beyond the Fence is a commercial project, undertaken for a television documentary, that has produced a musical show billed as “the world’s first computer-generated musical”¹. Several computational creativity systems have been used in the production of various parts of this musical, which has been performed in London’s West End for a two week run in 2016. Having been involved in this project as an informed commentator who was not involved in creating any of the software, I consider two questions that together form the main contribution of this paper: (1) To what extent is the project successful? and (2) To what extent does this project demonstrate computational creativity? Investigations into these questions show that *Beyond the Fence* has successfully shown how existing creative software can indeed be used to create a plausible and acceptable musical. The resulting musical has been moderately well-received by most critics, though standards are raised high for computational creativity in the public eye. The project has also raised the profile of computational creativity research. Some useful lessons have also been highlighted for computational creativity; in particular, computational creativity should include more than merely replicating norms, and completing independent tasks within the creative process (with little feedback or collaboration between tasks). The impact for computational creativity is that for these larger scale multi-system public-facing projects to be more successful, we are reminded of the need to develop as well as replicate human creative achievements, and to allow our systems to be able to communicate and refine work as well as offer inspirational material.

Introduction

In musical theatre, those involved in shows are accustomed to being reviewed. Essentially this paper provides a review of the 2016 musical *Beyond the Fence* - but a review with a difference. *Beyond the Fence* has been billed as “the world’s first computer generated musical”.² With several computational creativity software packages and computational data analyses providing data, frameworks and content for the musical in collaboration with (human) musical theatre experts,

¹<http://www.wingspanproductions.co.uk/news-and-awards/read/48/Beyond-the-Fence-the-world-s-first-computer-generated-musical> (Mar’16).

²<http://beyondthefencemusical.com/> (Mar’16).

Beyond the Fence tests the theory of whether computational creativity can be used to create a musical. This work was undertaken by the television production company Wingspan Productions (led by Dr Catherine Gale) for a two-part documentary about the process, commissioned by a UK satellite channel (Sky Arts) with support from Wellcome Trust.

At the time of writing, the musical is coming to the end of a two-weeks-long run in London’s West End, an area of London with an extremely vibrant theatre and musical theatre scene (to the extent that this part of London is colloquially referred to as ‘Theatreland’). Having been involved in *Beyond the Fence* as an informed commentator who was not involved in creating any of the software, I have had the opportunity to gather and discuss information about this project with a variety of different sources, from people behind the software to the cast performing the show. From this perspective, two interrelated questions have emerged:

1. To what extent is *Beyond the Fence* successful?
2. What does this contribute to computational creativity?

These questions guide this paper, in evaluating the *Beyond the Fence* project via each of these questions. Some details are given about how *Beyond the Fence* has been undertaken and what creative entities have been involved.

To evaluate the project in a computational creativity context and address the above two questions, it is treated as an example of interactive creativity and is evaluated using a framework advocated for this type of creativity (Kantosalu, Toivanen, and Toivonen 2015). During evaluation, in conjunction with personal communications with Gale (2016) on how to judge success, some questions are considered in this paper as possible metrics for gauging success. Accompanying this, various traditional metrics for success in musical theatre are explored below. Evaluation affords us insight on the extent to which this project has been a success, and the engagement of this project with computational creativity research.

The paper concludes with a discussion of what the field of computational creativity can learn from the *Beyond the Fence* project. Where has computational creativity successfully contributed to *Beyond the Fence*? What has not worked so well in terms of computational creativity’s application to this problem of creating a new musical? And where would future work in this direction be most usefully directed?

Details of the project

“September 1982. Mary and her daughter George are celebrating one year of living at the Greenham Common peace camp. The group of women they have joined are all committed to stopping the arrival of US cruise missiles through non-violent protest. When Mary is faced with losing her child to the authorities, an unlikely ally is found in US Airman Jim Meadow. How can she continue to do what is best for her daughter while staying true to her ideals? *Beyond the Fence* is a powerful new musical about hope, defiance, unity and love.”³

Another story of interest to a computational creativity audience, however, is not the end product itself but the process and interactions that took place between different creative entities to create this musical. This process was driven by a team from Wingspan Productions, who describe the engagement with computational creativity as follows:⁴

“The process began with a predictive, big data analysis of success in musical theatre, conducted by Dr James Robert Lloyd, Dr Alex Davies and Prof Sir David Spiegelhalter (Cambridge University). They interrogated everything from cast size, to backdrop, emotional structure to the importance of someone falling [in] love, dying (or both!) - in more and less successful shows - to create a set of constraints to which the musical had to conform, to theoretically optimise chances of success. Next, the team visited what’s known as the What-If Machine at Goldsmiths, University of London. With Prof Simon Colton, Dr Maria Teresa Llano and Dr Rose Hepworth at the helm, the machine generated multiple central premises, featuring key characters, for the new show. The team selected this as the starting point and the original idea for the musical: What if a wounded soldier had to learn how to understand a child in order to find true love?”

A plot structure for the musical was also generated computationally, thanks to work led by Dr Pablo Gervás (Complutense University of Madrid). A brand new analysis of musical theatre narratives enabled him to adapt an existing story telling computer system, called PropperWryter, to turn its hand to musicals and build the core narrative arc of the new show.

Taken together, all of the above enabled the precinct for the emerging story to be identified: Greenham Common. The team then wrote a book and lyrics (with the assistance of some other computational tools) that fitted all these constraints.

Finally, the music material has been provided by Dr Nick Collins (Durham University), who has created a computer composition system based on a machine listening analysis of musical theatre music, conducted by Dr Bob Sturm (QMUL) and Dr Tillman Weyde (City University). Additional computer music material [was] generated using the FlowComposer system created by Dr Pierre Roy and Dr Francois Pachat (SonyCSL, Paris).⁵

In the credits for the musical, the ‘creative team’ listing includes the software programs involved and key researchers

³<http://beyondthefencemusical.com/about-the-show> (Mar’ 16).

⁴A paper has been produced by the teams involved, giving fuller details of how *Beyond the Fence* was constructed. This paper is not intended to duplicate these descriptions, but to critique the project from an independent perspective.

⁵<http://beyondthefencemusical.com/the-science> (Mar’ 16).

on each piece of software, plus two human musical theatre experts (Benjamin Till and Nathan Taylor), who curated the software outputs into its final musical format. The accompanying documentary shows how human members of the creative team took care to adhere to the spirit of this project: using as much computer-generated material as possible even when this caused difficulties. As Neil Laidlaw (producer of the stage performance) says, “we have to honour what we’ve signed up for” (Wingspan Productions 2016).

During investigations, Gale and her team became interested in what ‘being creative’ actually means and how creativity might emerge from rules and be assessed, given the difficulties we have in assessing creativity in humans. One particular debate Gale looked at in her conversations with computational creativity researchers relates these thoughts directly back to *Beyond the Fence*: does a generative process have to result in a good quality product (i.e. the musical) to qualify that process as having been creative? This distinction between the process and the generated artefact when assessing/recognising creativity has often arisen in computational creativity research (Gervas 2009; Jordanous 2016, for example). For *Beyond the Fence*, then, can the project be considered successful even if the generated show is not well received as a musical in its own right?

Evaluation

To what extent is *Beyond the Fence* successful and what does this contribute to computational creativity? To evaluate both of these questions, we consider *Beyond the Fence* as a case of *interactive creativity* generating a musical. Evaluation models typically focus on evaluating a single system, but Kantosalo, Toivanen, and Toivonen posit the DECIDE framework (Rogers, Sharp, and Preece 2011) as a model of evaluation suitable for evaluation of several systems collaborating and being co-creative with humans; this is the scenario we have with the creative systems used for *Beyond the Fence*. Hence to evaluate the interactive creativity in this project, following Kantosalo, Toivanen, and Toivonen (2015), we use the DECIDE framework:

1. “Determine the goals
2. Explore the questions
3. Choose the evaluation methods
4. Identify the practical issues
5. Decide how to deal with the ethical issues
6. Evaluate, analyze, interpret, and present the data”

DECIDE: Determine the goals The Wingspan Productions team conducted this project to explore if a ‘computer-generated musical’ was possible. Specifically, their goal was to create and stage a musical generated in collaboration between creative software and human musical theatre experts.

DECIDE: Explore the questions As part of this project, the Wingspan Productions team sought out and engaged with several leading computational creativity researchers, as described above. During this process, the team grew more interested in various debates and issues around computational creativity, and how the *Beyond the Fence* project sits in the wider context of computational creativity. During the process, Gale and her team explored how the *Beyond the Fence*

related to various key areas of concern in computational creativity research. To gain better understanding of computational creativity research, specific questions emerged (Gale, 2016, personal communications):

1. “How has our attitude towards how we use computers changed in recent decades?”
2. Why do people develop machines that are creative?
3. So is it right to paint a portrait of machines as a young artist? One that is maturing?
4. What kind of systems have been/are being developed?
5. What different approaches do people take? ‘Heroic’ methods where computer is [an] artist in its own right, or more collaborative approaches?”⁶

In conversations with various computational creativity researchers, including myself, Gale also investigated questions about the role of computer software in the ‘creative conversation’: “Computers can become another voice in the room - speaking ‘from the data’ as it were - and we instinctively question that” (Gale 2016, personal communications). Gale observed through this experiment that people are often surprised at the challenges and difficulties in computationally generating creative artefacts - perhaps underestimating the complexity of the tasks involved. She also saw resistance in people’s reactions to computers being creative. One example of this resistance is illustrated in the documentary that reports on this project (Wingspan Productions 2016). Benjamin Till reflects a number of times on his apprehensions about working with computational software, such as this quote from his interactions with the products of the *Android Lloyd Webber* music-writing system. He says: “maybe I was a little bit harsher on it than I should have been” (Wingspan Productions 2016), going on to explain that because *Android Lloyd Webber* was computer software, he partly felt that he did not want the results to be good.

Concentrating on the *Beyond the Fence* musical project itself, Gale and her team were interested in what computational creativity researchers thought about the project; such discussions receive attention in the documentaries resulting from this project (Wingspan Productions 2016). For example, Gale was interested in whether the *Beyond the Fence* project was doing work that was in some way different to existing current work in computational creativity, or work that was exciting for the field. Relevant aspects that emerged in such discussions included the collaborative aspects of the creation of the musical, and the scale of the overall project (especially as the project resulted in public performances presented in a venue in a high profile London location.).

DECIDE: Choose the evaluation methods For this experiment, what will constitute a success? The question of whether there was an underlying hypothesis for the project was raised in a question-and-answer session post-performance I took part in with Catherine Gale (Wingspan Productions), Bob Sturm and Benjamin Till (respectively representing computational and human parts of the music generation team) on 2nd March 2016. This discussion revealed some difficulty in pinning down an exact scientific hypothesis for the project (one by which the project success

⁶Probably inspired by d’Inverno and McCormack (2015).

could be tested against and/or measured). However in this discussion and in other personal communications with Gale (2016), two different ways of considering the success of the project emerged: (1) traditional evaluation of success in musical theatre, via reviews and other metrics (given the information available at the time of writing); and (2) the wider contributions of the project to pursuits of knowledge, independently of the success of the musical.

The big data analysis by the Cambridge team distinguished between ‘hits’ (culturally and commercially successful), ‘flops’ (culturally and commercially unsuccessful), ‘critically acclaimed’ musicals (successful culturally but not commercially) and ‘crowd pleasers’ (successful commercially but not culturally). Commercial success is relatively easy to gauge after a show has been performed for a length of time: for example by looking at the amount of money a musical makes, the length of its run, the number tickets sold and any touring the show does. Typically, cultural value is more subjective and so trickier to measure objectively. While people involved in the project have reported that they felt the show was a success (Wingspan Productions 2016) (and Taylor, 2016, personal communications), empirical metrics are possible (Jordanous, Allington, and Dueck 2015). For musical theatre, cultural value can be measured via awards, press attention, reviews, influence on other shows, audience reaction, pick-up by amateur companies, location of show venue, funding, and other non-commercially measurable aspects.

DECIDE: Identify the practical issues Practical issues in the *Beyond the Fence* project poses an interesting challenge in terms of evaluation of the creativity of the software involved. Multiple software was used during this project, as well as interactions with creative people. Hence we could either evaluate the creativity of each constituent software-based part of the creative team, or focus on evaluating the overall collective project. As this paper’s aim is to evaluate the multiple parts of the project as a whole, it focuses on the latter aim; the task of evaluating individual software falls better to papers that report on individual software.

The Standardised Procedure for Evaluating Creative Systems (SPECS) (Jordanous 2012) asks researchers to (1) identify a characterisation of creativity by which to evaluate creative systems (2) derive standards or benchmarks by which to measure our systems, and (3) devise suitable tests to evaluate our systems against these benchmarks. We have a number of models available to us that we could use as our base characterisation of creativity for Step 1. For example, the components of creativity derived in (Jordanous 2012) represent a general basis for creativity. Alternatively, the FACE model (Colton, Charnley, and Pease 2011) analyses creative systems on their ability to use and/or generate new methods for using Frames (natural language descriptions of what the software has done), Aesthetics (measures), Concepts (underlying theory/ies that guide the creative process) and Example outputs. The Creative Tripod (Colton 2008) asks whether the system under question can be considered as a candidate for a creative system, through identifying the system’s ability to demonstrate skill, imagination and appre-

ciation. Ritchie has devised empirical criteria for evaluating creative systems (Ritchie 2007), though the criteria are less applicable for this evaluation as they are based upon ratings of the typicality and value of multiple artefacts produced by a creative system, whereas we only have one output to judge.

Considering the interactive nature of the *Beyond the Fence* project, it is difficult to generalise our currently available models across multiple systems acting collaboratively. There is also the question of whether to include human parts of the creative team. These issues make the above-mentioned existing evaluation methodologies and methods difficult to apply in this evaluation.⁷ One criterion of what makes a good methodology for evaluating creative systems (Jordanous 2014) is the usefulness of feedback. As we have seen, the *Beyond the Fence* project comprises multiple software, each tackling different types of creative task. This illustrates the vast and varied scope of creativity in musical theatre. For this current work, this evaluation is intended to uncover formative feedback for future development and recognition of this work's contribution to research.

DECIDE: Decide how to deal with the ethical issues Gale (2016, personal communications) has reflected on whether computers could (and should) be considered creative entities, at a level which is comparable or equal to humans. She discussed with various computational creativity researchers what might make people working in the creative industries more receptive or better suited to collaborate with computers than others, and questioned how to talk to creatives about their attitudes, perceptions and potential biases towards working with computers. Certainly this latter question deserves greater attention if computational creativity research is to reach a wider audience and broader range of collaborative partners. Some interesting points have already been raised which we can build upon such as how people may perceive the creativity of a system by looking for key aspects that a system should generate if it might be described as 'creative' (Colton 2008); and the role of people's reactions on interacting with creative systems, as a key contribution to that system's creativity (Gervas 2009; Gervás and León 2014; Jordanous 2016). Computational creativity researchers could also consider to what extent it is reasonable (or productive?) to attribute creative agency to a computer when featuring computational creativity software in public engagement activities, following discussions on creative agency and creative responsibilities (Maher 2012; Johnson 2014; Bown 2015).

Ethically, there are also challenges for people in experiencing *Beyond the Fence* as a computer-generated artefact - as we saw, many of the reviews mentioned a feeling of disconnect at times - something missing from the experience. This was discussed in the previous section. Another ethical issue relates not to the human participants in the creative team, but the computational participants. Is it fair to test computational systems at a professional level, where they

⁷Tackling the question of what is the most faithful model of creativity in this particular scenario would be interesting and challenging work to carry out as a follow-up project in its own right.

are being required to generate material at a standard which it takes humans years to reach (founded upon decades and more of human experience more generally?) Typically, evaluation of computational creativity systems has been undertaken on a more controlled and less professional level, away from the public eye - although there are notable exceptions to this as exemplified by the *Painting Fool* (Colton 2012) or the *Unnatural Selection* (Eigenfeldt 2015), both of which have recently 'participated' in public professional displays of their creativity (exhibitions and concerts respectively).

One other issue relating to the computational participants is the level at which we are evaluating them. Each software takes on a creative task assigned to it, but here we judge the overall success of the project rather than the success of each task. What we do not consider is the success to which individual tasks have been identified, and to what extent the software fulfils any original requirements. If poor decisions have been made and a vital part of the task overlooked, the computational participants may be judged more harshly as a result, even though they were not asked to perform this missing part of the task. The global focus of this current analysis can only touch on individual system issues; we leave more detailed analyses of success at individual tasks up to the researchers behind the systems involved. Here we focus on what we can learn from the project as a whole.

DECIDE: Evaluate, analyze, interpret, and present the data As outlined above, we consider the cultural and commercial value of the *Beyond the Fence* musical, considering both the success of the show itself as a piece of musical theatre and by the contributions made to computational creativity.

Data on commercial benchmarks for success of *Beyond the Fence* is not yet available. However, from informal feedback, one of the creative team reported that the show was getting good audiences every night, which he had been able to observe since he had attended every show to date (due to having different guests coming to see every show - a mark of cultural interest in its own right). The creative team member was happy about this observation, particularly as the show was being performed in a London 'West End' venue.

In terms of cultural value, the timing of this paper also means it is not yet possible to reflect on awards, influence on other shows or whether the show is picked up by amateur companies. However we can already see that the show is being performed in (and supported by) the Arts Theatre, London, a reasonably well-known 'indie' theatre in a part of London strongly connected to theatre. The overall project was supported by the Sky Arts television channel as well as Wellcome Trust funding. As reported by *the Londonist* publication, Phil Edgar-Jones (Director of Sky Arts) was very positive about this "fascinating project that we're extremely proud to be a part of. At Sky Arts, we're always excited by innovation and this venture offers an intriguing glimpse into how technology is changing music evolution. Can an algorithm create music with all the humanity, emotion and drama that a person can bring? This question captivates us. We cannot wait to see the result."⁸

⁸<https://www.londontheatre1.com/news/121796/new->

An exhaustive search of Google results towards the end of the musical's run reveals reasonably extensive national/local press attention in the form of 20 reviews, from specialist musical theatre sources to national newspapers. The remainder of this section focuses on an analysis of key issues mentioned in these reviews. Figure 1 shows the most frequently-occurring words seen in this review corpus. A large proportion of these words relate to the content of the musical, as seen in reviews of more typical musicals. Reviews contain many comments relating to work by humans in this musical, such as the strong cast. However, words such as 'computer' and 'experiment' in this word cloud illustrate that these reviewers are well aware of the computational origins of *Beyond the Fence*; several interesting points are examined.⁹

Reviews of success As one reviewer comments, the *Beyond the Fence* project is indeed “[s]eemingly wanting to be judged as the output of an experiment rather than a ‘proper show’ ” (BroadwayBaby, see Table 1). While this seems like a criticism, it is actually not too far from the truth as an appraisal of the project's aims (Gale, 2016, personal communications). Typically, reviews focused on judging the show on its quality and fit as a “proper show”: “Computers can help write a musical, it seems, but they can't yet write a good one” (Engadget UK). “This show is as bland, inoffensive, and pleasant as a warm milky drink” (Guardian). The Londonist was a little more encouraging: “[w]hat's our measure for success? Well, the audience we saw *Beyond The Fence* with was applauding just about every number and was brought to tears by at least two of them.”

Several reviewers reflected at the experiment as a whole as part of their reviews. What's On Stage asked if computers can create more challenging material and concludes: “[o]ne day, maybe, but not yet. Not yet.” The Financial Times reviewer judges that “this bold experiment doesn't solve the many contradictions it throws up.” Similarly, the Reviews Hub says that “*Beyond the Fence* is an interesting experiment but it shows that computers are a long way off from creating a hit musical.” “Here, the puppet masters' digital strings are still a little too visible” (Musical Theatre Review). The West End Frame concludes that “as a theatrical event the show is remarkable. However, as a piece of theatre in its own right *Beyond The Fence* doesn't stand strong; it feels contrived and clunky.” But perhaps the computational creativity community can feel more heartened for future work here: “it is an interesting development in the intersection between theatre and technology that I suspect we haven't heard the last of” (‘There ought to be clowns’ blog).

The validity of co-creativity in *Beyond the Fence* “This experiment ... has plainly benefited from a lot of human intervention ... To call it ‘computer-generated’ is misleading. “Computer-initiated” and “computer-assisted”, though less grabby, are more accurate - and in their own way provide

computer-generated-musical-beyond-the-fence-to-premiere-at-the-arts-theatre-in-2016/ (Mar' 16).

⁹For sources for each review quote here, see Table 1.

a thought-provoking novelty” (Telegraph). A number of reviewers commented on how human members of the creative team “are, essentially, curating and correcting the computers' output” (What's on Stage). Rarely if ever do the reviewers allow the software participants any creative agency or responsibilities for their output. Instead of being treated as co-creators in an interactive creative situation, computers are often portrayed in reviews as learners rather than creatives, whose work the human participants are being asked to bring up to professional level for the final product. In this project the computer software is not able to engage in revision, respond to feedback (particularly as musicals can change from night to night in response to feedback) - they provide material for people to curate. “[h]ere is where the computer generated claim starts to unravel. There's no software that can put all of these elements together and turn them into a musical. That requires a human”¹⁰ (Engadget UK).

One notable and fascinating exception to this generalisation is by Musical Theatre Review: ““What if a wounded soldier had to learn how to understand a child in order to find true love?”” was generated by WHIM, the “What If Machine”. And in tribute, Ako Mitchell's US soldier Jim rubs his thigh in pain occasionally - less wounded soldier, more bloke who should have done a few more warm-ups before exercising. There's an emotional need for him that is implied in WHIM's question that is not addressed here, leaving the show's central love story to feel a little anaemic.” In other words, software has supplied a creative idea to its human collaborators that is not used well, to the detriment of the overall effect of the show.

Too formulaic? “[A]ny show built according to a formula runs the risk of sounding, well, formulaic. *Beyond the Fence* doesn't avoid this risk: despite the cutting-edge technology involved in its creation, the show itself is middle-of-the-road and predictable.” (Financial Times). “Follow a formula and - who would have thunk it - you get something formulaic” (What's on Stage). Most of the reviewers criticised the musical for feeling too pattern-driven and predictable, rather than including content to challenge rather than replicate musical theatre. Several reviewers criticised the musical for its lack of avant-garde, challenging or ground-breaking content. “Picking through old shows for tricks evidently means the plotting is predictable and at times a bit shallow” (Londonist). The issue with this type of criticism is that the software involved was developed on the task of “replicating the past, not challenging it” (What's On Stage). “Nothing moved the needle. Nothing felt fresh.” (Engadget UK).

Where computer-generated aspects moved away from typical output, this also attracted criticism. For example several reviewers criticise the lyrics for being atypical - an ironic example of this is where during such criticism, a reviewer highlights the ‘We are Greenham’ song as one of the songs “that work” (Musical Theatre Review), even though that song consists entirely of computer-generated lyrics.¹¹

¹⁰A provocative request for future research?

¹¹It should be noted that this song's lyrical content was generated differently to many other songs, using corpus analysis of protest

Sometimes criticism of machine-generated content permeated into parts of the production humans had responsibility for: one reviewer suggested that the live band (of human performers, performing music that had been orchestrated by humans), “sounded extremely robotic and monotonous - it sounded as if backing tracks were being used” (West End Frame). The Financial Times said, though: “writers have been using formulae for years to make commercially-minded culture and so what difference does it make if it’s a formula developed by a machine? This is the main talking point of the show ... a debate that will run and run.”

Lack of context awareness? As the Guardian reviewer observed, “The software appears to have ... zero grasp of 1980s feminism and the Greenham Common women’s peace camp. A pity, because that’s where it’s set.” (Guardian). But is this a fair criticism, given that the Greenham Common setting and feminist themes were chosen by the humans in the creative team rather than via software? Certainly many of a younger generation of musical theatre professionals could also be criticised for not knowing about this particular event in UK history, and the software is given no chance to research these themes post-decision to use them for *Beyond the Fence*. Criticising the computer participants for not being more knowledgeable seems harsh. But should (and could) the computers have used more contextual awareness, to develop ideas based on contextual information available e.g. via web resources? It seems from a number of reviewers’ comments that this wider contextual knowledge was expected, for example where reviewers criticised the show for not engaging more with feminism issues, or debates about nuclear weapons that concurrently happened at the same time as the musical was being performed. More than one reviewer commented that they would have liked to see a plot centred around scenarios a computer might have some knowledge of (e.g. the Financial Times suggest “circuit boards in revolt”), though no follow-on comment considers how human audiences might perceive the results.

One area where the computational creativity software was roundly criticised was in the ability to understand content unfolding over time, in longer-term structures. “Even if they give you a stroke of genius, they can never follow that up... every thought is a new thought” says Benjamin Till, in (Wingspan Productions 2016), where Nick Collins also reflects that this is an area for further research.

Gimmick by boffins? Biases and preconceptions It was interesting to see several reviewers make negative comments to the effect that poor human-produced musicals appear as if they were written by a machine, e.g. “Plenty of musicals written by humans sound as if they have been composed by a machine” (Guardian). The Telegraph reviewer reported how “the evening somehow over-rode my default scepticism”; others made more negative comments about ‘gimmicks’ introduced by ‘boffins’. Interestingly, during development, the work-in-progress musical was performed to a test audience of regular theatre-goers who were unaware of the origins of songs from Greenham Common rather than the Cloud Lyricist.

much of the material being computer-generated. The audience reacted positively to the preview performance, but what is more telling is their reaction once they were informed about the computational work and its contribution to the musical (Wingspan Productions 2016). They reacted with stunned silence, followed by nervous laughter. Preconceptions about what computers can (and cannot) do are there to be dealt with, in computational creativity - the reviews here show that this issue should not be ignored when engaging with the public in computational creativity research.

Conclusions and future directions

“Beyond the Fence is conceived by computer and substantially crafted by computer.”¹²

The *Beyond the Fence* project has achieved the goal of staging a musical which has been generated through collaboration between creative computer software and human musical theatre experts. At the time of writing, the *Beyond the Fence* musical is reaching the end of its two-week run in London’s ‘Theatreland’ (the West End district of London, UK). The premise, plot elements, storyline, music and approximately a quarter of its lyrics were computer-generated, using various creative systems in conjunction with human experts. The project has been recorded in documentary form (Wingspan Productions 2016). The human creatives involved have reported that they feel the project has been a success, and the show has been performed to good-sized and receptive audiences each evening of its run. While “this bold experiment doesn’t solve the many contradictions it throws up” (Financial Times, see Table 1), it has made these “contradictions” and debates open and relevant for discussion among a much broader audience.

What can the field of computational creativity learn from the *Beyond the Fence* project? Where has computational creativity successfully contributed to this project? What has not worked so well in terms of computational creativity’s application to this problem of creating a new musical? And where could future work in this direction be directed?

In the short term, this project has played an important part in raising the profile of computational creativity research. This project has taken on an ambitious task, and has tasked computational creativity researchers with applying the software we create both as individual pieces of software and (importantly for computational creativity) in combination with other systems. While some work has been done in combining different creative systems for a broader perspective on creative tasks (Monteith et al. 2011, e.g.), the idea of different creative systems communicating and/or collaborating with each other (Corneli et al. 2015, e.g.) is an exciting area to look at (especially now many different creative systems have been developed and are potentially at our disposal).

The project led the Wingspan team to become interested in questions to do with creativity in different domains. Perhaps because of her biochemistry research background, Gale particularly focused on people’s perceptions of creativity outside of artistic domains, and cultural issues that may affect how we distinguish between creativity in scientific do-

¹²<http://beyondthefencemusical.com/about-the-show> (Mar’16).

Table 1: The twenty reviews of *Beyond the Fence* sourced via search, for analysis in this paper

Source	Link
NATIONAL NEWSPAPERS	
Guardian	http://www.theguardian.com/stage/2016/feb/28/beyond-the-fence-review-computer-created-musical-arts-theatre-london
Telegraph	http://www.telegraph.co.uk/theatre/what-to-see/beyond-the-fence-arts-theatre-review-computer-says-so-so/
Financial Times	http://www.ft.com/cms/s/0/5f993b32-dee2-11e5-b67f-a61732c1d025.html
Independent	http://www.independent.co.uk/arts-entertainment/theatre-dance/reviews/beyond-the-fence-arts-theatre-review-despite-my-reservations-i-was-won-over-a6900836.html
The Times	http://www.thetimes.co.uk/tto/arts/firstnightreviews/article4702133.ece
SPECIALIST THEATRE PUBLICATIONS	
The Stage	https://www.thestage.co.uk/reviews/2016/beyond-the-fence-review-at-arts-theatre-london-futuristic-composition-with-traditional-problems/
What's on Stage	http://www.whatsonstage.com/london-theatre/reviews/beyond-the-fence-arts-theatre_39847.html
The Reviews Hub	http://www.thereviewshub.com/beyond-the-fence-arts-theatre-london/
Musical Theatre Review	http://musicaltheatreview.com/beyond-the-fence-arts-theatre/
West End Frame	http://www.westendframe.com/2016/02/review-beyond-fence-at-arts-theatre.html
British Theatre Guide	http://www.britishtheatreinfo.info/reviews/beyond-the-fence-arts-theatre-12609
Theatreworld Internet Mag	http://www.theatreworldim2.com/#!/beyond-the-fence-arts-theatre/fcmry
West End Wilma	http://www.westendwilma.com/beyond-the-fence-review/
BroadwayBaby	http://www.broadwaybaby.com/shows/beyond-the-fence/710587
Carns Theatre Passion	http://carnstheatrepassion.com/2016/02/27/beyond-the-fence-arts-theatre-west-end-until-5th-march/
EVENTS LISTINGS WEBSITES	
Londonist	https://londonist.com/2016/02/computer-penned-musical-beyond-the-fence-reviewed
There Ought To Be Clowns	http://oughttobeclowns.blogspot.co.uk/2016/02/review-beyond-fence-arts.html
Time Out	http://www.timeout.com/london/theatre/beyond-the-fence
TECHNICAL BLOGS	
Engadget UK	http://www.engadget.com/2016/03/02/beyond-the-fence-computer-generated-musical/

mains compared to artistic domains (Gale 2016, personal communications). Although this current project concentrated on creativity in musical theatre, it will be intriguing to see the directions of any future projects by Wingspan Productions concerning computational creativity.

Returning to the current project under discussion, what contributions does this *Beyond the Fence* experiment make to the computational creativity field: currently and longer-term? And given directions in computational creativity research, what might this musical be like in a few years?

Discussions during filming centred around a key point in the generation of the musical: in Gale's words: "right now, [do] humans have to be part of a project like this? [Do we] need some people in the mix to curate? and to make choices? as currently the computers involved can't censor their output very well, and they don't talk to each other yet either!" (Gale 2016, personal communications). Recent computational creativity research has focused on how creative computer systems might be able to interact with each other, communicate and give each other feedback towards refining and developing their own creative work (Gervás and León 2014; Román and Pérez y Pérez 2014; Corneli and Jordanous 2015). One exciting potential area for future work is in using computational creativity to carry out this 'curation' step. Can computational creativity software curate parts of a musical (lyrics, plot, characters, emotional timelines) into a single production? To what extent is interaction with a human(s) necessary in this process? Responding to the criticisms raised in reviews that *Beyond the Fence* is not computer-generated, but 'computer-assisted' or 'computer-

initiated' (as discussed above): to what extent can computer software actually generate the full content of a musical? Could software do everything? And if several systems are involved, how can we best evaluate the results? Could social media comments also be harnessed for evaluation - perhaps as a way of garnering instant feedback which can inform software in making edits to the show for the next evening's performance? Or does this lead us into a trap where we judge the programs doing tasks set by humans, via opinions of the end result rather than the success at each smaller task (without evaluating decisions taken on how to break complex creative tasks into subtasks)?

In this current project, the computer software are for the most part passive participants, in a process curated by humans. Essentially, as discussed above, the computational participants typically contribute material that is shaped by the human participants in the creative team, and the human participants have the final say in what makes it to the 'final cut'. Perhaps, to paraphrase the title of this musical, the 'fence' in musical theatre represents the recognition of computers as genuine creative participants contributing to the creative process. In this interpretation, the *Beyond the Fence* project does not fully see creative software moving 'beyond' this 'fence'. But certainly the debate on computers being creative has been opened up to wider public scrutiny, a debate to which the project makes a significant contribution.

To summarise: the *Beyond the Fence* project has been successful at showing how existing creative software can indeed be used within the scenario of creating a and acceptable musical, which has been moderately well-received by most crit-

The *Beyond the Fence* Musical and *Computer Says Show* Documentary

Simon Colton^{1,2}, Maria Teresa Llano¹, Rose Hepworth¹, John Charnley¹,
Catherine V. Gale³, Archie Baron³, François Pachet⁴, Pierre Roy⁴, Pablo Gervás⁵,
Nick Collins⁶, Bob Sturm⁷, Tillman Weyde⁸, Daniel Wolff⁸ and James Robert Lloyd⁹

¹Goldsmiths College, London, UK ²Falmouth University, UK ³Wingspan Productions, London, UK

⁴Sony Computer Science Laboratory, Paris, France ⁵Universidad Complutense, Madrid, Spain ⁶Durham University, UK

⁷Queen Mary University of London, UK ⁸City University London, UK ⁹Qlearsite Organisational Science, UK

Abstract

During 2015 and early 2016, the cultural application of Computational Creativity research and practice took a big leap forward, with a project where multiple computational systems were used to provide advice and material for a new musical theatre production. Billed as the world's first 'computer musical ... conceived by computer and substantially crafted by computer', *Beyond The Fence* was staged in the Arts Theatre in London's West End during February and March of 2016. Various computational approaches to analytical and generative sub-projects were used to bring about the musical, and these efforts were recorded in two 1-hour documentary films made by Wingspan Productions, which were aired on Sky Arts under the title *Computer Says Show*. We provide details here of the project conception and execution, including details of the systems which took on some of the creative responsibility in writing the musical, and the contributions they made. We also provide details of the impact of the project, including a perspective from the two (human) writers with overall control of the creative aspects the musical.

Introduction

There are very few types of cultural creations more complex than a musical theatre show. The very basics required to design and produce a new musical include the creation of: an overall concept, a narrative arc, characters, dialogue and plot lines; music, orchestration and lyrics; set design, artwork, and lighting routines; and finally crafting the overall performance through rehearsals and direction, along with producing advertising material to promote the event. While stand-alone computational systems able to create artefacts in many of these areas exist, it is quite far beyond the state of the art of Computational Creativity practice to imagine the full creation of a musical via a single creative system or even by multiple systems under some automated directorial control. Hence a fully computer generated musical stands as a grand challenge for our field, which can serve to drive technologies forward, broaden public understanding of Computational Creativity and highlight new areas of research.

In late 2014, Wingspan Productions, a London-based television production company (wingspanproductions.co.uk), began to develop the idea of devising and filming an experiment in which software was used as much as possible to take on some of the creative responsibilities of writing a musical theatre piece. The project was carried through to completion

successfully, and the end results were: (a) the staging of the *Beyond The Fence* musical in the Arts Theatre in London's West End in February/March 2016, the design of which was heavily influenced by software analysis of musical theatre data, and by creative software which contributed an original concept, plot lines, lyrics and music, and (b) two 1-hour television documentaries for Sky Arts entitled *Computer Says Show*, charting the making of the musical, with emphasis on how software was used in the creative process and on the journey of discovery that the two (human) writers went on within an experiment where they were asked to use the software's advice and output as much as (humanly) possible.

Below, we provide first person overviews of the conception of the project, from the Wingspan Productions team. Following this, we describe the analytical and generative systems which contributed material used to both guide the writing of the musical and explicitly in the final production, in terms of how the systems operate, how they were employed and what they produced. We also provide details of the cultural impact of the staging of the show and the airing of the two documentaries. This includes details of the press coverage, critical reviews and a perspective on working with creative software by the two writers of the show. We conclude by highlighting the importance of projects such as this for the field of Computational Creativity.

Project Conception

The driving forces behind the entire project were Archie Baron, Executive Producer, and Catherine Gale, Series Producer/Director of *Computer Says Show* and *Beyond the Fence*, from Wingspan Productions. The following are first person accounts of the project conception from the perspectives of this production team:

The Idea (Archie Baron)

The musical *Beyond The Fence* is the unlikely offspring of two television documentaries we made in 2014. *The Joy of Logic* (BBC) explored the hypothesis – based on Alan Turing's notion that the brain and a computer are both essentially information processing systems governed by logical rules – that one day we could in theory program a computer to reproduce and rival all of human thought. *Our Gay Wedding – The Musical* (Channel 4) was just that: Benjamin Till and Nathan Taylor's wedding on the first day same-

sex weddings were legalised, composed by the grooms and broadcast as a joyous sung-through musical. An improbable mash-up of these two projects led us, in discussion with the British television channel Sky Arts, who were interested in what might underpin success and failure in the arts, to the basic idea for the two-part documentary series *Computer Says Show*: could we use machine learning to analyse what makes a hit musical, then team up with computers to create and stage one?

From the outset, the project was conceived as both a serious experiment and a provocative and audacious event. It was also geared as much towards documenting the process as investing in the outcome. This ‘experiment’ and the resulting documentaries and musical were above all designed to provoke a debate. What constitutes success in creating music or stories? What parts of the creative process can be automated? How do scientists, writers, composers and audiences respond to work created in collaboration with computers? What might the general public make of the leading edge of this field – in terms of the science, technology and philosophy that underpins it? This is a debate worth having. What was it Alexander Graham Bell said? “I truly believe that one day there will be a telephone in every town in America.” The future – as with so much technology – is probably nearer and more extraordinary than most of us imagine.

The Practicalities (Catherine Gale)

Bringing together the team to create *Beyond The Fence* was one of the most fascinating and fulfilling challenges of my career so far – in science or programme making. For brevity, it can be broken down into three main stages:

1. Initial research: scoping out the leading edge in the theory and practice of computer generated art – in particular music, story, dialogue/lyrics.
2. Consortium formation and pre-production: bringing together a group of academics interested in turning their attention to musical theatre, to contribute to the project such that both the musical itself and their own research projects would benefit. Significant data gathering and annotation was required, as well as collaborative discussions about methodology and goals, and documentary filming to tell the story of the ‘experiment’ as it unfolded (see figure 1).
3. Writing the show: all the data analysed (Cambridge dataset), systems developed/made available (WHIM for musicals, PropperWryter, The Cloud Lyricist, Flow Composer) or material generated (from Nick Collins) had to be used by the writers, to create the new show. These processes were also documented on film. The writers worked to a set of guidelines that ensured that computer-generated content would remain the core ‘raw material’ for the show. This enabled a story to be told around the experience of the (human) writers having to work with computer-generated material when putting the show together; the actors and creative team’s experience of working with it; and the audience’s response to it. It also meant that we could explore in detail the science behind each process, whether predictive analytics, machine listening, algorithmic composition, plot analysis and generation, or lyric generation.

An exciting feature of this project is that, having reached the finish line in terms of the musical and the documentaries, we now have a ‘case study’ in hand, which perhaps only answers a small number of questions in and of itself, but poses far more about the role that computational systems can, will and should play in the future of all of our lives – as creators and consumers of both art and technology.

The Guidelines (Archie Baron)

One of the toughest challenges for the project was framing sufficiently rigid protocols that articulated for everyone involved the basic framework for the project, while having sufficient flexibility that we could document and observe an unfolding process about whose outcomes we were extremely uncertain until very late in the day. We codified these protocols into a formal set of ground rules which we asked the writers and everyone involved in the creative team to work to in an attempt to ensure that computer-generated content would remain the core ‘raw material’ for the new show. The fact that the project had two key aims which were not necessarily compatible – to model a hit musical using the Cambridge data (see below) which West End theatregoers would buy tickets for and to maximise the computer-generated content therein – became an editorial and creative challenge (and an important narrative angle for the documentaries).

Contributing Systems and Application Details

As portrayed in figure 1, how the systems used in the experiment contributed to the musical is somewhat complicated. In overview, two writers were in overall creative control of the writing of the musical: Benjamin Till and Nathan Taylor. Software contributed to the music, lyrics and story of the musical, and also provided overall guidance to the writers, which was referred to throughout the process.

A corpus analysis was used to identify factors associated with successful and unsuccessful shows, which influenced all other aspects of the writing process. The story for the musical was then developed from an original concept produced by *The WhatIf Machine* ideation engine developed at Goldsmiths College, and output from *PropperWryter*, a plot generation system developed at Universidad Complutense, both of which are described below. Data input to *PropperWryter* included an annotated dataset of stories, created specifically for the project. The writing of the musical score was informed by a music information retrieval (MIR) analysis of the musicological features of successful and less successful musicals, produced via a study at City University London and Queen Mary University of London, described below.

The analysis informed the first of two algorithmic composition techniques, a corpus based generative system developed at Durham University, described below. A second generative system called *Flow Composer*, from the Sony Computer Science Laboratory, was also used, producing new music in an interactive way, via style modelling, as described below. Finally, also described below, new software called *The Cloud Lyricist* was developed which used a neural network approach trained on musical theatre lyrics to generate text segments which were carefully selected by the writers. The music, lyrics and story inputs from the computational

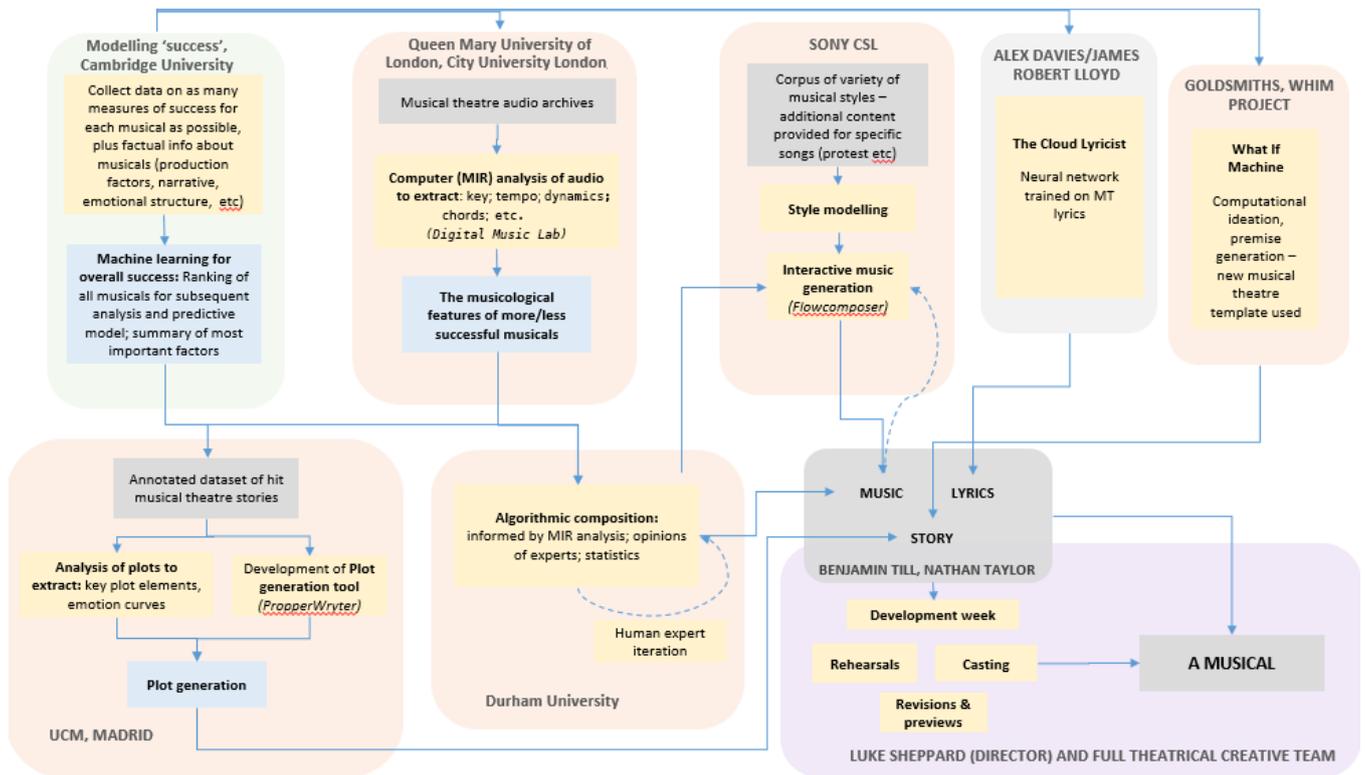


Figure 1: Research teams and their contributions to the project.

systems were combined by Till and Taylor into the final musical, guided by the machine learned analysis, then shared with the director, Luke Sheppard, and team, for subsequent development, rehearsals and staging.

The Statistics for Success in Musical Theatre

The first stage was the study ‘Musical Theatre by Numbers’ combining descriptive statistics, machine learning and predictive analytics conducted at Cambridge University by James Robert Lloyd, Alex Davies and David Spiegelhalter. They built a corpus of information about 1696 musicals, including 946 synopses and in-depth surveys annotating individual shows. All musicals were measured for both commercial and critical success based on length of run and awards won, and then classified into one of 4 categories (hits: commercial and critical success; crowd pleasers: commercial success only; critically acclaimed: awards but shorter runs; flops: neither commercial nor critical success). The aim was to discover the factors that are associated with (and might be predictive of) success, via exploratory data analysis using logistic regression and decision trees.

The relative importance of high level features such as cast size, the gender of the lead, musical styles, geographic and temporal settings, star power, a happy ending, the incidence of comedy, death, spectacle, dance within a show, were all investigated. A thematic study of shows was also achieved through synopsis text analysis. The emotional journey (or story arc) throughout a selection of 52 representative shows was also annotated and analysed. Volunteers listened to soundtrack recordings of complete musicals they knew well, recording how they felt about each song in ten different emo-

tional classes (to attempt to capture the emotional trajectory of musicals in terms of energy and vitality; love and tenderness; tension and sadness; comedy). Clear differences survived the averaging of these emotional arcs to distinguish hits from flops, and these became key features defining the target emotional structure of what became *Beyond the Fence*. The results of this analysis were presented to the writers as a presentation (with an example graphic from the presentation given in figure 2) focusing on the key decisions that might increase the chances of writing a hit.

The WhatIf Machine

In the creative arts and industries, the production of fictional ideas around which to write stories, paint pictures or design advertisements, is an essential activity. Motivated by this, in the European WHIM project (www.whim-project.eu), via the building of The WhatIf Machine, we are undertaking the first large-scale study of how software can invent, evaluate and express fictional ideas of real cultural value.

We refer to fictional ideas as modifications of knowledge where the perceptions we hold about existing concepts of the world are altered and new representations are produced. The Goldsmiths approach in the WHIM project largely consists of applying controlled alterations and combinations of facts from a knowledge base (KB), represented as triples which relate two concepts. We have developed a set of ideation methods which have been reported in (Llano et al. 2016). For instance, given the natural language template: $t = \text{What if a } X \text{ learned how to } Y?$, the simple ideation method:

$$im(t) = \{r \mid \langle c_1, \text{NotCapableOf}, c_2 \rangle \in KB \quad (1)$$

$$\wedge r = \text{instantiate}(t, [X = c_1, Y = c_2])\}$$

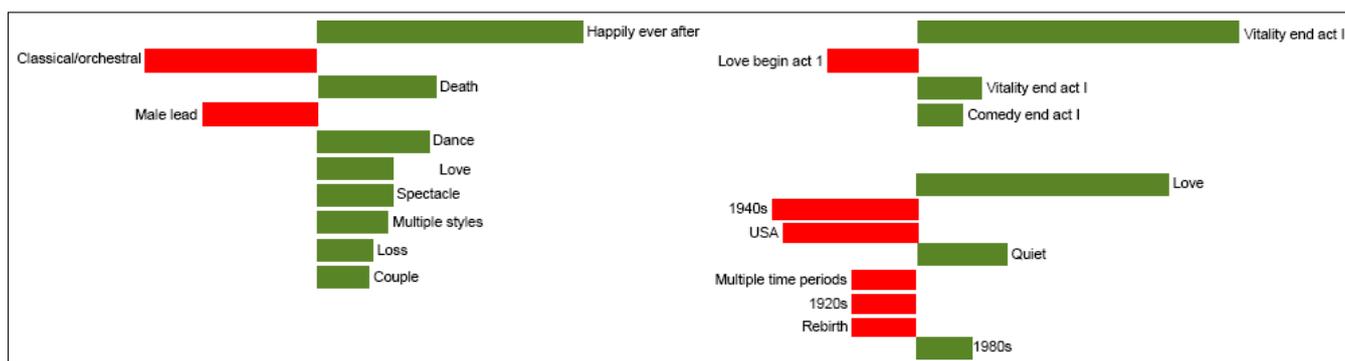


Figure 2: Statistical analysis of (green) hits and (red) flops, with bar length indicating the propensity of the factor in the respective category.

produces a fictional idea using t , for facts in the KB that focus on the *NotCapableOf* relation. For instance, given the fact $\langle \text{dog}, \text{NotCapableOf}, \text{ride_horse} \rangle$, (1) yields the fictional idea: *What if a dog learned how to ride a horse?* A set of conditions and heuristics are applied to narrow down the input knowledge and to perform more complex combinations of facts. Moreover, the generated ideas are evaluated against a set of narrative measures, which are then compared with an audience model that returns the list ranked according to a machine learned predictor of value (details omitted).

The Cambridge thematic analysis identified four general themes which were passed on to the WHIM team: *journey*, *aspiration*, *love* and *lost king*, and for each theme a set of keywords were also identified. Additionally, the writers summarised a set of musical synopses, from hits, crowd pleasers, critically acclaimed and flops, into what-if style sentences, which acted as targets. Finally, we were provided with a description of Bookers' seven basic plots (Booker 2004) as standard plot types followed in musical theatre. Because of the short time span to complete the experiment, we focused on three of these plot types, namely: *rags to riches*, *quest* and *rebirth*. Note that, while – as per figure 2 – rebirth was identified as a negative factor, it was included due to handover timing issues. We combined the plot types with the themes mentioned above in order to produce different types of fictional ideas in the musical context. All this data was used to: (i) retrieve further input knowledge from different linguistic resources (using the theme keywords), (ii) create template what-if premises (using the musical premises), and (iii) select the components required by the premises to be generated by the system (using the plot types).

To illustrate, following the description of the *Rebirth* plot type, which states: *'The protagonist is a villain or otherwise unlikable character who redeems him/herself over the course of the story'*, and the theme *love*, we knew that the character of the statement should carry out a transformative action in order to gain someone's affections. The transformative action was selected as a combination of data obtained from the stereotypical properties of selected characters and the further knowledge obtained through the keywords. Figure 3 shows a screenshot of the musicals area of the web interface to The WhatIf Machine, which shows some sam-

ple *Rebirth/Love* outputs.

The Wingspan Productions team used The WhatIf Machine to choose and print out 600 ideas suitable as the overall concept for the musical. The choosing of candidates from this set was filmed on a stage at Goldsmiths, where all 600 were laid out. Some ideas considered were not practical, such as *"What if there was a poor boy who was born with a horn, which made him good at communicating, so he went on to become a famous slave"*. However, the writers were able to take away a shortlist of ideas for further consideration. Guided by the Cambridge analysis that the musical should be set in a conflict of the 1980s, not in America, and involve a female protagonist, they finally settled on the idea: *"What if there was a wounded soldier who had to learn to understand a child in order to find true love?"* Elements from this phrase, plus other constraints (e.g. death, loss) were typed into a search engine and one of the top hits was a songbook from the Greenham Common Womens' anti-nuclear peace camp (www.yourgreenham.co.uk). Ultimately, this led to the writers' interpretation of the idea, whereby a male soldier who has been wounded and posted to Greenham Common nuclear base in the UK befriends the mute child of a female protester, which ultimately leads to him and the protester falling in love.

PropperWryter

PropperWryter is a program that generates the narrative structure for a single plot line, described in terms of a vocabulary of abstract representations of events that may happen in such a plot. It evolved from the Propper system (Gervás 2015), which generated plot structures for Russian folk tales based on Propp's (1968) *Morphology of the Folktale*.

Propp identified a set of regularities across a corpus of Russian folk tales in terms of *character functions*, understood as acts of the character, defined from the point of view of their significance for the course of the action. To extend this approach to the generation of plots for musicals, the vocabulary was adapted and extended to cover the range of acts of characters typically involved in musicals, and data had to be collected on how elements from this customised vocabulary appeared in existing musicals, and on how these elements interacted with one another. The new vocabulary for

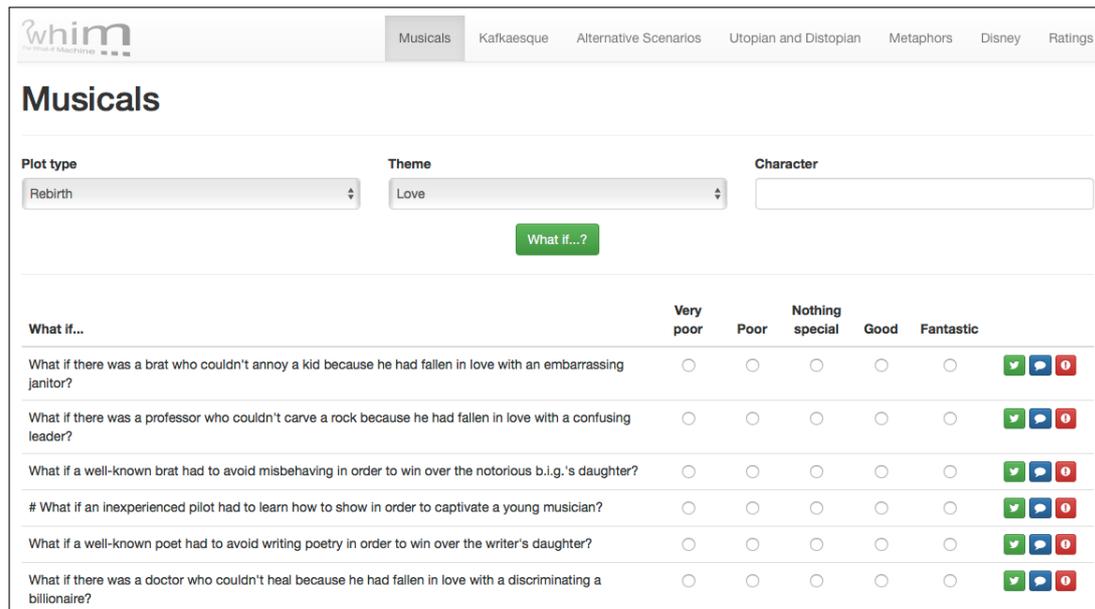


Figure 3: The web interface for The WhatIf Machine.

describing musical theatre plot was built as a new specific set of abstractions in the spirit of character functions – re-named as *plot elements* to avoid confusion – and constructed from a number of sources: the original set of Propp’s character functions; additional abstractions mined from alternative sources in the literature on narrative (Gervás, León, and Méndez 2015); and abstractions specific to the description of the plot of musical theatre. Knowledge resources on how this vocabulary is used in existing musicals were compiled from the results of an annotation effort of musical plots carried out by 35 volunteers at the University of Surrey, supervised by Julian Woolford, Programme Leader for the MA in Musical Theatre, and the Wingspan Productions team.

The ProperWryter system relied on these knowledge resources to construct sequences of plot elements – from the new vocabulary – that describe a plot line. On reviewing earlier versions of the prototype, which allowed a user to provide an initial brief to drive the plot construction process, the writers requested that this functionality be disabled, to avoid any risk of any features in the output being attributed to the input rather than the machine’s efforts. During the writing phase, the writers used ProperWryter to generate a number of possible plots and selected one that started with an Aspiration plot element. The plot structure of *Beyond the Fence* is, therefore, entirely underpinned by a ten-point arc generated by the software.

Music Analysis with the Digital Music Lab

To study the acoustic and musical characteristics of musicals from the four categories established by the Cambridge team, we (Bob L. Sturm, Tillman Weyde and Daniel Wolff) employed the Digital Music Lab (DML) infrastructure – the outcome of a recent project developed for large-scale music analysis in a collaboration of City University London, Queen Mary University of London, UCL and the British

Library (Weyde et al. 2014). The musicals project encapsulates an excellent use case from the DML: we needed to study a massive amount of music material from a variety of different perspectives in a short amount of time. We extracted low-, mid- and high-level features from our corpus of 77 full-length commercial recordings of musicals – over 130 hours of sound recording of diverse styles and from different historic periods – and analysed dimensions such as loudness, brightness, tempo, dynamics, key, and harmonies.

Our analysis delivered expected results: we find a predominant use of major keys across all categories, and we find a lack of strong prediction power of these features (which for the most part are far from describing the experience of music) for the four categories. Averaging the features across musicals in each of two categories (flop (N=23) or non-flop (N=54)) – taking care to normalise the time scale of the musicals – showed some intriguing patterns. Figure 4 shows the change in dynamics (loudness) from the mean of a track – essentially (de)crescendi in a track. We see a tendency in the final 25% music of flops for tracks to become increasingly more dramatic. We also see that the final song in flops tend to be faster than average. At a higher level, we found that flops in our corpus tend to have more harmonic progressions containing the subdominant.

Our results must be handled with caution for at least two reasons: 1) we do not demonstrate any causation, e.g., that the use of the subdominant affects the success of a musical; and 2) the flops in our corpus are the “best kind”, i.e., successful enough to be commercially released as a recording. The computational analysis of music remains far away from human listening experience and expertise. Nonetheless, our involvement in this project shows the success of the DML infrastructure for studying the questions posed by the production team, as well as providing high-level musical features for the modeling and generation of music.

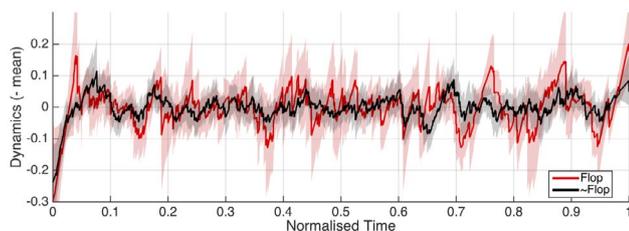


Figure 4: The tempo relative to track average over normalised duration for flop and non-flops classes.

Claude-Machine Schönbot

The music generating program Claude-Machine Schönbot (formerly known as 'Android Lloyd Webber') was used to produce the majority of the computer generated music for *Beyond the Fence*. It is based on a corpus of chord changes from hit musicals, and an additional corpus of transcribed leadsheets (main melody and accompanying chords) from hit musicals. Which musicals were denoted hits, and the chord changes extracted from cast recording audio files (53 musicals, 1124 audio files, around 53 hours of audio), arose from other groups involved in the project (see above). Markovian machine learning (of variable order, via prediction by partial match (Pearce and Wiggins 2004) is deployed, alongside hard-coded rules informed by the statistics of the melodic examples. As all corpus-based algorithmic composition makes representational assumptions on the nature of music data analysed and generated, and given the pragmatics of theatre shows with hard deadlines, the involvement of explicit rules – which were developed via feedback from the writers – was seen as acceptable. The program could generate melody and chord leadsheets as pure music, or with rhythmic phrases constrained by lyrics provided, following a basic metrical stress analysis of scansion.

Flow Composer

Flow Composer is an online lead sheet editor with a composition module relying on style-imitation: the system builds a *statistical model* from a corpus of lead sheets and generates new musical material in the corpus style (Pachet and Roy 2014). Composing a lead sheet is achieved in a progressive way, involving several rounds of user-system interactions. The user starts with an empty or partial lead sheet (see figure 5). First, Flow Composer is repeatedly used to generate music for the selected zone until something interesting comes up. The user deselects the part they want to keep and uses the generator on the remaining selected zone. At all times, the user may freely modify the current lead sheet and use the history to go back to a previous state. The music can be listened to using an audio engine producing natural renderings in many musical styles. Flow Composer samples random sequences subject to constraints from the statistical model (Papadopoulos et al. 2015). The musical elements to generate (selected zone) are treated as random variables and constraints represent the notes and chords that the user wants to keep unchanged (non-selected zones).

Flow Composer was initially designed for jazz, Brazilian, and pop music generation. It is coupled with a large database

containing 13,000 songs in these styles (Pachet, Suzda, and Martinez 2013). The writers thought it was appropriate to train the system with songs composed by the women of Greenham Common (a songbook is available at yourgreenham.co.uk). They chose six songs that were added to the database. With such a small corpus (75 bars in total), the system often fails to generate music fitting the user's input, leading to frustrating interactions. We changed the training mechanism so that the system learns the user's input, i.e., as the user enters new music, or modifies the system's output, the corpus is enriched and the model is retrained, reducing failures. We also added the ability to add transposed copies of original songs in the corpus, to create more diversity.

The writers expressed the need to control some musical parameters of the generated sequences. The sampling mechanism guarantees that the generated music has the same statistical properties as the training corpus, but provides no way to control the generation. We added controls for five musical parameters: harmonic tightness, average melodic interval, average note duration, proportion of rests, and frequency of chord changes. Each parameter is controlled using a slider widget on the graphical user interface (see top of figure 5). These criteria are taken into account by a simple generate-and-test mechanism: several sequences are sampled, and the one that best fits the criteria is returned. The speed of the sampling mechanism makes it possible to generate thousands of sequences in a few seconds.

The writers used Flow Composer for three songs: *Scratch that Itch*, *We Are Greenham* and *Unbreakable*, modifying the raw output, for instance to fit the rhythm to the lyrics or to create an ending to the melody, which conforms to the type of iterative interaction Flow Composer was designed for. This also suggests interesting directions to improve the system in the future. Flow Composer is currently being used for the composition of a pop music album.

Clarissa, The Cloud Lyricist

Once the data and software-guided basic structure, setting, key features, central idea and plot arc were in place, the writers proceeded to prepare the spoken dialogue for the show. Alex Davies and James Robert Lloyd, who had already had some success with computer generated poetry, trained a recurrent neural network language model (based on Andrej Karpathy's Char-RNN: karpathy.github.io/2015/05/21/rnn-effectiveness/) on a corpus of some 7,000 musical theatre lyrics (after first pre-training on a portion of Wikipedia and a corpus of 10,000 poems) and built The Cloud Lyricist (or Clarissa as 'she' became known in the rehearsal room). Clarissa supplied an initial 1,000,000 characters of lyrics for the writers and computer lyric dramaturge Kat Mace to draw from and incorporate into the songs. They later developed the system with a web interface with a 'creativity factor' setting which could be set by the user to increase the abstraction level (from 0 to 1) in the language produced (using the temperature parameter in Char-RNN). The setting of 0.6 or 0.7 proved to be the most useful.

The lyric writing process required time-consuming trawls through Clarissa's limitless oeuvre, usually to find brief (usually part line or single line) stretches of usable lyrics

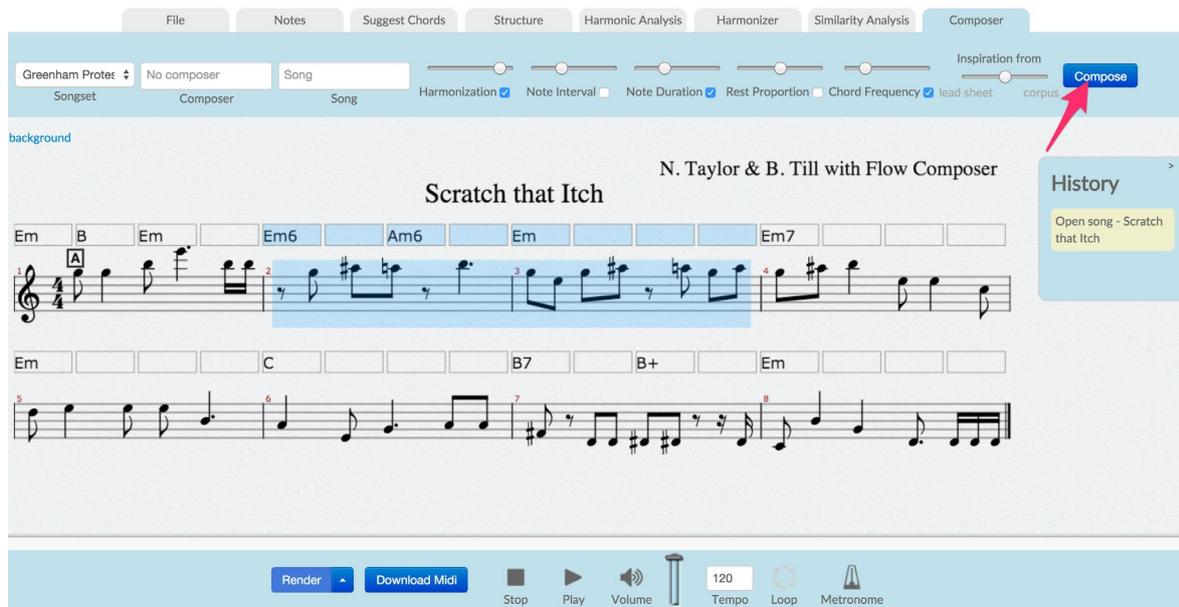


Figure 5: The web interface for Flow Composer. The Compose button generates music for the selected (highlighted) part.

which could be incorporated with other computer lines and/or lines from the writers. In total, almost a quarter of the lyrics in *Beyond the Fence* were computer generated (ranging across the 16 numbers from 6% in the song *Graceful* to 32% in *So Much to Say*). Close textual analysis of the libretto would be interesting: with the computer lyrics perhaps standing out as the more ‘original’, metaphorical, daring or unusual. To assist further, the Cambridge team also completed a statistical analysis of lyric word frequencies across different musical theatre song types designated by the team (Angry, I Am, I Want, Love, Comedy, Comfort, Duets, Protest) and gave the writers word clouds to access during curation and writing of the lyrics. The song *We are Greenham* was entirely constructed from lyric word clouds made from the real protest songs sung by the original Greenham Common peace protesters and another word cloud of lyrics used in protest songs in musical theatre.

Impact

Beyond the Fence had 15 performances between 22nd February and March 3rd 2016. 3 performances were previews before the official opening, 2 were matinees, and 1 was a gala event. 3,047 people saw the musical, representing 59% of the total capacity across the run, which is impressive, given that there was almost no significant advertising or paid marketing, unlike for most musicals of this scale. For the last three performances, the audiences were polled and asked to rate their enjoyment of the show from 1 (low) to 5 (high). Of the 57 respondents, the poll revealed an overwhelmingly high level of enjoyment, with rank/percentages as follows: 1/1.7%, 2/1.7%, 3/10.3%, 4/17.3% and 5/69.0%.

Press engagement was managed in two phases: first with the announcement of the project/on-sale for tickets (December 2015); second in the run-up to the opening of the show and transmission of the documentaries (February 2016). *Be-*

yond the Fence and *Computer Says Show* received extensive domestic and international coverage: in print and online (including but not limited to The Guardian, The Times, The Evening Standard, The Independent, The Telegraph, BBC News Online, Financial Times, Daily Mail, New Scientist and Vice); and on television and radio (BBC Breakfast News, BBC Radio 4-World at One, CBC Radio-Canada, Sky News, Reuters TV, NHK-Japan, CBS News-US and the Guardian Tech Podcast). Interviews were conducted with the television and theatre production teams, the two writers, and some of the academics involved.

Reports from stage management document consistently engaged and reactive audiences, and reactions on social media have also been very positive. Audience members tend to express surprise that they are moved – often to tears – by parts of the show (with some expressing surprise that computers could have been involved in triggering their emotions, others insisting that the emotional sections must have been down to the human inputs, and some feeling ‘more manipulated’ than they would otherwise because they know computers were involved in playing with their emotions).

Thanks to footfall in the West End, and the frequency of walk-in trade, on several occasions people have attended the show not actually knowing about its computational genesis. On one such an occasion a gentleman remained for a post-show Q&A ‘with the scientists’, and raised his hand to tell everyone he “... had no idea about all of this, and I’m just amazed – I thought it was brilliant, so well done”. Another audience member, on the same evening added, in relation to a discussion around the challenges that Till and Taylor faced “I thought it was brilliant, and if you’ve done this when it was all so hard, I can only imagine that the next one – when the computers get better – is going to be even better!” Various images depicting *Beyond the Fence* and *Computer Says Show* are given in figure 6.

The following is an extract from an interview conducted with the two writers of the musical, which appeared in the programme for the show:

Writers' Perspective (Benjamin Till and Nathan Taylor)

Collaborating with computers is utterly unlike anything either of us have encountered before, and at times, it has been incredibly frustrating. The systems for ideation, and for sparking the inspiration of creativity in the human mind are brilliantly helpful, and often lead to ideas that we would never have come up with on our own. It gets trickier, however, when you get into the realms of generating actual material. In a musical, nothing is left to chance – every note, every word, every idea, needs to support and inform not only the overall story, but also the characters' individual journeys, and a million other things that all need to feed into each other. With these computer systems, nothing is bespoke. We waded through probably a thousand pages of computer-generated tunes to find the fragments and phrases that felt right for the show's needs, and were suitable for developing into the songs that have become *Beyond the Fence*.

Once we had found the material we wanted to develop, it didn't make any difference that some of it had come from us, and some from the computers. It all went through the same process of refinement and evolution as any other songs we've ever written. As a result of that refinement and development, we both feel every bit as much ownership of the piece as if we had written it all ourselves. It would be hard not to, as we have had to immerse ourselves in it, and put as much of ourselves into it as with anything else.

Beyond the Fence really is a world's first, and a piece of history in the making – it is a true collaboration between humans and computers, bringing together human endeavour and the best that technology can currently offer in the field of Computational Creativity. These computer processes are still very much in their infancy, and we feel privileged to be the vanguard of this kind of work. In a few years, who knows: you might be able to push a button, and out pops *My Fair Lady*, but somehow, I doubt it. I rather think that the future holds ways of allowing human artists to work with computers more comfortably, and with more control of their output, ultimately to support and perhaps shape their own creativity in ways they might not have been able to envisage.

Critical Reviews

The critics were divided in terms of how they evaluated *Beyond the Fence* – some as they would any new musical; some focusing on the computer-generated nature of the material. There seems to have been a desire to pick apart the content in the show, in order to evaluate the 'more human' and 'more computer'-generated parts separately. The following *Broadway Baby* review highlights this tendency well:

"Their solos (everyone has a 'tick-box' solo) are the absolute highlights – particularly Matthewson who manages to create pathos through the roller-skating based song *Graceful* and gets what seems to be the most genuine applause of the night with a performance that is resonant of Victoria Wood's play on word delivery with Julie Walter's comic timing. It's interesting to note that *Graceful* is the song that

had the fewest lyrics delivered by the computers (only 6%)."

Interestingly, the song *Graceful* was in fact inspired by one initial computer-generated line ('Now let me be fat'), from which a full lyric was written. Music was then algorithmically composed to the lyrics, lending the song much of its unusual rhythm, developed by the writers to become a song that brings the house down every night. This is just one example, but it indicates the challenge for reviewers in formulating a response, when attending to the material with both human and computer 'agency' in mind. This was expressed in more general terms in *The Independent*:

"... I wonder if the computer-generated tag will help or hinder: it's hard to think you'd watch the show without being more interested in the process than the product. And am I being romantic in thinking it's telling that while the story and songs work fine, the thing that makes it zing is the human-chosen setting?"

The critical response has been extensive and represents an unprecedented volume of expert reaction to, and opinion on, work in this field. The following are a selection of quotes from some of the reviews *Beyond the Fence* received:

*** "A unique experiment in musical theatre composition"
The Stage

*** "Despite my reservations I was won over"
The Independent

** "Hokey, but effective" *The Times*

** "As a theatrical event the show is remarkable"
West End Frame

*** "Wins you over with the weight of its clichés"
Time Out

*** "What's our measure for success? Well, the audience was applauding just about every number and was brought to tears" *The Londonist*

**** "Extremely moving and emotional ... it could be one of the most important pieces of theatre to come out of London this year" *West End Wilma*

*** "It may not herald a brave new world, but it does work as a night out ... in a world where flops are the norm, no mean feat" *Daily Telegraph*

Unsurprisingly, the more harsh reviews focused on the formulaic nature of the musical imposed through the machine learning exercises. These included the following:

** "Guess what? When you get a computer to create a musical, as Sky Arts has done – using data from the structure, scores and scripts of hundreds of musicals to generate scenarios, melodies and lyrics – it sounds just like a musical composed by a computer. This show is as bland, inoffensive, and pleasant as a warm milky drink." *Guardian*

** "Isn't it obvious? If you take the average of a load of hit musicals, you'll end up with something pretty average. Follow a formula and – who would have thought it – you get something formulaic." *What's On Stage*

Conclusions

This paper acts primarily as a record of the project which led to the *Beyond the Fence* musical and *Computer Says Show* documentaries. This project stands as perhaps the largest ever application – in the sense of breadth of software employed and the scale of achievement – of generative software for cultural benefit, and an important experiment in the usage of Computational Creativity research prototypes by non-experts. While the number of systems employed and the scale of the output that the software influenced – an entire West End musical theatre production – were impressive, the software was sometimes a hindrance rather than a benefit to the writers. In most instances, the software acted as more of a muse and/or creativity support tool to the two writers of the musical rather than as the primary author. Computational Creativity exists as a research field to help bring about a future where creative software positively affecting our lives is as commonplace as the benefits of telecommunications systems or social networking technologies. This is not going to happen through scientific experimentation alone: the challenge of building and utilising creative software needs to be taken up by larger portions of society, including technology industries, creative industries and the arts.

Due to the recent completion of the project, it has not yet been possible to fully scientifically evaluate questions such as the level of creative contribution each piece of software made, how easy/difficult the writers found using software to guide their creative process rather than merely enabling it, or whether the musical was a popular and/or critical success. All these questions merit further research. However, from a cultural point of view, the project has clearly been successful. This can be measured in terms of: the increase in diversity of audiences for our research outputs, to include musical theatre lovers; the sheer number of people exposed to ideas from Computational Creativity research through the press and television news reviews and the Sky Arts documentaries; and the fact that there has, to the best of our knowledge, been no societal backlash against the idea of software contributing creatively to cultural projects. Such projects as the one described here help to bring about acceptance of the notion of software being our creative partners, adding to cultural life in useful and meaningful ways, and ultimately bringing many benefits across society.

Acknowledgements

The contribution of The WhatIf Machine to this project has been supported by EC grant WHIM (FP7 grant 611560), and an EPSRC Leadership Fellowship grant (EP/J004049). We would like to thank Catherine Bellamy for excellent work in supporting the link between WHIM project members and the Wingspan Productions team. We would also like to thank the organisers of the PROSECCO network (FP7 grant 600653) and particularly the organisers of the 2015 *Show, Tell, Imagine* event at Queen Mary University of London, where the writers of *Beyond the Fence* and the Wingspan Productions team met to discuss potential collaboration. Flow Machines is funded by the European Research Council (ERC) under the European Union 7th Framework Programme (FP7/2007-2013), ERC Grant Agreement number 291156. We thank the

anonymous reviewers for their very helpful comments.

Computer Says Show (Gale, Baron, and Lomax 2016) and *Beyond the Fence* (Till, Taylor et al. 2016) were commissioned by Sky Arts, who were the majority funders (Director Phil Edgar-Jones and Commissioning Editor Siobhan Mulholland). It was developed with the assistance of the Wellcome Trust. It was produced by Wingspan Productions. The Head of Production for the project was Lil Cranfield. *Beyond the Fence* was first performed at the Arts Theatre London on 22nd February 2016. The General Manager was Neil Laidlaw Productions.

References

- Booker, C. 2004. *The Seven Basic Plots: Why we tell stories*. Continuum.
- Gervás, P.; León, C.; and Méndez, G. 2015. Schemas for Narrative Generation Mined from Existing Descriptions of Plot. In *Proc. 6th Workshop on Computational Models of Narrative (CMN 2015)*, volume 45 of *OpenAccess Series in Informatics (OASISs)*.
- Gervás, P. 2015. Computational drafting of plot structures for Russian folk tales. *Cog. Computation* 8(2), 187-203.
- Gale, C, V.; Baron, A.; and Lomax, K. 2016 *Computer Says Show (episodes 1 and 2)*. Sky Arts www.wingspanproductions.co.uk
- Llano, M. T.; Colton, S.; Hepworth, R.; and Gow, J. 2016. Automated fictional ideation via knowledge base manipulation. *Cognitive Computation* 8(2), 153-174.
- Pachet, F., and Roy, P. 2014. Imitative Leadsheet Generation with User Constraints. In *European Conference on Artificial Intelligence, ECAI*, 1077–1078.
- Pachet, F.; Suzda, J.; and Martinez, D. 2013. A Comprehensive Online Database of Machine-Readable Lead-Sheets for Jazz Standards. In *Proc. Int. Soc. for Music Inf. Retrieval Conference*.
- Papadopoulos, A.; Pachet, F.; Sakellariou, J.; and Roy, P. 2015. Exact Sampling for Regular and Markov Constraints with Belief Propagation. In *Proc. Principles and Practice of Constraint Programming*, 341–350.
- Pearce, M., and Wiggins, G. 2004. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research* 33(4):367–385.
- Propp, V. 1968. *Morphology of the Folktale*. University of Texas Press.
- Till, B.; Taylor, N.; The WhatIf Machine; Propper-Wryter; Claude-Machine Shönbot; Flow Composer; The Cloud Lyricist; Mace, K. *Beyond the Fence*. Sky Arts and Wingspan Theatricals. Theatre director Luke Sheppard; Television broadcast director Tim van Someren. www.wingspanproductions.co.uk
- Weyde, T.; Cottrell, S.; Dykes, J.; Benetos, E.; Wolff, D.; Tidhar, D.; Gold, N.; Abdallah, S.; Plumbley, M. D.; Dixon, S.; Barthelet, M.; Mahey, M.; Tovell, A.; and Alancar-Brayner, A. 2014. Big data for musicology. In *Proc. of the 1st Int. Work. on Digital Libraries for Musicology*, ACM.

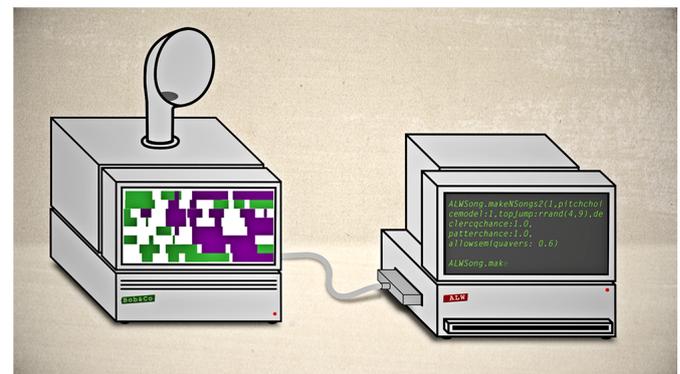
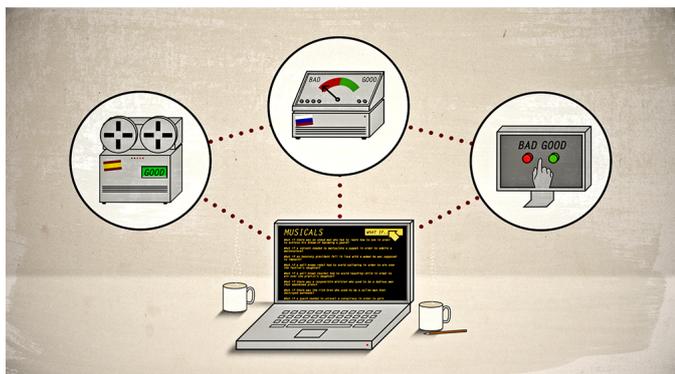
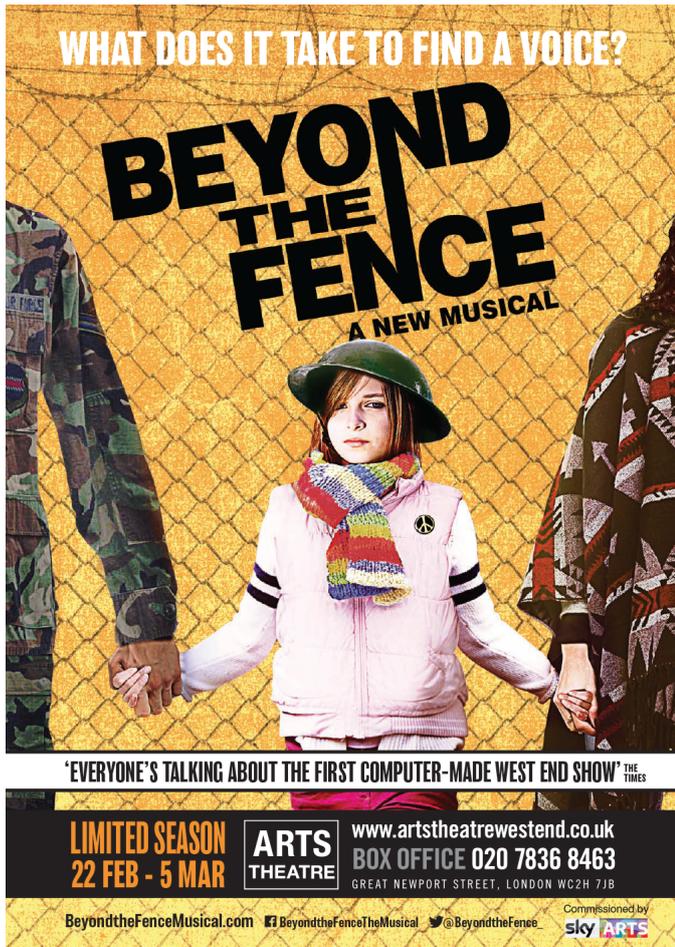


Figure 6: (a) The poster for the show (b) the creative billing, where the software systems take their rightful place among the team that developed the musical (c) images from the musical *Beyond the Fence* (credit: Robert Workman), and (d) graphics used in the *Computer Says Show* documentary (credit: Andy A'Court).

BLENDING



Free Jazz in the Land of Algebraic Improvisation

Claudia Elena Chiriță, José Luiz Fiadeiro

Dept. of Computer Science, Royal Holloway University of London, UK
claudia.elena.chirita@gmail.com, jose.fiadeiro@rhul.ac.uk

Abstract

We discuss the connection between free-jazz music and service-oriented computing, and advance a method for formal, algebraic analysis of improvised performances; we aim for a better understanding of both the creative process of music improvising and the complexity of service-oriented systems. We formalize free-jazz performances as complex dynamic systems of services, building on the idea that an improvisation can be seen as a collection of music phase spaces that organise themselves through concept blending, and emerge as the performed music. We first define music phase spaces as specifications written over a class of logics that satisfy a set of requirements that make them suitable for dealing with improvisations. Based on these specifications we then formalize free-jazz performances as service applications that evolve by requiring other music fragments to be added as service modules to the improvisation. Finally, we present a logic for specifying free jazz based on one of Anthony Braxton's graphic notations for composition notes.

Introduction

Complex dynamical systems, more than being simply complicated, are systems whose structure is intrinsically unpredictable, as they are open to redesign. Their emergent behaviour is dictated by the interconnections and the interactions of cooperation and competition between their entities. These systems are based on non-mereological composition, and thus should be seen as more than the sum of their parts. We argue that both free-jazz performances and service-oriented systems exhibit the characteristics enumerated above and we thus adhere to the tenet expressed in works such as (Borgo and Goguen, 2005), (Borgo, 2005), and (Blackwell and Young, 2004), stating that free-jazz performances are instances of complex systems.

What is free jazz? As it happens more often than not with complex systems and creative processes, free jazz usually gets its diagnosis *per exclusionem*, as it is easier to be understood as the sum of things it is not, rather than of the things it is. We adopt the stance of Mazzola and Cherlin (2008), who propose a positive characterisation of free jazz, instead of the usual negative definition: free jazz is the form of jazz in which the performers are the only ones held accountable for the music that is being played, since (generally) no (standard) notations are followed. The music results from a dynamic, complex game that changes its rules throughout the

performance. The success of the game is determined by the identity that emerges from both coherence and conflict – the emergent “dynamical orderings” of the music “that are both surprising and comprehensible” (Borgo and Goguen, 2005). As highlighted in (Borgo, 2005), free jazz is by no means random or lacking rules, even if the evolution of an improvising act is a priori unpredictable due to its transforming constraints and rules: the standards of quality are high, albeit different from the ones of traditional music. Free-jazz improvising is not typically pursuing the classical rhythms, harmonies, or melodies; its valuable aspects are rather the pervading creativity, the discovery of new musical dimensions, the emergence of a collective purpose, or the unexpected synchronizations that interrupt divergence moments. These features seem to challenge (or disregard at least) the existing means of analysing and evaluating conventionally notated music. This is why we believe new tools are required for studying the phenomenon of self-organising music.

Computational models for free jazz Even if we do not intend to address to the utmost the complexity of musical improvisations, we focus on the dynamics of these performances by means of formal, algebraic methods for complex systems. What differentiates our paper from other previous work is the way we approach complexity. We formalize free-jazz performances as complex dynamic systems of services, building on the idea that an improvisation can be seen as a collection of music phase spaces that organise themselves through concept blending, and emerge as the performed music. The music improvised up to a point plays the role of a service application, while all the music fragments that could continue the performance are seen as external service providers; these offer their intrinsic characteristics as services meant to satisfy the ‘needs’ of the ongoing music act. In (Borgo and Goguen, 2005) free-jazz performances are modelled as non-linear dynamical systems of equations, and in (Blackwell and Young, 2004) as swarm optimization processes. It is worth mentioning that even if optimality is hardly a global feature of free jazz, as improvisation is almost never concerned with meeting objectives in an optimal way, all these computational approaches exploit optimization techniques to model a step in the evolution of the system. Borgo (2005) expresses the “sink or swim” character of improvisations as the “sync or swarm” behaviour of a complex system: either we select the states with the best fitness,

or we decide to let a process end. The swarm formalization aims to echo the emergence of a collective direction of the music performance as a whole, despite the seemingly divergence of its individual components. Although we choose to depart from the randomness inherent to swarm optimization, our approach could be aligned to the *Live Algorithms for Music* manifesto of Blackwell and Young (2004) if, in the modular view of these complex systems, we replaced the “Swarm” component with the service-oriented framework that we propose. Another notable difference between this swarm approach and the model we advance is the fact that the object of our optimisation is intrinsic to the music: since the constraints are not extraneous, no input (to which the music that we are creating should be compared) is needed.

The study of Ramalho and Ganascia (1994), proof of the long-standing interest in modelling and simulating the musical creativity of improvisations, states similar claims to our assumptions on knowledge and reasoning in jazz performances, such as the fact that musical actions depend on contexts that evolve over time, and that musicians integrate rules and constraints into their actions dynamically. Moreover, it sets similar goals with respect to simulating creativity: obtaining a suitable trade-off between the ‘flexibility and randomness’ and the ‘control and clear semantics’ in modelling creativity in terms of classical problem solving.

Modelling aims The primary goal of the model that we propose for free-jazz improvisations falls, according to the taxonomy of motivations for the formalization and automation of music compositional processes defined by Pearce, Meredith, and Wiggins (2002), in the domain of computational modelling of music cognition and musical creativity – see (Wiggins et al., 2009). This includes studies having both cognitive motivations and musicological goals that are not focused on generating aesthetically appealing music or obtaining useful compositional tools, but are rather interested in the degree in which a model serves the comprehension of the cognitive processes within composition and improvisation. Although our main aim is not to propose and evaluate hypotheses on stylistic properties of jazz compositions, on a secondary level, our study lies at the intersection of computational modelling of musical styles and design of compositional tools: the framework could be implemented and thus used to create new computationally creative music systems.

Free-jazz semiotics: music phase spaces We follow the lines of (Borgo and Goguen, 2005) and regard improvisation processes as self-organisational systems of musical *phase spaces*. We consider that the continuous flow of a free improvisation can be segmented into musical sections (regions of a phase space) that capture a distinct musical feature or that have a certain level of cohesion – a prominent qualitative character (related to the rhythm, tempo, timbre etc.). The passage between these sections plays the important role of a bifurcation in the evolution of the modelled improvisation, and will be referred to as a *phase transition*.

The phase space of a system is understood in (Borgo and Goguen, 2005) as a multi-dimensional map which facilitates the description of the dynamics of the given system. The number of dimensions is given by the number of musical variables. The standard music notation captures, for

example, a small number of dimensions: time, pitches, and other marks regarding the tempo or other details. One might think that since free jazz does not commit to notations, and since it claims to be more flexible about tonalities and timbre, we would have to deal with phase spaces having large dimensions. What Borgo and Goguen (2005) propose is actually a reduction of the unnecessary variables. We abstract over this representation of the phase spaces, and consider them simply as “musical idea spaces”, or semiotic spaces. We loosen the algebraic formalisation of semiotic spaces of Goguen (1999) by considering that a musical phase space could be described in principle through the use of algebraic specifications such as sets of sentences over a given logic – see (Sannella and Tarlecki, 2012). Thus, one should think of a music phase space as a collection of music fragments that share certain salient features and could be played at a certain moment in the evolution of the improvisation. Examples of phase spaces, together with a formal definition of the concept, are discussed in the next section.

Service binding as concept blending The notion of concept blending as used by Goguen (1999) plays a key role in defining the composition of musical phase spaces, which in turn determines the outcome of the improvisation process. Similarly to the studies of Eppe et al. (2015) and Kaliakatsos-Papakostas et al. (2014) on the role of conceptual blending in computational invention of cadences and chord progressions in jazz, we model the composition of musical phase spaces as categorical colimits of algebraic specifications. The context in which we consider such colimits is that of service-oriented computing – a paradigm that supports the development of complex software applications based on dynamic reconfigurations of networks of systems (Fiadeiro, 2012). These reconfigurations arise from interactions between software entities, are governed by a need-fulfilment mechanism (software applications connect to external suppliers in order to meet their business goals), and consist of three distinct run-time processes: service discovery, selection and binding. What is particularly important for our work is that the binding of services, which is technically achieved through colimits, can be regarded as the service-oriented counterpart of phase-spaces composition.

The consequences of constraining free jazz

Constraint programming has already proved to be appropriate for computational music composition and modelling music theory disciplines such as harmony, rhythm, instrumentation and counterpoint (Anders and Miranda, 2011). The declarative nature and the modularity of such constraint satisfaction problem (CSP) systems match the way in which composition rules are commonly expressed in standard music theory. Deciding the satisfaction of certain properties or rules based on the true/false dichotomy is however often inadequate for the purpose of composing music, even more so when dealing with improvisation. Soft constraint satisfaction problem systems, which generalise the classical crisp variant of CSP by evaluating constraints over richer truth structures like c-semirings, valuation structures, or residuated lattices, mitigate the problem of expressing loose rules

and provide more flexibility in writing musical guidelines. We enrich the specifications of music phase spaces with the mechanism of soft CSP, considering that each musical section has a set of preferences (soft constraints) that need to be satisfied by the music that will follow it. The values with which the requirements are satisfied are elements of the truth structure's underlying set denoted herein by *Sat*. The final aim of this soft CSP formalization is to find those specifications of music phase spaces that optimize the satisfaction of the given constraints.

We illustrate the formalization of free-jazz improvisations in the context of service-oriented computing starting from the example presented in (Borgo and Goguen, 2005). The authors analysed an excerpt entitled "Hues of Melanin" from the 1973 Sam Rivers Trio's concert at Yale University. They proposed a sectional interpretation of the performance and highlighted the transitions between the music phases in order to demonstrate the nonlinear dynamics of the improvisation. The segmentation is natural and determined by the frequent and clear variations – rhythmic, timbral or chromatic – of the music flow. We focus on the first part of this examination, namely sections A to H. Although we try to make the presentation of this example self-explanatory, the reader is encouraged to consult the section "Hues of Melanin" of (Borgo and Goguen, 2005).

We consider that each musical section imposes some constraints regarding the tempo, texture, intensity, or technique details of the next musical phase to be played. But we keep in mind that, as the improvisation builds, the constraints of a musical segment evolve and adapt to the already unfolded music: the same musical section or trigger of a transition could require different continuations if played at two different moments of the performance.

```

spec FREEJAZZ =
sorts Phase, Tempo, Texture, Instrument, Detail, Transition
ops slow, medium, fast : Tempo
    repetition, groove, complexity, fragmentation, rubato : Texture
    trill, cadence, groove, drone, glissando, ascent, pedal : Detail
    N, T1, T2, T3, T4, T5, T6, T7 : Transition
    tempo : Phase → Tempo
    texture : Phase → Texture
    detail : Phase → Detail
    transition : Phase → Transition

```

Figure 1: The specification FREEJAZZ

We zoom in on the two transitions triggered by a soprano saxophone trill on D (sections C and G in Figure 2) and we regard the trill as a determining component in the evolution of the improvisation. We model these musical phases as specifications written over first-order logic using a CASL-like syntax (Mosses, 2004), sharing two common sub-specifications: one describing the truth structures used to evaluate the satisfaction of the constraints, i.e. residuated lattices (Galatos et al., 2007), and the other, the specification FREEJAZZ in Figure 1, listing the instruments played by the musicians, possible values for measuring the tempo, texture descriptors, techniques, and ornamentations that constitute the salient details of musical segments, as well as the

Letter	Time	Transition Type	Overall Texture
C	5:29	T2 (soprano trill on D)	FREE
C2	5:48	T7 (drum cadence)	
C3	6:43	T2 (soprano high note)	
C4	7:05	T2/T4/T6 (soprano, bass low A)	
G	13:10	T2 (trill on D), T5 (bass groove)	GROOVE
G2	14:04	metric sync	
G3	14:52	T6 (bass triggers descent)	FREE

Transition types: T2 *pseudo-cadential segue* – an implied cadence with sudden and unexpected continuation; T4 *feature overlap* – one feature of the antecedent section is sustained and becomes part of the consequent section; T5 *feature change* – a gradual change of one feature that redirects the flow (usually subtly); T6 *fragmentation* – a gradual breaking up, or fragmenting, of the general texture and/or rhythm; T7 *internal cadence* – a prepared cadence followed by a short silence then continuation with new material.

Figure 2: Sections and Subsections of "Hues of Melanin" (excerpt from (Borgo and Goguen, 2005), Figure 1)

types of transitions between the sections. Apart from these, a specification describing a musical phase also records the characteristics of the fragment and preferences on the musical section that will continue it. These are expressed as ordinary first-order sentences. In Section "Anthony Braxton Graphic Notation Logic" we make explicit the temporal distinction that separates them into properties of the current music fragment and properties of the next fragment.

```

spec TRIGGERINGPHASE = FREEJAZZ and RESIDUATEDLATTICES
then ops available : Phase × Instrument × Instruments → Sat
    tempoPref : Tempo → Sat
    texturePref : Texture → Sat
    instrPref : Instrument → Sat
    instrumentsPref : Set(Instrument) → Sat
    detailPref : Detail × Instrument → Sat
    transitionPref : Transition → Sat
    ∀ p : Phase; i : Instrument; is : Instruments
    • available(p, i, is) = 1 ⇔ (detail(p) = trill) ∧ (i = sax)
    • tempoPref(slow) ≤ tempoPref(medium)
    • tempoPref(fast) ≤ tempoPref(slow)
    • texturePref(complexity) ≤ texturePref(rubato)
    • texturePref(complexity) ≤ texturePref(groove)
    • instrPref(bass) = instrPref(drums) = instrPref(sax)
    • instrPref(flute) ≤ instrPref(sax)
    • instrumentsPref(S) = *(instrPref(S))
    • detailPref(trill, sax) = 0
    • transitionPref(N) = 0

```

Figure 3: The specification TRIGGERINGPHASE

For the triggering passage, we can distinguish a number of general constraints that do not depend on the context in which it is played, such as basic guidelines on the tempo, the texture, or the instruments to be played. Even though more specific preferences regarding the details of the section are left to be fixed at the actual moment of the musicking, there are some constraints regarding the need of a transition, and

thus a continuation of the passage – the concert can’t end with the trill – and the repetition of the passage – it shouldn’t be continued with another saxophone trill. These restrictions are expressed as soft constraint sentences, or *requirements* (here first-order terms), that extend the specification TRIGGERINGPHASE in Figure 3:

cvars phase : Phase; instrument : Instrument;
instruments : Instruments

- tempoPref(tempo(phase)) • texturePref(texture(phase))
- instrumentsPref(instruments) • transitionPref(transition(phase))
- detailPref(detail(phase), instrument)

These restrict the entire musical phase space to several smaller phase spaces that satisfy the ‘needs’ of our musical component. In (Borgo and Goguen, 2005), the sections C and G that follow the saxophone trills on D are considered to be alternatives for the development of the performance from our transitional point onwards: we can regard them as different solutions for a constraint problem. Although both sections satisfy the requirements imposed by playing the trill (the sentences of the specification TRIGGERINGPHASE), the valuation of the constraint sentences above (which come with the already performed music) will discriminate between one choice and the other. We can think of sections C and G as epitomes of two growth directions or musical phase spaces: in the first phase, a medium-tempo short bass groove passage is soon abandoned for a rubato (the phase space fights the groove towards a modal area), while in the second section a groove similar to the one that was ended prematurely is explored (the phase space comprises passages of groove exploration and/or increased complexity).

The specifications PHASESPACEC and PHASESPACEG correspond to the two alternative phase spaces presented in our example above. Each of these contains one of the two sections played during the actual improvisation.

spec PHASESPACEC = FREEJAZZ and RESIDUATEDLATTICES
then ops available : Phase \times Instrument \times Instruments \rightarrow Sat
p1, p2 : Phase
 $\forall p$: Phase; i : Instrument; is : Instruments
• available(p, i, is) = 1 \Leftrightarrow (p, i, is) belongs to the following table

ph	tempo	texture	instruments	detail + in	T
p1	medium	rubato	drums, bass, sax	cadence: drums	T7
p2	slow	rubato	drums, bass	cadence: drums	T6

Figure 4: The specification PHASESPACEC

spec PHASESPACEG = FREEJAZZ and RESIDUATEDLATTICES
then ops available : Phase \times Instrument \times Instruments \rightarrow Sat
p1, p2, p3 : Phase
 $\forall p$: Phase; i : Instrument; is : Instruments
• available(p, i, is) = 1 \Leftrightarrow (p, i, is) belongs to the following table

ph	tempo	texture	instruments	detail + in	T
p1	medium	groove	drums, bass, flute	drone: bass	T5
p2	medium	groove	drums, bass, sax	groove: bass	T5
p3	fast	complexity	drums, bass, sax	trill: sax	T2

Figure 5: The specification PHASESPACEG

Service-oriented computing

Building on these specifications of music phases, we will model free-jazz improvisations as service-oriented processes. The framework of service-oriented computing that we consider herein is in keeping with (Chiriță, Fiadeiro, and Orejas, 2016) and deals with two kinds of entities: *service applications* and *service modules*. The former are the executing units that trigger the discovery of a required service or resource, whereas the modules providing services will be executed only after they are bound to the application. Service applications have an orchestration part, a specification defining their behaviour, and interfaces describing the services required: interfaces are sub-specifications of the given orchestration together with properties that express preferences regarding the use of a given service. Modules are similar to applications in that they comprise an orchestration and interfaces; in addition, they include a description of the functionality or the resources provided.

Definition 1. A *service application* (Σ, I) consists of a specification Σ , called *orchestration*, together with a finite family $I = \{(i_x, r_x)\}_{x \in \bar{n}}$ of *interfaces*, each of which consisting of a mapping $\iota_x : \Sigma_x \rightarrow \Sigma$ such that Σ and Σ_x share the same residuated lattice, and a *requirement* $r_x \in \text{CSen}(\Sigma_x)$, where $\text{CSen}(\Sigma_x)$ is the set of all constraint sentences that can be associated to Σ_x .

Definition 2. A *service module* (Ω, P, J) consists of an *orchestration* Ω , a *provides-property* $P \in \text{CSen}(\Omega)$, and a finite family $J = \{(j_y, q_y)\}_{y \in \bar{m}}$ of *interfaces*, consisting of mappings $j_y : \Omega_y \rightarrow \Omega$ and *requirements* $q_y \in \text{CSen}(\Omega_y)$.

We consider that the execution of service applications takes place in the context of a fixed set of service modules – a *service repository*. Each execution step is triggered by the need to fulfil a requirement of the current application, which in the context of our work corresponds to a *requires-interface*. Similarly to conventional soft-constraint satisfaction problems, the goal is to maximize the satisfaction of the requirement. To this end, we distinguish three elementary processes: *discovery*, *selection* and *binding*. For a service application and one of its *requires-interfaces*, the *discovery process* will provide a set of possible matches, i.e. pairs of service modules from the repository and attachment mappings from the *requires-specification* to the *orchestrations* of the modules. Note that the output of the *discovery process* only depends on the repository and the selected *requires-interface*, and not on the application itself. In the *selection process*, for every match retrieved in the *discovery phase*, a score of compatibility with the requirement will be computed using the concept of graded semantic consequence (Diaconescu, 2014): the value with which the *provides-property* semantically entails the requirement. The application then commits to the chosen provider through the *binding process*: the orchestration of the application is blended with the orchestration of the selected service module (via the computation of a pushout of the two specifications), while the fulfilled requirement is replaced with the requirements of the added module. For more technical details on how the service processes are modelled, we refer the interested reader to (Chiriță, Fiadeiro, and Orejas, 2016).

Music improvising as service-oriented processes

We model a free-jazz performance as a service application, and each musical section to be played as a module. In both cases, the first-order specifications of the music fragments act as orchestrations, while the constraint sentences act as requires-interfaces. Each musical transition determines a round of processes of service discovery, selection and binding: for each section played, a best supplier (musical passage) will be selected from the pool of all possible continuations, and it will be “added” to the current application, thus extending the record of the music already played. Choosing a version or another at a given point in the development of the music depends on the way the selection of a best model is defined, and also on the content of the orchestration at that point: playing a musical trigger will influence both the way we assign a preference value to a musical-fragment candidate – the interpretations of the constraints – and the space of the satisfaction values upon which we judge the compatibility of two musical segments.

Upon the selection of one provider, the binding of the application to the chosen service module represents the commitment of the music performance to one of the two phase spaces: the music blending is realized by the pushout of the two specifications that define the orchestrations.¹ We stress the fact that in this case, the concept blending is not determined by the mere juxtaposition of musical fragments, but by the fact that the class of possible music exploration paths is narrowed with each binding through refinement. At the end of each reconfiguration round, the performed improvisation could be evaluated using two results presented in (Chiriță, Fiadeiro, and Orejas, 2016): the formalization of the concept of α -satisfiability (where α is a degree of satisfaction) and a theorem stating that the binding process is sound with respect to α -satisfiability.

Modelling “Hues of Melanin”

Consider the service application $\mathcal{A} = (\Sigma, I)$ whose orchestration Σ is TRIGGERINGPHASE (as in Figure 3), and whose interface consists of the identity map and the requirement

$$\begin{aligned} & \text{tempoPref}(\text{tempo}(\text{phase})) \wedge \text{texturePref}(\text{texture}(\text{phase})) \\ & \wedge \text{instrumentsPref}(\text{instruments}) \wedge \text{transitionPref}(\text{transition}(\text{phase})) \\ & \wedge \text{detailPref}(\text{detail}(\text{phase}), \text{instrument}). \end{aligned}$$

We fix the repository $\text{Rep} = \{\mathcal{C}, \mathcal{G}\}$, where

- the service module $\mathcal{C} = (\Omega, P, J)$ has the orchestration PHASESPACEC in Figure 4, the provides-property $P = \text{available}(\text{phase}, \text{instrument}, \text{instruments})$, and no requirements, and
- the service module $\mathcal{G} = (\Omega', P', J')$ is such that Ω' is as in Figure 5, and $P' = P$.

¹Although we use colimits to model concept blending, we are not strictly following the approach from (Goguen, 1999). We do not generate the input morphisms and the base space: the base space and one of the input spaces are part of the service application; the other input space, together with its corresponding morphism, follow from the discovery process, which is modelled here as an external mechanism that can be thought of as a ‘black box’.

When selecting a best supplier for \mathcal{A} , the music phases that best fit the preferences are phase $p1$ for \mathcal{C} and phase $p2$ for \mathcal{G} . In principle, we would need to compute the compatibility scores between TRIGGERINGPHASE and PHASESPACEC and PHASESPACEG, respectively, using all possible models. However, due to the way the specifications are written, the choice of the best phase for each phase space can be inferred directly from the axioms. First, the constraint variables phase, instrument and instruments are limited to the interpretations defined in the tables. Second, the axioms of TRIGGERINGPHASE that express specific preferences, such as for a tempo, make it feasible to determine the best phases provided by each phase space for any model. With respect to tempo, phase $\mathcal{G}.p3$ is the least preferred, while $\mathcal{C}.p1$ and $\mathcal{G}.p1, \mathcal{G}.p2$ are the most preferred because $\text{tempoPref}(\text{fast}) \leq \text{tempoPref}(\text{slow}) \leq \text{tempoPref}(\text{medium})$. However, we cannot decide which one of the two phase spaces would be more suitable, since we cannot decide whether $\mathcal{C}.p1$ or $\mathcal{G}.p2$ satisfies better the constraints. Therefore, any of the two modules could be non-deterministically selected.

Logics for improvising

First-order logic, although standard for formal specifications, may not be the most suitable logic for describing the features of a music fragment; we could specify music by employing other logics with a more convenient syntax, closer to the usual musical notation. To ensure that the framework presented in the previous section can still be used for dealing with improvised music, we impose a set of reasonable restrictions that the new logics must meet. We argue that any logic satisfying the constraints described in (Chiriță, Fiadeiro, and Orejas, 2016) is generally suitable for capturing free-jazz improvisations: informally, the logic should

- permit the expression of constraints, as the aim of free jazz is “to play together with the greatest possible freedom – which, far from meaning without constraint, actually means to play together with sufficient skill and communication to be able to select proper constraints in the course of the piece”(musician Ann Farber, see (Borgo, 2005), Chapter “Reverence for Uncertainty”),
- permit the partial satisfaction of constraints, as improvisation requires flexibility and non-rigid answers, and
- allow the change of the truth system, as players in a collaborative performance usually have different beliefs and value systems that they impose to the group alternately.

Non-standard notations used in composition notes and guide scores for improvised performances could be used as a basis for defining logics having the expressiveness needed for specifying free jazz. In the following, we show how such notations can be formalized as appropriate logics for improvisation processes, focusing on one of Anthony Braxton’s alternative notations for free-jazz composition.

Anthony Braxton Graphic Notation Logic

Through the extensive use of graphic and symbolic notations, Braxton’s music positions itself at the fuzzy border between composition and improvisation – see (Lock, 2008).

Neither completely notated, nor completely free, the scores can be seen as an incipient guideline for the improviser: the visual elements force the performer to assign personal interpretations to rather abstract forms that would otherwise make the scores unplayable by immutably following the more conventional notations. The improviser must hence intervene considerably in the composition, not in the usual form of jazz extended solos, but with “tiny pockets of improvisational space” (Lock, 2008) that should fill the non-finished musical structure. This porosity, an inviting-to-improvisation characteristic of his compositions, is also apparent in the work on which we will focus: “Composition 94 for Three Instrumentalists” (Braxton, 1988). In section B of this piece, Braxton uses an *image grouping notation* consisting of three types of contours that are overlaid on top of standard pitches to create the so-called *liquid, shape, and rigid formations*. The role of the formations is to indicate the performers the outlines that they should follow in playing the notes inside them: these pitches should not be played as they appear in the score, but transformed according to the distinctive interpretation of each improviser. The pitches inside liquid formations, figures resembling clouds, should be played as “clouded mass sound imprints”, the shape formations should suggest “harder edges”, while the rigid formations, closer to geometrical figures, should highlight their “composite state”. In this study, we loosen the restrictions for the interpretation of the formations, blurring the distinction between the three types of formations, and we accept as valid the improvisations that replace the notes within the shapes with completely different pitches given that they both allude to the original notes, and evoke the contours. We will reduce the problem of quantifying the improvisation’s reminiscence of the original pitches to the problem of measuring the similarity of two music fragments (the interested reader is referred, for example, to (Mongeau and Sankoff, 1990) for further details on the comparison of musical fragments).

The observations above lead to a straightforward formalization of Braxton’s graphic notation as a many-valued logic \mathcal{BGN} . To obtain a representation of the scores, and furthermore, to express properties of the music segments at certain positions in a score, the language of \mathcal{BGN} must comprise the universe of all possible music fragments written in a standard notation, a set of formations, and an appropriate truth structure that will allow us to manipulate partially true statements. We define hence a signature Σ as a triple $(\mathcal{L}, \text{MF}, \text{FS})$, where \mathcal{L} is a residuated lattice, MF is a set of music fragments, and FS is a set of formation symbols. The morphisms of signatures, i.e. mappings that permit translations from one language to another, are defined component-wise: a morphism $\varphi: \Sigma \rightarrow \Sigma'$ consists of a morphism of residuated lattices $\varphi_{rl}: \mathcal{L}' \rightarrow \mathcal{L}$, a function φ_{MF} between the sets of fragments MF and MF', a function φ_{FS} from the set of formation symbols FS to FS', and a natural number l representing a delay between the moment of playing a score written over the first signature and the moment of playing a score written over the second one. We admit three types of atomic sentences built using the symbols in the signatures:

- $m@p$, with $p \in \mathbb{N}$ and $m \in \text{MF}$, which should be read as “at position p we have the music fragment m ”,

- $\sim m@p$, with $p \in \mathbb{N}$ and $m \in \text{MF}$, which should be read as “at position p we have a fragment similar with m ”, and
- $s(@p)$, with $s \in \text{FS}$ and $p \in \mathbb{N}$, which should be read as “the fragment at position p is in the shape of s ”.

We will denote by $s_p(m)$ the conjunction $s(@p) \wedge \sim m@p$.

For any morphism of signatures $\varphi: \Sigma \rightarrow \Sigma'$, we can translate the atoms over Σ to atoms over Σ' as follows:

- $\varphi(m@p) = (m@(p+l))$
- $\varphi(\sim m@p) = \sim m@(p+l)$
- $\varphi(s(@p)) = \varphi_{\text{FS}}(s)(@p+l)$

The semantics of \mathcal{BGN} is given by classes of models corresponding to every signature: every Σ -model M consists of a set $|M|$ of interpretations of music fragments, a method $M_{\sim}: |M| \times |M| \rightarrow \mathcal{L}$ for measuring the similarity of two segments as a value of \mathcal{L} , interpretations $M_s: |M| \rightarrow \mathcal{L}$ of the formation symbols $s \in \text{FS}$, and a sequence of music fragments $M_{\text{seq}} \in |M|^*$ to be played. We will usually denote the fragment at the position p in M_{seq} by $M_{\text{seq}}[p]$; we will sometimes describe a sequence through a regular-expression-like string in which interrogation points mark the parts that are yet to be fixed (containing formation symbols), and the $*$ symbol marks the fact that the sequence is open and admits any possible succession of music fragments.

Models M can satisfy a sentence ρ with a many-valued truth degree from the residuated lattice, denoted by $M \models \rho$:

- $M \models (m@p)$ is defined as $\begin{cases} 0, & \text{if } M_{\text{seq}}[p] \neq m \\ 1, & \text{if } M_{\text{seq}}[p] = m \end{cases}$
- $M \models (\sim m@p)$ is given by the similarity of m and the music fragment at position p , i.e. $M_{\sim}(m, M_{\text{seq}}[p])$,
- $M \models (s(@p))$ is given by the resemblance $M_s(M_{\text{seq}}[p])$ of the fragment at position p with the shape s .

The fact that a signature does not determine the interpretation of the music fragments, the similarity measure, or the interpretation of the formation symbols, makes the logic \mathcal{BGN} too general for suitably specifying music: we would like to be able to control, for example, which similarity measure to use in comparing music fragments. We hence define \mathcal{SBGN} to be the logic having as signatures pairs consisting of \mathcal{BGN} -signatures Σ and fixed classes \mathcal{M} of Σ -models. The signature morphisms of \mathcal{SBGN} , which will play an important role in defining service discovery and binding, are defined as usual, as the \mathcal{BGN} -signature morphisms $\varphi: \Sigma \rightarrow \Sigma'$ for which the associated model reduct $\varphi_{\mathcal{M}}^2$ satisfies the property $\varphi_{\mathcal{M}}(\mathcal{M}') \subseteq \mathcal{M}$.

A “Clapping Music” improvisation

To illustrate how an improvised performance based on composition notes written using the graphic notation of Anthony Braxton can be seen as a series of dynamic processes between service modules specified over \mathcal{SBGN} , we choose to

²We recall that any Σ' -model M' can be reduced along the signature morphism φ to a Σ -model $\varphi_{\mathcal{M}}(M')$ simply by forgetting the interpretations of the new symbols that the morphism introduces; see (Sannella and Tarlecki, 2012) for more details.



Figure 6: Scores A, B, C

reduce to a minimum the details particular to music-theory. This simplification is intended to: (1) underline the fact that the freeness of the performance does not reside primarily in the qualitative aspects of the resulted music, but in the nature of the musicking process itself, and (2) alleviate the effort of the reader unfamiliar with basic notions of music theory.

We formalize our example starting from Steve Reich’s minimalist composition “Clapping Music” written in 1972. Although a complete composition, with no musical segments meant to be improvised, the piece constitutes a good reference for our purpose due to its simplicity. Written around a basic pattern very similar to the standard African 12/8 bell pattern, the piece should be played by two performers: one should continuously and unvaryingly repeat the basic pattern, while the other should repeat and shift the pattern with one note after each eight bars.

We use fragments of this composition to exemplify the binding of services that specify incomplete musical segments written in Braxton’s notation: we start from the first bar of the piece, the basic pattern (see Figure 6, A), and we let the performance develop according to both fixed, rigid instructions, and loose, subject to improvisation guidelines. Bar a is followed by a shape formation \sqsubset specifying that the pattern should be repeated, but in a transformed state reminding of descending steps (fragment x), and by the fixed fragment b , to which other fragments may be added.

In the following, we will consider the universe MF of musical fragments to be the set of all the possible score fragments obtained from composing the basic pattern a and the patterns obtained by shifting it, together with the prefixes of these shifts. We formalize the starting score as a service application $\mathcal{A} = (\Sigma, I)$ with

- the orchestration Σ given by $(\mathcal{L}_\Sigma, \text{MF}, \{\sqsubset\}, \mathcal{M}_\Sigma)$, where \mathcal{M}_Σ is the class of the models that correspond to sequences of music fragments described as $a?b^*$;
- a single interface $(i: \Sigma_1 \rightarrow \Sigma, R)$, where i_{rl} , i_{MF} and i_{FS} are all identities, $i_l = 1$ (to indicate that the variable fragment x appears at position 1) and $i_{\mathcal{M}}$ the inclusion of \mathcal{M}_Σ in the class \mathcal{M}'_Σ of models that correspond to all

sequences of music fragments, which are described as $*$,³ and $R = \sqsubset_0(x) = \sqsubset_0(@0) \wedge (\sim x@0)$.

To refine and continue the given music score, we consider a round of processes of discovery, selection and binding of other music fragments to our original fragment. Let the result of the discovery process be the set of the scores B and C in Figure 6. Formally, they are service modules $\mathcal{B} = (\Omega_B, P_B, J_B)$ and $\mathcal{C} = (\Omega_C, P_C, J_C)$ as follows:

- their orchestrations are $\Omega_B = (\mathcal{L}_{\Omega_B}, \text{MF}, \emptyset, \mathcal{M}_{\Omega_B})$ and $\Omega_C = (\mathcal{L}_{\Omega_C}, \text{MF}, \{\sqtriangleleft\}, \mathcal{M}_{\Omega_C})$ with the classes of models \mathcal{M}_{Ω_B} and \mathcal{M}_{Ω_C} defined by the sequences of music fragments cbe^* and $bebe^?*$, respectively;
- they guarantee to begin with the fragments c and $bebe^4$ through the provides-properties $P_B = c@0$ and $P_C = bebe@0$;
- we model the fact that the score B is completely fixed, not demanding improvisation, by considering the set of requirements to be empty;
- the interface $j_C: \Omega'_C \rightarrow \Omega_C$ of \mathcal{C} is defined similarly to the interface of \mathcal{A} : it consists of identities for the residuated lattice, the music fragment space and the set of formation symbols, the natural number 1, indicating the position of the formation symbol in the score, and the inclusion of the class of models \mathcal{M}_{Ω_C} in the class $\mathcal{M}'_{\Omega'_C}$ of models having all possible sequences of music fragments;
- the requirement $Q_C = \sqtriangleleft_0(y) = \sqtriangleleft_0(@0) \wedge (\sim y@0)$.

In order to perform a selection between the two service modules, we would have to further limit the classes of models \mathcal{M}_Σ , \mathcal{M}_{Ω_B} and \mathcal{M}_{Ω_C} . To determine how suitable one fragment is for the intended improvisation, we would need to fix a set of measures of similarity between music fragments and of acceptable interpretations for the formation symbol \sqsubset : How similar are the fragments c and $bebe$ to x , and how much can they be perceived as sounds in the shape of \sqsubset ? Let us skip this phase of our running example, considering the score B to be the result of an arbitrary selection process, and focus on the binding of modules as a process of blending music fragments.

By computing the pushout of the morphisms i and ϕ that map the requires-specification to the orchestration of the application and to the provides-specification of the service module, we amalgamate the models⁵ in \mathcal{M}_Σ and \mathcal{M}_{Ω_B} , and hence their musical sequences, $a?b^*$ and cbe^* respectively, obtaining the contiguous music score $acbe^*$. (Note the role of the delay 1 corresponding to the morphisms i and j .)

$$\begin{array}{ccc}
 \Sigma_1 & \xrightarrow{i} & \Sigma & & * & \xrightarrow{1} & a?b^* \\
 \phi \downarrow & & \downarrow i' & & 0 \downarrow & & \downarrow 0 \\
 \Omega_B & \xrightarrow{j} & \Sigma' & & cbe^* & \xrightarrow{1} & acbe^*
 \end{array}$$

³We choose not to limit the class of models through the interface because we want to be able to consider as a candidate in the selection process any music fragment satisfying the requirement R .

⁴Note the $\times 4$ superscript in Figure 6, B denoting that c is the repetition of the highlighted fragment four times.

⁵Here we use the specification-theoretic notion of model amalgamation, see for example (Sannella and Tarlecki, 2012).

The substitution of the orchestration Σ of \mathcal{A} with the vertex Σ' of the pushout will hence determine the refinement of the class of models \mathcal{M}_Σ , both filling the improvisational gaps of the performance, and molding its evolution.

Conclusions

In this paper we have shown how the fundamental processes of service discovery, selection and binding can be used to model free-jazz performances. The main step in this endeavour was to identify which logical formalisms are suitable for capturing and reasoning about free-jazz, and at the same time are compatible with the principles of the service-oriented computing paradigm. To this end, we have first discussed a variant of many-valued first-order logic endowed with constraints, which is closer to the formalisms used in previous developments on service-oriented computing (Chiriță, Fiadeiro, and Orejas, 2016); we have then defined a novel logic, particular to free-jazz, based on Anthony Braxton's graphic notations. The proposed formalization paves the way to reasoning about improvisation processes: one can now study aspects related to reliability (to what extent the user's expectations can be met?), determine which music fragments are hardly reachable (never or seldom used during a play), or make predictions about the evolution of an improvisation (e.g. how does the choice of a music fragment affect the subsequent use of other fragments?).

Our work has focused on the musicking process itself, not on the resulting music: we do not provide a way to record the improvisation; instead, the music is played at run-time, whenever a new fragment is sublated into the performance. We would like to continue to pursue this line of research by implementing an *SBGN* specification and programming language whose operational semantics extends the logic programming of services from (Țuțu and Fiadeiro, 2015) by taking into account many-valued truth spaces. In this way, we could create novel computationally creative music systems that exploit the operational semantics of service applications to deliver improvised music to users based on an online repository of music fragments. The repository would be ever-changing, hence, even if the user's input were the same, the composed music could vary significantly.

Acknowledgements

The authors are grateful to Nuno Barreiro for his continuous encouragement and helpful discussions throughout the course of this work.

References

- Anders, T., and Miranda, E. R. 2011. Constraint programming systems for modeling music theories and composition. *ACM Comput. Surv.* 43(4):30.
- Blackwell, T., and Young, M. 2004. Self-organised music. *Organised Sound* 9(02):123–136.
- Borgo, D., and Goguen, J. 2005. Rivers of consciousness: The nonlinear dynamics of free jazz. In *Jazz Research Proceedings Yearbook*, volume 25.
- Borgo, D. 2005. *Sync or swarm: Improvising music in a complex age*. A&C Black.
- Braxton, A. 1988. *Composition notes*, volume 3. Synthesis Music.
- Chiriță, C. E.; Fiadeiro, J. L.; and Orejas, F. 2016. Many-valued institutions for constraint specification. In *FASE 2016*, volume 9633 of *LNCS*, 359–376. Springer.
- Diaconescu, R. 2014. Graded consequence: an institution theoretic study. *Soft Comput.* 18(7):1247–1267.
- Eppe, M.; Confalonieri, R.; Maclean, E.; Kaliakatsos-Papakostas, M. A.; Cambouropoulos, E.; Schorlemmer, W. M.; Codescu, M.; and Kühnberger, K. 2015. Computational invention of cadences and chord progressions by conceptual chord-blending. In *IJCAI 2015*, 2445–2451. AAAI Press.
- Fiadeiro, J. L. 2012. The many faces of complexity in software design. In *Conquering Complexity*. Springer. 3–47.
- Galatos, N.; Jipsen, P.; Kowalski, T.; and Ono, H. 2007. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Studies in Logic and the Foundations of Mathematics. Elsevier Science.
- Goguen, J. 1999. An introduction to algebraic semiotics, with application to user interface design. In *Computation for metaphors, analogy, and agents*. Springer. 242–291.
- Kaliakatsos-Papakostas, M.; Cambouropoulos, E.; Kühnberger, K.-U.; Kutz, O.; and Smaill, A. 2014. Concept invention and music: Creating novel harmonies via conceptual blending. In *CIM2014*.
- Lock, G. 2008. What I call a sound: Anthony Braxton's synaesthetic ideal and notations for improvisers. *Critical Studies in Improvisation* 4(1).
- Mazzola, G., and Cherlin, P. B. 2008. *Flow, gesture, and spaces in free jazz: Towards a theory of collaboration*. Springer Science & Business Media.
- Mongeau, M., and Sankoff, D. 1990. Comparison of musical sequences. *Comput. Hum.* 24(3):161–175.
- Mosses, P. D. 2004. *CASL Reference Manual*, volume 2960 of *LNCS*. Springer.
- Pearce, M.; Meredith, D.; and Wiggins, G. 2002. Motivations and methodologies for automation of the compositional process. *Musicae Scientiae* 6(2):119–147.
- Ramalho, G., and Ganascia, J. 1994. Simulating creativity in jazz performance. In *AAAI 1994*, 108–113. AAAI Press / The MIT Press.
- Sannella, D., and Tarlecki, A. 2012. *Foundations of Algebraic Specification and Formal Software Development*. Springer.
- Țuțu, I., and Fiadeiro, J. L. 2015. Service-oriented logic programming. *LMCS* 11(3):1–38.
- Wiggins, G. A.; Pearce, M. T.; Müllensiefen, D.; et al. 2009. Computational modeling of music cognition and musical creativity. In *Oxford Handbook of Computer Music*. Oxford University Press. chapter 19.

An Argument-based Creative Assistant for Harmonic Blending

Maximos Kaliakatsos-Papakostas^a, Roberto Confalonieri^b, Joseph Corneli^c,
Asterios Zacharakis^a and Emiliios Cambouropoulos^a

^aDepartment of Music, Aristotle University of Thessaloniki, Greece

^bArtificial Intelligence Research Institute, IIIA-CSIC, Bellaterra (Barcelona), Spain

^cDepartment of Computing, Goldsmiths, University of London, UK

^a{max,aszachar,emilios}@mus.auth.gr

^bconfalonieri@iia.csic.es

^cj.corneli@gold.ac.uk

Abstract

Conceptual blending is a powerful tool for computational creativity where, for example, the properties of two harmonic spaces may be combined in a consistent manner to produce a novel harmonic space. However, deciding about the importance of property features in the input spaces and evaluating the results of conceptual blending is a nontrivial task. In the specific case of musical harmony, defining the salient features of chord transitions and evaluating invented harmonic spaces requires deep musicological background knowledge. In this paper, we propose a creative tool that helps musicologists to evaluate and to enhance harmonic innovation. This tool allows a music expert to specify arguments over given transition properties. These arguments are then considered by the system when defining combinations of features in an idiom-blending process. A music expert can assess whether the new harmonic idiom makes musicological sense and re-adjust the arguments (selection of features) to explore alternative blends that can potentially produce better harmonic spaces. We conclude with a discussion of future work that would further automate the harmonisation process.

Introduction

The invention of new harmonic spaces in this paper is conceived as a computational creative process according to which a new harmonic idiom is created by means of blending the ‘atoms’ of harmony, i.e., transitions between chords. The blended transitions are created by combining the features characterising pairs of transitions belonging to two idioms (expressed as sets of potentially learned transitions) according to an amalgam-based algorithm (Confalonieri et al., 2015a; Eppe et al., 2015b) that implements Fauconnier and Turner (2002)’s theory of conceptual blending. The transitions are then used in an extended harmonic space that accommodates the two initial harmonic spaces, linked with the new blended transitions.

When modeling creative processes computationally, one of the key questions is how good are the created artefacts. The approach to evaluation that has been applied most frequently within computational creativity requires a human to evaluate attributes of the created work or the system’s operation. Basic measures consider the *typicality* of a generated artefact within a particular genre, or the *quality* of the generated work according to the users’ aesthetic judgement (Ritchie, 2007).

In music blending, the evaluation of artefacts is not a trivial matter. This is due not only to the time evolving nature of the final output, but also to the lack of clearly defined criteria for their assessment. In the particular case of transition¹ blending, which is how harmonic blending is approached in this paper, the evaluation of the blends is of key importance, in order to produce musically meaningful extended harmonic spaces. To evaluate the set of blended transitions and the corresponding generated extended harmonic space, several musical features need to be taken into account according to indications by musicologists. The importance of each particular feature, however, is not known in advance and musicologists need to make adjustments by experimenting with a large set of test cases.

To ease this task, in this paper, we propose a creative tool (Figure 1) that assists a musicologist with the evaluation of harmonic blends. The system allows a musicologist to specify *arguments*—abstracting the properties of chords and transitions—and to use them for an iterative evaluation of the blended outcome, based on the transitions that the system proposes in order to connect two (potentially remote) harmonic spaces.

Using arguments to make and explain decisions has been proposed and explored in Artificial Intelligence (Bench-Capon and Dunne, 2007), where an argument is a reason for believing a statement, choosing an option, or doing an action. In most existing works on argumentation, an argument is either considered as an abstract entity whose origin and structure are not defined (Dung, 1995), or it is a logical proof for a statement where the proof is built from a knowledge base (Amgoud and Prade, 2009). The use of argumentation in concept invention is, on the other hand, less frequent. Confalonieri et al. (2015b) use Lakatosian’s reasoning to model dialogues in which users engage to discuss the intended meaning of blended concept.

In our approach, arguments encapsulate desirable properties that the user would like to have in the resulting transition blends. Arguments are specified by the user by answering specific questions over the features of the idioms selected as input for the transition blending process. Providing some

¹For the rest of the paper, the term transition will be referring to a pair of chords where one follows the other. For example, G7 → C is a transition describing the perfect cadences in tonal music.

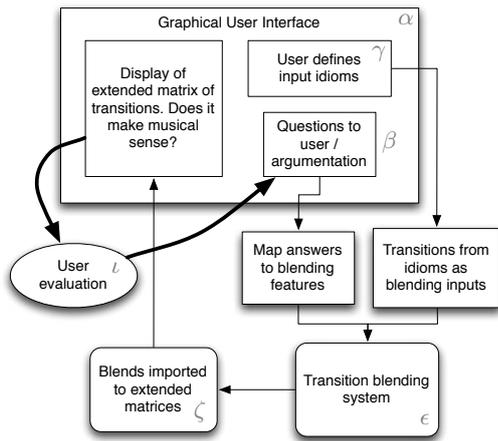


Figure 1: A schematic diagram of the system's workflow.

higher level arguments as inputs to the system is equivalent to allowing a musical expert to interact with it in a language he/she understands. This offers the user a flexible way to adjust the harmonic blending properties according to different input scenarios in order to improve the creativity of the system. Extended experimentation with the system—by making use of the available arguments—can enable music experts to provide valuable feedback regarding the functionality of transition features, thus directly intervening in the blending process by answering simple questions. In a future scenario, the assessment of the system will be based on merely musicological criteria that should be more clearly defined.

The paper is organised as follow. In the next section, we describe the harmonic blending creative process embedded in a creative assistant tool we implemented. Next, we describe the methodology of transition blending and extending harmonic spaces. We show how user arguments are used to evaluate transition blends based on two criteria resembling two of the optimality principles of conceptual blending. Then, we present a process-based system evaluation that focuses on the creative acts of programmers (Colton et al., 2014). This evaluation is helpful in guiding further developments of the system. These are discussed in a concluding section.

System overview and test cases

Figure 1 illustrates a diagram of the presented system. The user (music expert) interacts with the system through the Graphical User Interface (GUI), where she/he selects two initial idioms (harmonic spaces expressed as sets of permitted chord transitions) in γ and defines the important features used in conceptual blending by answering specific questions (argumentation) in β . The selected initial idioms are described as sets of chord transitions, while the provided answers to questions are mapped to enabling/disabling features of transitions (see Section ‘Chord transitions description and blending’) that define the outcome of transition

blending (see Section ‘Evaluation of transition blending via arguments’).

Afterwards, pairs of transition in the two initial harmonic spaces are given as inputs to the transition blending system in ϵ where new transitions are invented through conceptual blending.² These transitions are then integrated into an *extended* musical idiom that includes the initial idioms selected by the user. While the role of the new transitions is to provide musically meaningful connections between the initial harmonic spaces. The created extended idiom is displayed to the user in the GUI in terms of a transition matrix (see Section ‘From transition blends to transition matrices’). By observing the matrix, the music expert evaluates (ι) the results produced by the current blending setup, i.e., the given questions to the argumentation module (β), and re-adjusts her/his answers in β accordingly.

Several scenarios for initial idiom combinations are available to the user. The system includes several harmonic blending test cases according to which the user can blend simple ‘artificial’ harmonic spaces as well as harmonic spaces trained from data in different tonalities. The artificial harmonic spaces are manually constructed to include simple transitions that can typically be found in tonal music in order to allow clear interpretations of the results, e.g., a C major space included the chords C, F and G7. Among the trained idioms that have been examined, there are sets of Bach chorales in major and minor mode, and sets of modal chorales in several modes.

The test cases, in which harmonic spaces in different tonalities are blended, resemble the musical task of finding transition paths for tonality modulations (changing the tonality of a given harmonic space). This task allowed music experts to identify arguments for defining the features of transition blending that connect potentially remote harmonic spaces (e.g., C major with F \sharp major) in a manner that is explainable in music theory in terms of tonality modulations. Through the processes offered by the system, the music experts were able to come to conclusions about what transition features are important for constructing meaningful connections between different combinations of pairs of initial harmonic spaces.

Methodological aspects of transition blending and extending harmonic spaces

The cognitive theory of conceptual blending by Fauconnier and Turner (2002) has been extensively used in linguistics, music composition (Zbikowski, 2002), music cognition (Antovic, 2009, 2011) and other domains mainly as an analytical tool, which is useful for explaining the cognitive process that humans undergo when engaged in creative acts. According to this theory, human creativity is modeled

²For instance, if the two initial harmonic spaces are a tonal C minor (I_1) and a modal C phrygian space (I_2), then a pair of transitions for blending could be $G7 \rightarrow Cm$ (from I_1) and $B\flat m \rightarrow Cm$ (from I_2). A possible resulting transition from blending these input transitions is the tritone substitution cadence, $C\sharp 7 \rightarrow Cm$, as computed in Eppe et al. (2015b) and Zacharakis, Kaliakatsos-Papakostas, and Cambouropoulos (2015).

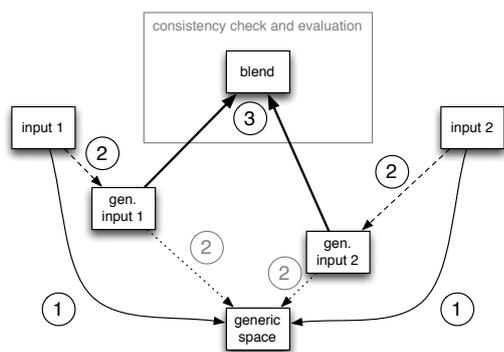


Figure 2: Conceptual blending based on amalgam. The generic space is computed (1) and the input spaces are successively generalised (2), while new blends are constantly created (3). Some blends might be inconsistent or purely evaluated according to blending optimality principles or domain specific criteria.

as a process by which a new concept is constructed by taking the commonalities among two *input spaces* into account, to form a so-called *generic space*, and by projecting their non-common structure in a selective way to a novel blended space, called a *blend*.

In computational creativity, conceptual blending has been modeled by Goguen (2006) as a generative mechanism, according to which input spaces are modeled as *algebraic specifications* and a blend is computed as a categorical *colimit*. A computational framework that extends Goguen's approach has been developed in the context of the COncEpt INVENTion Theory³ (COINVENT) project (Schorlemmer et al., 2014) based on the notion of *amalgams* (Ontañón and Plaza, 2010). According to this framework, *input spaces* are described as sets of features, properties and relations, and an *amalgam*-based workflow finds the blends (Confalonieri et al., 2015a; Eppe et al., 2015b). The amalgam-based workflow generalises input concepts until a generic space is found and 'combines' generalised versions of the input spaces to create blends that are consistent or satisfy certain properties that relate to the knowledge domain (Figure 2).⁴

Amalgam-based conceptual blending has been applied to invent chord cadences (Eppe et al., 2015a; Zacharakis, Kaliakatsos-Papakostas, and Cambouroopoulos, 2015). In this setting, cadences are considered as special cases of chord transitions—pairs of chords, where the first chord is followed by the second one—that are described by means of features such as the roots or types of the involved chords, or intervals between voice motions, among others. When blending two transitions, the amalgam-based algorithm first finds a generic space between them (point 1 in Figure 2). For instance, in the case of blending the perfect with the

³<http://www.coinvent-project.eu>

⁴In the process of blending through amalgams, the notions of 'amalgam' and 'blend' are the same. Therefore, in the following paragraphs they are used interchangeably.

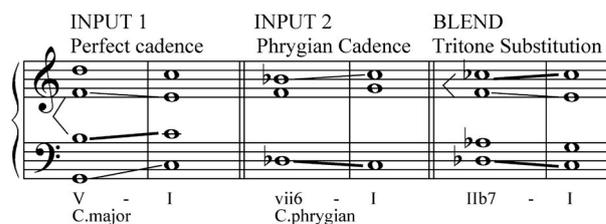


Figure 3: Example of blending cadences, which are special case of transitions, where blending the perfect and the Phrygian produce the tritone substitution cadence blend.

Phrygian cadences (Figure 3)—described by the transitions $I_1: G7 \rightarrow C$ and $I_2: B\flat m \rightarrow C5$ respectively—their generic space consists of any transition that has a second chord with the root note C, since this is the root note of both inputs' second chords (C and C5).

After a generic space is found, the amalgam-based process computes the amalgam of two input spaces by *unifying* their content. If the resulting amalgam is inconsistent, then it iteratively generalises the properties of the inputs (point 2 in Figure 2), until the resulting unification is consistent (point 3 in Figure 2). For instance, trying to directly unify the transitions $I_1: G7 \rightarrow C$ and $I_2: B\flat m \rightarrow C5$ would yield an inconsistent amalgam, since a transition cannot both include and *not* include a leading note to the second chord's tonic (which is a property of I_1 and the I_2 respectively). Therefore, the amalgam-based process generalises the clashing property in one of the inputs (e.g., the property describing the absence of leading note would be left empty in I_2) and tries to unify the generalised versions of the inputs again. After a number of generalisation steps are applied (point 2 in Figure 2), the resulting blend is consistent (point 3 in Figure 2). However, it may be the case that the blend is not complete, in the sense that this process may have generated an over-generalised term.

After several blends have been computed, an automated evaluation process ranks them according to some optimality principles (Fauconnier and Turner, 2002, Chapter 16). These principles are a necessary aspect of conceptual blending since they allow to filter interesting blends from the (potentially too) many possible ones.⁵ A complete description of optimality principles is out of the scope of this paper and the reader is referred to Goguen and Harrell (2010) for applications of such principles in the *Alloy* algorithm. We give, however, two extreme examples of 'bad blends' for clarifying the importance of using optimality principles in conceptual blending. *Example 1*: Each of the input spaces is a trivial form of a (bad) blend, since no information from the other input spaces is considered. *Example 2*: A blend that includes all properties of the generic space, but no other information of the inputs spaces is a bad blend, since it has the least possible connections with the input spaces. These examples suggest two criteria for ranking the blends; we provide a computational characterisation of them below.

⁵The amalgam-based algorithm produces many blends by following alternative generalisation paths.

Chord transitions description and blending

Individual chord transitions are the ‘atoms’ of the methodology followed herein to construct new transition matrices. Specifically, transition sets from two musical idioms provide input transitions for blending, producing a list of blended transitions that are afterwards embedded in an extended harmonic space. This methodology is described briefly in the next section while some definitions regarding chord transitions follow.

Definition 1. A chord transition c is described by a set of features \mathcal{F} .

In this work a transition is represented by 17 features. Features 1-6 refer to the involved chords. Features 8 to 10 indicate changes during the transitions and are based on the Directed Interval Class (DIC) vector (Cambouropoulos, Katsivalos, and Tsougras, 2013; Cambouropoulos, 2012). Feature 7 accounts for the change that occurred regarding the chords’ root notes. The features considered important in this work are the following:

1. *fromRoot*: the root pitch class of the first chord,
2. *toRoot*: the root pitch class of the second chord,
3. *fromType*: the type of the first chord (GCT base),
4. *toType*: the type of the second chord (GCT base),
5. *fromPCs*: the pitch classes included in the first chord,
6. *toPCs*: the pitch classes included in the second chord,
7. *DICinfo*: the DIC vector of the transition,
8. *DIChas0*: Boolean value indicating whether the DIC of the transition has 0,
9. *DIChas1*: As above but for DIC value 1,
10. *DIChasMinus1*: As above but for DIC value -1 ,
11. *ascSemZero*: Boolean value indicating whether the first chord has the relative pitch class value 11,
12. *descSemZero*: As above but value 1,
13. *semZero*: As above but for value 11 or 1,
14. *ascSemNextRoot*: Boolean value indicating whether the first chord has a pitch class with ascending semitone relation with the pitch class of the second chord’s root,
15. *descSemNextRoot*: As above but with descending semitone,
16. *semNextRoot*: As above but with either ascending or descending semitone, and
17. *5thRootRelation*: Boolean value indicating whether the first chord’s root note is a fifth above of the second’s.

Each feature can be considered as a function that assigns a value to a chord transition c . Features’ values are defined differently depending on the properties they represent. For instance, features 3 to 8 are set-value functions that assign a set of values to a chord. We refer to them as $F_i(c)$. The value of the feature 7 is a vector and we refer to it as $\vec{f}(c)$. Finally, all the other features are binary functions and we refer to them as $f_i(c)$.

From transition blends to transition matrices

In the literature, an effective and common way to describe chord progressions in a music idiom in a statistical manner is by using first-order Markov models (see Kaliakatsos-Papakostas and Cambouropoulos (2014); Simon, Morris, and Basu (2008), among others). Such models reflect the probabilities of each chord following other chords in the idiom, as trained or statistically measured throughout all the pieces in the examined idiom. In this context, individual transitions play an important role on indicating particular characteristics of an idiom.

A convenient way to represent a first order Markov model is through transition matrices, which include one respective row and column for each chord in the examined idiom. The probability value in the i -th row and the j -th column exhibits the probability of the i -th chord going to the j -th —the probabilities of each row sum to unit. The utilised chords are actually represented by chord group exemplars, obtained by the method described in Kaliakatsos-Papakostas et al. (2015), while transitions between chords that pertain to the same chord group are disregarded. The representation of chords is based on the General Chord Type representation (Cambouropoulos, Kaliakatsos-Papakostas, and Tsougras, 2014).

Then, an important question is: *How would a blended idiom be expressed in terms of a transition matrix, provided that the transition matrices of two initial idioms are available?*

The idea examined in the present system is to create an *extended* transition matrix that includes new transitions that allow moving across chords of the initial idioms by potentially using new chords. The examined methodology uses transition blending to create new transitions that: (a) maximally preserve the common parts of transitions between the two initial spaces, and (b) incorporate blended characteristics for creating a smooth ‘morphing’ harmonic effect when moving from chords of one space to chords of the other. An abstract illustration of an extended matrix is given in Figure 4.

By analysing the graphical representation of an extended matrix as depicted in Figure 4 the following facts are highlighted:

1. By using transitions in I_i , only chords of the i -th idiom are used. When using these transitions, the resulting harmonisations preserve the character of idiom i .
2. Transitions in A_{i-j} create direct jumps from chords of the i -th to chords of the j -th idiom. Blended transitions in A_{i-j} can be directly included in the extended matrix.
3. Transitions in B_{i-x} constitute harmonic motions from a chord of idiom i to a newly created chord by blending. Similarly, transitions in the B_{x-j} arrive at chords in idiom j from new chords. For moving from idiom i to idiom j using one external chord c_x that was produced by blending, a chain of two transitions is needed: $c_i \rightarrow c_x$ followed by a transition $c_x \rightarrow c_j$, where c_i in idiom i and c_j in idiom j respectively. A chain of two consecutive transitions with one intermediate external chord from chords of i to chords of j will be denoted as B_{i-x-j} .

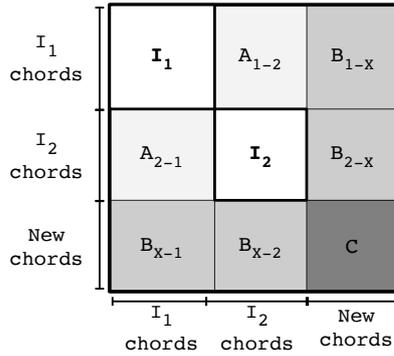


Figure 4: Graphical description of an *extended* matrix that includes transition probabilities of both initial idioms and of several new transitions generated through transition blending. These new transitions allow moving across the initial idioms, creating a new extended idiom that is a superset of the initial ones.

- Sector C transitions are disregarded since they incorporate pairs of chords that exist outside the i -th and j -th idioms, violating our hypothesis for moving from one known chord sets to the other using one new chord at most.

Based on this analysis of the extended matrix, a methodology is proposed for using blends between transitions in I_1 and I_2 . Thereby, transitions in I_1 are blended with those in I_2 and a certain number of best blends is stored for further investigation, creating a pool of best blends. Based on multiple simulations, a large number of the best blends (i.e. 100) in each blending simulation should be inserted in the pool of best blends (\mathcal{B}), so that several commuting scenarios can be created between the initial transition spaces. Thus, a greater number of blends in the pool of best blends introduces a larger number of possible commuting paths in A_{i-j} or in B_{i-x-j} .

Evaluation of transition blending via arguments

By applying the aforementioned blending process a pool of best blends is created that is afterwards used for connecting the transition blocks of two initial idioms, through forming an extended matrix. When a music expert is using the system, she/he is able to select pairs of initial input idioms⁶, choose which aspects of blending are important through arguments (analysed in the following paragraphs) and evaluate/re-adjust this choice by observing the produced results in the extended matrix.

The user evaluates the importance of several transition features by answering questions based on the connecting transitions produced by blending in the extended matrix. The features related to the transitions and their constituent chords are classified into 9 questions (Table 1).

⁶Except from the idioms used in this paper, new idioms can (a) be manually constructed by providing permitted chord transitions and their probabilities, or (b) learned from data using the first order Markov matrix of chord transitions.

Question	Chord Properties	Transition Changes
Q1	<i>fromRoot</i>	
	<i>toRoot</i>	
	<i>fromType</i> <i>toType</i>	
Q2	<i>fromRelPCs</i> <i>toRelPCs</i>	
Q3		<i>DIChas0</i>
Q4		<i>DIChas1</i> <i>DIChasN1</i>
Q5		<i>DIChas2</i> <i>DIChasN2</i>
Q6		<i>DICinfo</i>
Q7		<i>ascSemZero</i> <i>descSemZero</i> <i>semZero</i>
Q8		<i>ascSemNextRoot</i> <i>descSemNextRoot</i> <i>semNextRoot</i>
Q9		<i>5thRootRel</i>

Table 1: Abstraction of chords' and transition changes' features.

- Q1:** Are roots and types of chords important?
- Q2:** Are individual pitch classes of chords important?
- Q3:** Are repeating pitch classes in transitions important?
- Q4:** Are semitone steps in transitions important?
- Q5:** Are tone steps in transitions important?
- Q6:** Are the intervallic contents of transitions important?
- Q7:** Are semitone motions to the tonic important?
- Q8:** Are semitones to the second chord's root important?
- Q9:** Are motions of the chord roots by 5th important?

The first two questions concern characteristics of the chords that constitute the transition (features 1 to 6), while the remaining seven questions concern intervallic changes that occur within the transition (features 7 to 17). Relating questions to transition features was performed with the involvement of music experts, to ensure that the mapping is as accurate and as informative to the user as possible.

We denote the set of questions available to the user as \mathcal{Q} . When a user selects a question, an argument is automatically generated. For the sake of this paper, an argument is defined as follows.

Definition 2. An argument A is a tuple $\langle q, F \rangle$, where $q \in \mathcal{Q}$ and $F \subset \mathcal{F}$.

The user can specify at most 9 arguments, each of them is mapped to a set of properties. The set of user arguments $\{A_1, \dots, A_9\}$ corresponding to answers to \mathcal{Q} will be denoted by \mathcal{A} . We assume to have a function $\psi : \mathcal{A} \rightarrow \mathcal{F}$ that returns the set of chord and transition properties associated with an argument (e.g., for the purposes of the current analysis, Table 1 specifies ψ as a look-up function). The arguments are used to compute two criteria in order to rate a blend: *total association* and *symmetry*.

The total association indicates the total number of properties that a blend inherits from the inputs. A blend with higher input associations is preferable since it is structurally more deeply related with the inputs. The total association is calculated by taking the individual association of a blend w.r.t. the input chord transitions into account. The individual association of a blend b w.r.t. to an input I , denoted as $a(b, I)$, is defined as:

$$a(b, I) = \sum_{A_i \in \mathcal{A}} \text{Val}(A_i, b, I)$$

where $\text{Val} : \mathcal{A} \rightarrow \mathbb{R}$ is a function that takes an argument as input and aggregates the values of the chord and transition change properties associated with the argument, by interpreting them according to some music background knowledge. Depending on the type of argument, Val is defined in different ways.

When an argument refers to the roots and types of chords (A_1), Val is defined as:

$$\text{Val}(A_1, b, I) = \sum_{F_j \in \psi(A_1)} \text{equals}(F_j(I), F_j(b))$$

The value of A_1 is calculated by counting how many properties—among *fromRoot*, *toRoot*, *fromType* and *toType*—are equals between a blend b and an input I . *equals* is a function that returns 1 when two sets are equals and 0 otherwise.

When an argument refers to the individual pitch classes of chords (A_2), Val is defined as:

$$\text{Val}(A_2, b, I) = \sum_{F_j \in \psi(A_2)} |F_j(I) \cap F_j(b)|$$

The value of A_2 is calculated as the number of elements that are common in the set-value properties *fromRelPCs* and *toRelPCs* of a blend b and an input I .

When an argument refers to the intervalic contents of transitions (A_6), Val is defined as:

$$\text{Val}(A_6, b, I) = \text{norm}_{[0,1]}(\rho_{\vec{f}(I), \vec{f}(b)})$$

The value of A_6 is calculated as the Pearson's correlation coefficient of the vector-value property *DICinfo* of a blend b and an input I . Higher correlations in the DIC vectors of two transitions indicate higher resemblance; *norm* is a function that normalises the Pearson's coefficient from the interval $[-1, 1]$ to the interval $[0, 1]$.

For all the other types of arguments, Val is defined as:

$$\text{Val}(A_i, b, I) = \sum_{f_j \in \psi(A_i)} 1 - (f_j(I) - f_j(b))$$

Based on the above definitions, the *total association* value is the sum of the individual associations.

$$\text{assoc}(b) = \sum_{I_i \in \mathcal{I}} a(b, I_i)$$

where \mathcal{I} is the set of input spaces, containing in this specific case, I_1 and I_2 .

Symmetry, on the other hand, reflects the balance of properties that a blend inherits from both input spaces. A blend

has a high symmetry when it inherits an almost equal proportion of properties from both input spaces. Blends having higher symmetry are preferred to those with lower symmetry, since a high symmetry reflects a stronger hybridisation of structural characteristics. Hybridisation is an important principle to evaluate transition blends.

The blend symmetry is defined in terms of its 'asymmetry'. The asymmetry of a blend w.r.t. the inputs, denoted as $\text{asym}(b)$, is calculated as:

$$\left| \frac{a(b, I_1)^2 + a(b, I_1)a(b, I_2)}{a(b, I_1)^2 + a(b, I_2)} - \frac{a(b, I_2)^2 + a(b, I_1)a(b, I_2)}{a(b, I_2)^2 + a(b, I_1)} \right|$$

The value of $\text{asym}(b)$ is defined in $[0, 1]$, where 0 stands for a perfect symmetry (equal association with both inputs) and 1 stands for total asymmetry (association only with one input). Additionally, the non-absolute version of the above equation suggests the prevailing input, with a negative value indicating dominating association of the blend with the first input and a positive value contrarily.

The total rate of a blend is computed by taking the input association and asymmetry values into account.

$$\text{rate}(b) = \frac{\text{assoc}(b)(1 - \text{asym}(b))}{\text{assoc}(b) + (1 - \text{asym}(b))}$$

The above expression promotes pairs of association and symmetry that are both high, while a simple sum would allow a low value of the one to be covered by the other.

Finally, a decision making criterion to compare any pair of blends $b_1, b_2 \in \mathcal{B}$ can be defined as follows.

Definition 3 (Decision criterion). *A blend b_1 is preferred to a blend b_2 if and only if $\text{rate}(b_1) \geq \text{rate}(b_2)$.*

It is worthy to notice that the above criterion guarantees that any pair of blends is comparable, and, consequently, it allows to decide which blends are the best ones. This is an important property for blend evaluation and, generally, for approaches to argumentation-based decision making (Amgoud and Prade, 2009; Bonet and Geffner, 1996).

System evaluation

Referring to Figure 1, via the interface α , the user has access to modules γ , and β which can be used to specify *concepts* that will inform the resulting product, namely, the input idioms and arguments that impose constraints on the generated blend. These are translated by the system into process-friendly formats. Module ϵ embodies the (process-level) concept of a system that make use of the supplied idioms and the blending properties to generate *example* transition matrices, ζ . In the current version of the system, these transitions are evaluated by the user (music expert) in step ι using sophisticated harmonic knowledge that reflects an historically established musical *aesthetic*. The user can then return to the GUI α , and adjust the settings of γ and β to regenerate the transitions.

This is illustrated in Figure 5 in box **P1**, using the diagrammatic extension to the FACE model by Colton et al. (2014). Here, capital letters F , A , C , or E are creative

acts that generate a framing, aesthetic, concept, or example, respectively. Administrative acts S and T denote selection and translation. Lower-case letters denote the generated artefact in each case (e.g., the concept c corresponding to the concept-creation act C). Subscripts p , g , or m indicate whether the act takes place at the process, ground, or meta level. Inside each box, stacks show the dependence in development epochs, and arrows show run-time message passing. Acts taken by the programmer or user are decorated with a bar, whereas acts taken by the system itself receive no extra decoration.

In the current version of the system, apart from the *programmer's* creative acts specifying the modules and their interconnections, and the algorithm \overline{C}_p^ϵ that turns inputs into blends, the user, who is assumed to be a music expert, must intervene in the system in two places.

First, the *user* defines system settings \overline{C}_g^γ , \overline{C}_g^β that correspond to the selection of input idioms and of arguments respectively. Second, after the run completes, he or she evaluates the system output via \overline{A}_g^ϵ .

The *system's* primary responsibilities take place through the creative acts E_g^ϵ , which generate blends, and $S[a_g^{\beta,\gamma}](e^{\epsilon*})$, in which the aesthetic $A_g^{\beta,\gamma}$ (a unified label for assoc and asym, which are defined anew in each run, based on a fixed translation of the user's arguments, as specified in the previous section) is applied to rate the possible blends, and select to a final extended transition matrix.

Therefore, the key idea behind what has been implemented so far is an 'automated ranking/evaluation' step that guides the selection of blends, $S[a_g^{\beta,\gamma}](e^{\epsilon*})$ according to the arguments defined by the user. The development of the programmatic components that operationalise this process has relied on both computer science and musicological insights. This approach has been characterised as meaningful per se through informal feedback provided by musical experts – but is perhaps especially valuable because it constitutes a prototype for more involved automated evaluation of computer-generated harmonic spaces.

Indeed, the next step towards the development of a more autonomously creative system using the same architecture is fairly clear: future work would need to 'close the loop' computationally, connecting the evaluation of generated transition matrices with the parameter-setting (i.e., argumentation) stage, and making this run autonomously to refine the system's behavior. This as-yet hypothetical situation is illustrated in the box **P2**.

Here, the programmer has translated some of the user-specified aesthetics into code $\overline{T}[A_g^\epsilon]$, and invented a meta-level concept \overline{C}_m^α defining a system component that can automatically apply these aesthetics to the generated transition matrices e_g^ζ as in order to automatically generate new system settings C_g^γ , C_g^β .

Conclusion, Discussion and Future Work

In this paper, we described a methodology for harmonic blending and we proposed a creative system that assists musicologists with the evaluation and enhancement of harmonic innovation. We defined some harmonic features of

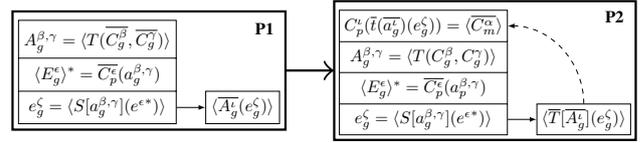


Figure 5: The current implementation **P1** prototypes automated evaluation of blends according to user's arguments; this points to a proposed future implementation **P2** with further automation.

chord transitions utilised for evaluating blends of transitions, leading to the invention of novel harmonic spaces. The system allows a musicologist to specify arguments over these features that are taken into account in the generation of new harmonic spaces. The music expert can then assess whether the new harmonic idiom makes musicological sense and re-adjust the arguments to explore alternative blends that can potentially produce better harmonic spaces.

The main advantage of the current system is the agile interaction through which the user can express desirable properties over the transition blends and their argument-based evaluation in order to produce musically meaningful results. The added value of argumentation is the ranking/evaluation of blended transition—obtained by conceptual blending of two input transition belonging to two musical idioms—by answering questions which abstract several properties of chord transitions. On the other hand, the evaluation of the creative output of the system, i.e., an extended harmonic space that includes blended transitions, is carried out by the user via an introspective argumentative dialogue.

In a future work we intent to use the argumentation-based process for evaluating the blended harmonisations of user defined melodies, i.e., actual music output. Additionally, mapping the properties of the blended idiom or, at a latter stage of a harmonised melody, back to the parameter-setting stage opens an interesting direction for future research and further improvements of the system. The added value of argumentation can be stressed, for instance, by letting the system suggest possible refinements of the initial user arguments, progressively converting part of the introspective user evaluation into a more explicit format. For example, a future version of the system would be based on identifying harmonic features of the input spaces that automatically suggest an 'optimal' set of initial arguments. The current version of the system is an already-usable prototype on the way towards the development of a more autonomous creative system.

Acknowledgments

This work is partially supported by the COINVENT project (FET-Open grant number: 611553).

References

- Amgoud, L., and Prade, H. 2009. Using arguments for making and explaining decisions. *Artificial Intelligence* 173(3-4):413–436.

- Antovic, M. 2009. Musical Metaphors in Serbian and Romani Children: An Empirical Study. *Metaphor and Symbol* 24(3):184–202.
- Antovic, M. 2011. Musical metaphor revisited: Primitives, universals and conceptual blending. *Universals and Conceptual Blending (February 17, 2011)*.
- Bench-Capon, T. J. M., and Dunne, P. E. 2007. Argumentation in artificial intelligence. *Artificial Intelligence* 171(10-15):619–641.
- Bonet, B., and Geffner, H. 1996. Arguing for decisions: A qualitative model of decision making. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI'96)*, 98–105.
- Cambouropoulos, E.; Kaliakatsos-Papakostas, M.; and Tsougras, C. 2014. An idiom-independent representation of chords for computational music analysis and generation. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*, ICMC–SMC 2014.
- Cambouropoulos, E.; Katsiavalos, A.; and Tsougras, C. 2013. Idiom-independent harmonic pattern recognition based on a novel chord transition representation. In *In Proceedings of the 3rd International Workshop on Folk Music Analysis (FMA2013)*, FMA 2013.
- Cambouropoulos, E. 2012. A Directional Interval Class Representation of Chord Transitions. In *Proc. of the 12th International Conference for Music Perception and Cognition, & 8th Conference of the European Society for the Cognitive Sciences of Music*, ICMPC-ESCOM 2012.
- Colton, S.; Pease, A.; Corneli, J.; Cook, M.; and Llano, T. 2014. Assessing progress in building autonomously creative systems. In Ventura, D.; Colton, S.; Lavrac, N.; and Cook, M., eds., *Proceedings of the Fifth International Conference on Computational Creativity*.
- Confalonieri, R.; Schorlemmer, M.; Plaza, E.; Eppe, M.; Kutz, O.; and Peñaloza, R. 2015a. Upward Refinement for Conceptual Blending in Description Logic – An ASP-based Approach and Case Study in EL⁺⁺. In *International Workshop on Ontologies and Logic Programming for Query Answering, International Joint Conference on Artificial Intelligence (IJCAI) 2015*.
- Confalonieri, R.; Corneli, J.; Pease, A.; Plaza, E.; and Schorlemmer, M. 2015b. Using Argumentation to Evaluate Concept Blends in Combinatorial Creativity. In *Proceedings of the 6th International Conference on Computational Creativity, ICCCI5*, 174–181.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321 – 357.
- Eppe, M.; Confalonieri, R.; Maclean, E.; Kaliakatsos-Papakostas, M.; Cambouropoulos, E.; Schorlemmer, M.; Codescu, M.; and Kühnberger, K.-U. 2015a. Computational invention of cadences and chord progressions by conceptual chord-blending. In *International Joint Conference on Artificial Intelligence (IJCAI) 2015*.
- Eppe, M.; Maclean, E.; Confalonieri, R.; Kutz, O.; Schorlemmer, M.; and Plaza, E. 2015b. ASP, Amalgamation, and the Conceptual Blending Workflow. In Calimeri, F.; Ianni, G.; and Truszczynski, M., eds., *Logic Programming and Nonmonotonic Reasoning*, volume 9345 of LNCS. Springer International Publishing. 309–316.
- Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending And The Mind's Hidden Complexities*. Basic Books.
- Goguen, J., and Harrell, D. F. 2010. Style: A Computational and Conceptual Blending-Based Approach. In Argamon, S., and Dubnov, S., eds., *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*. Berlin: Springer. 147–170.
- Goguen, J. 2006. Mathematical Models of Cognitive Space and Time. In Andler, D.; Ogawa, Y.; Okada, M.; and Watanabe, S., eds., *Reasoning and Cognition*, volume 2 of *Interdisciplinary Conference Series on Reasoning Studies*. Keio University Press.
- Kaliakatsos-Papakostas, M., and Cambouropoulos, E. 2014. Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*, ICMC–SMC 2014.
- Kaliakatsos-Papakostas, M.; Zacharakis, A.; Tsougras, C.; and Cambouropoulos, E. 2015. Evaluating the General Chord Type representation in tonal music and organising GCT chord labels in functional chord categories. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2015)*.
- Ontañón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *Proc. of the Int. Conf. on Case Base Reasoning*, volume 6176 of *Lecture Notes in Computer Science*, 257–271. Springer.
- Ritchie, G. D. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.
- Schorlemmer, M.; Smail, A.; Kühnberger, K.-U.; Kutz, O.; Colton, S.; Cambouropoulos, E.; and Pease, A. 2014. Coinvent: Towards a computational concept invention theory. In *5th Int. Conf. on Computational Creativity*.
- Simon, I.; Morris, D.; and Basu, S. 2008. Mysong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08*, 725–734. New York, NY, USA: ACM.
- Zacharakis, A.; Kaliakatsos-Papakostas, M.; and Cambouropoulos, E. 2015. Conceptual blending in music cadences: A formal model and subjective evaluation. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2015)*.
- Zbikowski, L. M. 2002. *Conceptualizing Music: Cognitive Structure, Theory, and Analysis*. Oxford University Press.

A Process Model for Concept Invention

Roberto Confalonieri, Enric Plaza, Marco Schorlemmer

Artificial Intelligence Research Institute, IIIA-CSIC
Campus de la Universitat Autònoma de Barcelona (UAB)
E-08193 Bellaterra (Barcelona), Catalonia, Spain
{confalonieri, enric, marco}@iia.csic.es

Abstract

In this paper, we propose a computational framework that models concept invention. The framework is based on conceptual blending, a cognitive theory that models human creativity and explains how new concepts are created. Apart from the blending mechanism modeling the creation of new concepts, the framework considers two extra dimensions such as origin and destination. For the former, we describe how a Rich Background supports the discovery of input concepts to be blended. For the latter, we show how arguments, promoting or demoting the values of an audience, to which the invention is headed, can be used to evaluate the candidate blends created. Throughout the paper, we exemplify the computational framework in the domain of computer icons.

Introduction

The cognitive theory of conceptual blending by Fauconnier and Turner (2002) models human creativity as a mental process according to which two input (mental) spaces are combined into a new mental space, called a blend. This theory, which was developed in the context of cognitive linguistics, posits that input mental spaces are somehow packaged by humans with the relevant information in the context in which the blend is created, and that blends are evaluated against some optimality principles (Fauconnier and Turner, 2002).

Existing computational models for concept invention — see the Related Work section for an overview — especially focus on the core mechanism of blending, that is, how blends are created, and re-interpret the optimality principles to evaluate the blends. In this position paper, we claim that a computational model also need to deal with two extra dimensions to which we refer as the *origin* and *destination* of concept invention. The origin considers from where and how input spaces are selected, whereas the destination considers to whom the creation is headed. These dimensions are justifiable if we think that there is no creation *ex nihilo* — thus, there is an origin — and there is usually a *purpose* in creating something new, and, consequently, there is a destination.

To this end, in this paper we propose to model concept invention by means of a process that consists of different sub-processes and components (Figure 1):

- **Rich Background and Discovery:** The origin consists of a Rich Background, the set of concepts available to

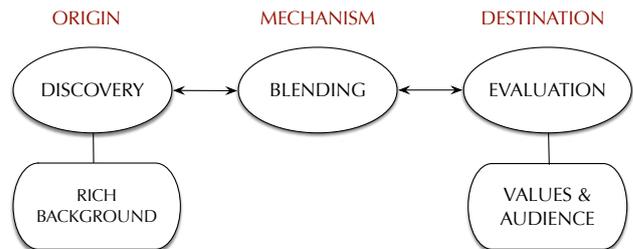


Figure 1: A process model for concept invention.

be blended. This set is finite but complex, diverse, poly-mathic and heterogeneous. Concepts are associated with a background, understood as a person’s education, experience, and social circumstances. The Rich Background supports a discovery process that finds pairs of concepts that can be blended.

- **Blending:** Conceptual blending is the mechanism according to which two concepts are combined into a blended concept. Blending is here characterised in terms of amalgams, a notion that was developed for combining cases in case-based reasoning (Ontañón and Plaza, 2010). Conceptual blending is modeled in terms of an amalgam-based workflow. The blending of two concepts may result in a large number of blends, that need to be evaluated.
- **Arguments, Values, Audiences and Evaluation:** Values are properties expected from a good blend. Values are considered as points of view and can be of different kinds, e.g., moral, aesthetic, etc. A destination or audience is characterised by a preference relation over these values. Arguments in favor or against a blend are built to evaluate the generated blends. An argument can promote or demote a value. In this way, the blends are evaluated depending on the audience for which they are created.

The above process model can be made more concrete in a domain such as *computer icon design*. In such a case, the Rich Background is what we can learn from, program about, specify of computer icons, such as a semiotic model of shapes, signs and relations between signs. This is understood as a finite and specific number of concepts given a particular set of icons (an icon library or a collection of libraries). Values, on the other hand, can be aesthetics such as simplicity or ambiguity, that matter for a specific type of

audience. These values serve to identify good icons that are created by the blending mechanism.

In the next section, we capture the process model above in terms of feature terms. This computational model is exemplified by means of a running example that shows the main processes that undergo the concept invention of new icons.

Related Work

Several approaches of formal and computational models for concept invention, inspired by the work of Fauconnier and Turner (2002), have been proposed.

Amalgam-based conceptual blending algorithms have been developed to blend CASL theories and \mathcal{EL}^{++} concepts in (Confalonieri et al., 2015b; Eppe et al., 2015a,b). In these works, input spaces are assumed to be given. Good blends are selected by re-interpreting some optimality principles.

The Alloy algorithm for conceptual blending by Goguen and Harrell (2005) is based on the theory of algebraic semiotics (Goguen, 1999). Alloy has been integrated in the Griot system for automated narrative generation (Goguen and Harrell, 2005; Harrell, 2005, 2007). The input spaces of the Alloy algorithm are theories defined in the algebraic specification language OBJ (Malcolm, 2000). In the algorithm, input spaces are assumed to be given, hence there is no discovery. The optimality principles by Fauconnier and Turner (2002) are re-interpreted as *structural* optimality principles, and serve to prune the space of possible blends.

Sapper was originally developed by Veale and Keane (1997) as a computational model of metaphor and analogy. It computes a mapping between two separate domains — understood as graphs of concepts — that respects the relational structure between the concepts in each domain. Sapper can be seen as a computational model for conceptual blending, because the pairs of concepts that constitute its output can be manipulated as atomic units, as blended concepts (Veale and Donoghue, 2000). Strictly speaking, Sapper does not work with *a priori* given input spaces. It is the structure mapping algorithm itself which determines the set of concepts and relations between these concepts. In Sapper, most of the optimality principles are captured and serve to rank and filter the correspondences that comprise the mappings computed by the algorithm.

Divago, by Pereira (2007), is probably the first complete implementation of conceptual blending. The Divago’s architecture includes different modules. A knowledge base contains different micro-theories and their instantiations. Of these, two are selected for the blending by the user or randomly, thus, no discovery is taken into account. A mapper then generates the generic space between the inputs, and passes it to a blender module which generates the ‘blendoid’, i.e., a projection that defines the space of possible blends. A factory component is used to select the best blends among the blendoid by means of a genetic algorithm. A dedicated module implements the optimality principles. Given a blend, this module computes a measure for each principle. These measures yield a preference value of the blend that is taken as the fitness value of the genetic algorithm.

Finally, another work that relates to ours is (Confalonieri et al., 2015a). The authors use Lakatosian reasoning to

model dialogues in which users engage to discuss the intended meaning of an invented concept. The main difference with the current work relies on the way in which arguments are generated and used. Here, an argument is a reason for choosing a blend and it is generated automatically, whereas, in (Confalonieri et al., 2015a), an argument is a reason to refine the meaning of a blend and is provided by the user.

Computational Model

Rich Background

Let the Rich Background be a collection of computer icons. We assume that computer icons are described in terms of *form* and a *meaning*. The form consists of a finite set of signs which are related by spatial relationships. Figure 2b(I) shows an example of an icon in which two signs, a MAGNIFYINGGLASS and a HARDDISK, are related by relation *on*. The meaning, on the other hand, is the interpretation that is given to an icon. For instance, a possible meaning associated to the icon in Figure 2b(I) is SEARCH-HARDDRIVE. We allow a sign to have different interpretations depending on the icons in which it is used.

We shall model the Rich Background by means of a finite set \mathcal{C} of feature terms (Carpenter, 1992; Smolka and Ait-Kaci, 1989), each representing a concept. In this paper, feature terms are defined over a signature $\Sigma = \langle \mathcal{S}, \mathcal{F}, \leq, \mathcal{X} \rangle$, where \mathcal{S} is finite set of sort symbols, including \top and \perp , which represent the most specific and the most general sort, respectively; \mathcal{F} is a finite set of feature symbols; \leq is an order relation inducing an inheritance hierarchy such that $\perp \leq s \leq \top$, for all $s \in \mathcal{S}$; and \mathcal{X} is a denumerable set of variables. Then, a feature term ψ has the form:

$$\psi := x : s[f_1 = \Psi_1, \dots, f_n = \Psi_n]$$

with $n \geq 0$, and where $x \in \mathcal{X}$ is called the root variable of ψ (denoted as $\text{root}(\psi)$), $s \in \mathcal{S}$ is the sort of x (denoted as $\text{sort}(x)$), and, for all j with $1 \leq j \leq n$, $f_j \in \mathcal{F}$ are the features of x (denoted as $\text{features}(x)$) and the values Ψ_j of the features are finite, non-empty sets of feature terms and/or variables (provided they are root variables of feature terms occurring in ψ). When the set of values of a feature is a singleton set, we will omit the curly brackets in our notation. We will write $\text{vars}(\psi)$ to denote the set of variables occurring in a feature term ψ .

We choose to model icons as concepts represented by feature terms over the signature with the following sort hierarchy \mathcal{S} :¹

```

ICON
SIGN < {ARROW, MAGNIFYINGGLASS, DOCUMENT,
        PEN, HARDDISK, CLOUD}
MEANING < {ACTION, OBJECTTYPE}
ACTION < {MODIFY, VIEWSEARCH, TRANSFER}
MODIFY < {EDIT, WRITE}
VIEWSEARCH < {SEARCH, FIND, ANALYSE}
TRANSFER < {UPLOAD, DOWNLOAD}
OBJECTTYPE < {INFOCONTAINER, DATACONTAINER}
INFOCONTAINER < {PAGE, DOC, FILE}
DATACONTAINER < {HARDDRIVE, CLOUD}

```

¹The notation $s < \{s_1, \dots, s_n\}$ denotes that s_1, \dots, s_n are sub-sorts of s .

$$x_1 : \text{ICON} \left[\begin{array}{l} \text{form} = \left\{ \begin{array}{l} x_2 : \text{MAGNIFYINGGLASS} \left[\begin{array}{l} \text{action} = x_4 \\ \text{on} = x_3 \end{array} \right] \\ x_3 : \text{HARDDISK} \left[\text{objectType} = x_5 \right] \end{array} \right\} \\ \text{meaning} = \left\{ \begin{array}{l} x_4 : \text{SEARCH} \\ x_5 : \text{HARDDRIVE} \end{array} \right\} \end{array} \right]$$

(a) Feature term representation of a computer icon.

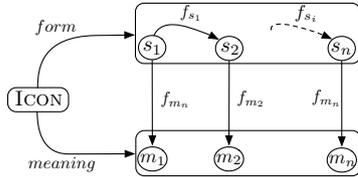


(b) Examples of computer icons.

Figure 2: Rich Background about computer icons.

and features $\mathcal{F} = \{form, meaning, on, below, left, right, action, objectType\}$.

In addition, feature terms representing icons need to be of the following form. A representation of the structure of an icon is presented below and its description follows.



Root variables are of sort ICON and have at most two features $form$ and $meaning$, modelling the signs (s_1, \dots, s_n) and the meaning (m_1, \dots, m_n) of these signs in the context of the icon. Each sign is again represented by means of a feature term whose root variable is of sort $s \geq \text{SIGN}$, and each meaning by means of feature terms whose root variable is of sort $s \geq \text{MEANING}$.

Features of sign terms $(f_{s_1}, \dots, f_{s_n})$ in the schema above are at most one of $on, left, right,$ or $below$, specifying the spatial relationship between signs; and at most one of $action$ or $objectType$, specifying the meaning of signs $(f_{m_1}, \dots, f_{m_n})$ in the schema above). The values of spatial relation features are root variables of feature terms that are in the value of the $form$ feature; and those of features $action$ and $objectType$ are root variables of feature terms that are in the value of the $meaning$ feature. In addition the root variables in the value of the $action$ feature are of sort $s \geq \text{ACTION}$, while those of the $objectType$ feature are of sort $s \geq \text{OBJECTTYPE}$. Figure 2a shows the feature term representation of the icon in Figure 2b(I).

A fundamental relation between feature terms is that of subsumption (\sqsubseteq). Intuitively, a feature term ψ_1 subsumes another one ψ_2 , or ψ_1 is more general than ψ_2 , denoted as $\psi_1 \sqsubseteq \psi_2$, if all the information in ψ_1 is also in ψ_2 .² We omit the formal definition of subsumption, which can be found in (Ontańón and Plaza, 2012) for feature terms as represented

²Notice that, in Description Logics, $A \sqsubseteq B$ has the inverse meaning “A is subsumed by B”, since subsumption is defined from the set inclusion of the interpretations of A and B.

in this paper. The subsumption relation induces a partial order on the set of all features terms \mathcal{L} over a given signature, that is, $\langle \mathcal{L}, \sqsubseteq \rangle$ is a poset.

Discovery

In cognitive theories of conceptual blending, input spaces to be blended are givens that represent how humans package some relevant information in the context in which the blend is created.

In our computational model, an input space is a concept belonging to a library of concepts. The packaging of some relevant information corresponds to a discovery process that takes certain properties, which the blends need to satisfy, into account. In the creation of computer icons, we can imagine that an icon designer knows the meaning of an icon he wishes to create, but he ignores its form.

The discovery takes a query over the meaning of an icon concept as input, looks for concepts in the Rich Background, and returns an ordered set of pairs of concepts that can be blended. The query is modeled as a feature term ψ_q in which only the meaning part of an icon is specified. For instance, a query asking for an icon with meaning SEARCH-DOC is modeled as:

$$\psi_q := x_1 : \text{ICON} \left[\text{meaning} = \left\{ \begin{array}{l} x_2 : \text{SEARCH} \\ x_3 : \text{DOC} \end{array} \right\} \right] \quad (1)$$

The matching of the query is not always a perfect match, since icon concepts in the Rich Background can have only one part of the meaning or similar meanings w.r.t. the meaning searched. To this end, the query resolution is modeled as a *similarity-based search*.

The main idea behind the similarity-based search is that, for each icon concept ψ_i in the Rich Background, we measure how ψ_q and ψ_i are similar and, we use this measure to rank the results. The similarity between two feature terms can be defined by means of their *anti-unification* or *Least General Generalisation* (LGG) (Ontańón and Plaza, 2012).

Definition 1 (Least General Generalisation) *The least general generalisation of two feature terms ψ_1 and ψ_2 , denoted as $\psi_1 \sqcap \psi_2$, is defined as the most specific term that subsumes both: $\psi_1 \sqcap \psi_2 = \{\psi \mid \psi \sqsubseteq \psi_1 \wedge \psi \sqsubseteq \psi_2 \wedge \nexists \psi' : \psi \sqsubset \psi' \wedge \psi' \sqsubseteq \psi_1 \wedge \psi' \sqsubseteq \psi_2\}$.*

The least general generalisation encapsulates all the information that is common to both ψ_1 and ψ_2 and, for this reason, is relevant for defining a similarity measure.

The least general generalisation can be characterised as an operation over a refinement graph of feature terms. The refinement graph is derived from the poset $\langle \mathcal{L}, \sqsubseteq \rangle$ by means of a generalisation refinement operator γ .

$$\gamma(\psi) = \{\psi' \in \mathcal{L} \mid \psi' \sqsubseteq \psi \text{ and } \nexists \psi'' \text{ s.t. } \psi' \sqsubset \psi'' \sqsubset \psi\}$$

The above definition essentially says that γ is an operation that generalises a feature term to a set of feature terms that is an anti-chain. The refinement graph, then, is a directed graph whose nodes are feature terms, and for which there is an edge from feature term ψ_1 to ψ_2 , whenever $\psi_2 \in \gamma(\psi_1)$. We shall call *generalisation paths* all finite paths $\psi \xrightarrow{\gamma} \psi'$ in a refinement graph, and denote with $\lambda(\psi \xrightarrow{\gamma} \psi')$ its length.

Ontañón and Plaza (2012) describe a generalisation operator for feature terms that consist of:

Sort generalisation, which generalises a term by substituting the sort of one of its variables by a more general sort;

Variable elimination, which generalises a term by removing the value of one of the features in one variables of the term (a variable is removed only when the variable does not have any features);

Variable equality elimination, which generalises a term by removing a variable equality and ensuring that \perp can be reached from any term.

We refer to (Ontañón and Plaza, 2012) for the formal details of the operator.

It is worthy noticing that, in case of variable equalities, it is not possible to define a generalisation operator that finds all possible generalisations of a feature term. However, for the purpose of defining a least general generalisation-based similarity, an operator which ensures that \perp is reachable in a finite number of steps will suffice.

Example 1 (LGG example) Let us consider the feature terms ψ_q in Eq. 1 and ψ_1 in Figure 2a. The LGG $\psi_q \sqcap \psi_1$ is:

$$x_1 : \text{ICON} \left[\text{meaning} = \left\{ \begin{array}{l} x_2 = \text{SEARCH} \\ x_3 = \text{OBJECTTYPE} \end{array} \right\} \right]$$

$\psi_q \sqcap \psi_1$ captures the information shared among the icon concept ψ_1 and the query ψ_q . Both of them have two meanings. According to the ontology previously defined, the most general sorts for variables x_2 and x_3 are SEARCH and OBJECTTYPE respectively. The form feature of ψ_1 is removed, since ψ_q does not contain this information.

As previously said, the least general generalisation of two feature terms $\psi_1 \sqcap \psi_2$ is a symbolic representation of the information shared by ψ_1 and ψ_2 . It can be used to measure the similarity between feature terms in a quantitative way. The refinement graph allows us to estimate the quantity of information of any feature term ψ . It is the length of the (minimal) generalisation path that leads from ψ to the most general term \perp . Therefore, the length $\lambda(\psi_1 \sqcap \psi_2 \xrightarrow{\gamma} \perp)$ estimates the informational content that is common to ψ_1 and ψ_2 . In order to define a similarity measure, we need to

compare what is common to ψ_1 and ψ_2 with what is not common. To this end, we take the lengths $\lambda(\psi_1 \xrightarrow{\gamma} \psi_1 \sqcap \psi_2)$ and $\lambda(\psi_2 \xrightarrow{\gamma} \psi_1 \sqcap \psi_2)$ into account. Then a similarity measure can be defined as follows.

Definition 2 (LGG-based similarity) The LGG-based similarity between two feature terms ψ_1 and ψ_2 , denoted by $S_\lambda(\psi_1, \psi_2)$, is:

$$\lambda(\psi_1 \sqcap \psi_2 \xrightarrow{\gamma} \perp)$$

$$\lambda(\psi_1 \sqcap \psi_2 \xrightarrow{\gamma} \perp) + \lambda(\psi_1 \xrightarrow{\gamma} \psi_1 \sqcap \psi_2) + \lambda(\psi_2 \xrightarrow{\gamma} \psi_1 \sqcap \psi_2)$$

The measure S_λ estimates the ratio between the amount of information that is shared and the total information content. From a computational point of view, S_λ requires to compute two things. The LGG and the three lengths defined in the above equation. The algorithms for computing S_λ can be found in (Ontañón and Plaza, 2012).

Example 2 (Similarity example) Let us consider the feature terms ψ_q in Eq. 1, ψ_1 in Figure 2a and their LGG in Example 1. Lengths $\lambda_1 = \lambda(\psi_1 \sqcap \psi_q \xrightarrow{\gamma} \perp) = 8$, $\lambda_2 = \lambda(\psi_1 \xrightarrow{\gamma} \psi_1 \sqcap \psi_q) = 12$, and $\lambda_3 = \lambda(\psi_q \xrightarrow{\gamma} \psi_1 \sqcap \psi_q) = 2$. Notice that λ_3 is very small (2 generalisations), while λ_2 is larger since ψ_1 has more generalised content. Therefore, the similarity between ψ_q and ψ_1 is:

$$S_\lambda(\psi_1, \psi_q) = \frac{8}{12 + 2 + 8} = 0.36$$

$S_\lambda(\psi_1, \psi_q)$ expresses that these two concepts share the 36% of the total information.

Given the above definitions, the discovery of concepts can be implemented by a discovery algorithm. The algorithm accepts a Rich Background of concepts \mathcal{C} , a query ψ_q , and the generalisation operator γ as input, and returns a ranked set of pairs of concepts. This ranking can be done according to different strategies. One way is to build all pairs of concepts and rank them in a lexicographical order. The discovery returns a set of pairs of concepts $\langle (\psi_j, \lambda_j), (\psi_{j+1}, \lambda_{j+1}) \rangle$ in which $\lambda_j \geq \lambda_{j+1}$.

Blending

The computational model of concept blending is based on the notion of *amalgams* (Ontañón and Plaza, 2010). This notion was proposed in the context of case-based reasoning. Amalgams have also been used to model analogy (Besold and Plaza, 2015). According to this approach, input concepts are generalised until a generic space is found, and pairs of generalised input concepts are ‘unified’ to create blends.

Formally, the notion of amalgams can be defined in any representation language \mathcal{L} for which a subsumption relation \sqsubseteq between formulas (or descriptions) of \mathcal{L} can be defined, together with an anti-unifier operation—playing the role of the generic space—and a unifier operation. Therefore, it can be defined for feature terms. We already defined the anti-unification of two feature term descriptions (Definition 1). Now, we proceed to define their unification.

Definition 3 (Unification) The unification of two feature terms ψ_1 and ψ_2 , denoted as $\psi_1 \sqcup \psi_2$, is defined as the most general term that is subsumed by both: $\psi_1 \sqcup \psi_2 = \{\psi \mid \psi_1 \sqsubseteq \psi \wedge \psi_2 \sqsubseteq \psi \wedge \nexists \psi' : \psi' \sqsubset \psi \wedge \psi_1 \sqsubseteq \psi' \wedge \psi_2 \sqsubseteq \psi'\}$.

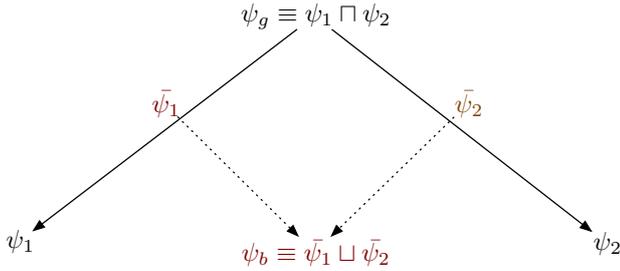


Figure 3: A diagram of a blend ψ_b from inputs ψ_1 and ψ_2 .

Intuitively, a unifier is a description that has all the information in both the original descriptions. If joining this information leads to inconsistency, this is equivalent to say that $\psi_1 \sqcup \psi_2 = \top$, e.g., they have no common specialisation except ‘none’.

An *amalgam* or *blend* of two descriptions is a new description that contains *parts from these two descriptions*. For instance, an amalgam of ‘a red French sedan’ and ‘a blue German minivan’ is ‘a red German sedan’; clearly, there are always multiple possibilities for amalgams, like ‘a blue French minivan’.

For the purposes of this paper, we define an *amalgam* or *blend* of two input descriptions as follows:

Definition 4 (Blend) A description $\psi_b \in \mathcal{L}$ is a blend of two inputs ψ_1 and ψ_2 (with LGG $\psi_g = \psi_1 \sqcup \psi_2$) if there exist two generalisations $\bar{\psi}_1$ and $\bar{\psi}_2$ such that: 1) $\psi_g \sqsubseteq \bar{\psi}_1 \sqsubseteq \psi_1$, 2) $\psi_g \sqsubseteq \bar{\psi}_2 \sqsubseteq \psi_2$, and 3) $\psi_b \equiv \bar{\psi}_1 \sqcap \bar{\psi}_2 \neq \top$.

The above definition is illustrated in Figure 3, where the LGG of the inputs is indicated as ψ_g , and the blend ψ_b is the unification of two concrete generalisations $\bar{\psi}_1$ and $\bar{\psi}_2$ of the inputs. Equality (\equiv) here should be understood as \sqsubseteq -equivalence, that is, $\psi_1 \equiv \psi_2$ iff $\psi_1 \sqsubseteq \psi_2$ and $\psi_2 \sqsubseteq \psi_1$.

Usually one is interested only in maximal blends, e.g., in those blends that contain the maximal information of their inputs. A blend ψ_b of two inputs ψ_1 and ψ_2 is maximal if there is no other blend ψ'_b of ψ_1 and ψ_2 such that $\psi_b \sqsubset \psi'_b$. The reason why one is interested in maximal blends is that a maximal blend captures as much information as possible from the inputs. Moreover, any non-maximal blend can be obtained by generalising a maximal blend.

However, the number of blends that satisfies the above definition can still be very large and selection criteria for filtering and ordering them are therefore needed. Fauconnier and Turner (2002) discuss optimality principles, however, these principles are difficult to capture in a computational way, and other selection strategies need to be explored.

We interpret blend evaluation in two steps. First, we discard those blends that do not satisfy a query ψ_q . Then, we order the blends by means of arguments, values and audiences in order to decide which blend is the best one.

Arguments, Values and Audiences

An argument is a central notion in several models for reasoning about defeasible information (Dung, 1995; Pollock, 1992), decision making (Amgoud and Prade, 2009; Bonet

and Geffner, 1996), practical reasoning (Atkinson, Bench-Capon, and McBurney, 2004), and modeling different types of dialogues such as persuasion (Bench-Capon, 2003). In most existing works on argumentation, an argument is a reason for believing a statement, choosing an option, or doing an action. Depending on the application domain, an argument is either considered as an abstract entity whose origin and structure are not defined, or it is a logical proof for a statement where the proof is built from a knowledge base.

In our model, arguments are reasons for accepting or rejecting a given blend. They are built by the agent when calculating the different values associated with a blend. Values are considered as points of view and can have different origins, e.g., they can be moral, aesthetic, etc.

Generally, there can be several values $\mathcal{V} = \{v_1, \dots, v_k\}$. Each value is associated with a degree that belongs to the scale $\Delta = (0, \dots, 1]$, where 0 and 1 are considered the worst and the best degree respectively. For our purposes, we will consider values such as *simplicity* and *unambiguity*.

The main idea behind simplicity is that we want to estimate how simple an icon is from a representation point of view. This can be done by counting the quantity of information used in the feature term describing an icon. We can assume that simple icons are those described with less information. Therefore, simplicity is defined to be inversely proportional to the total number of features and sorts used in the variables of a feature term ψ_b .

$$\text{Simplicity}(\psi_b) = \frac{1}{\sum_{x \in \text{vars}(\psi_b)} \text{features}(x) + \text{sorts}(x)}$$

Unambiguity, on the other hand, measures how many interpretations an icon has w.r.t. the Rich Background. Since icons are polysemic—they can be interpreted in different ways—there can be icons that contain the same sign but the sign is associated with a different meaning. To define the unambiguity value, let us first define the polysemic set of ψ_b as:

$$\text{Pol}(\psi_b) = \{\psi_j \in \mathcal{C} \mid \exists s \in \text{form}(\psi_j) \cap \text{form}(\psi_b) \wedge \text{meaning}(\psi_j, s) \neq \text{meaning}(\psi_b, s)\}$$

where $\text{form}(\psi_j)$ is a function that returns the value of feature *form*, i.e., the set of signs used in the icon represented by feature term ψ_j ; and $\text{meaning}(\psi_j, s)$ is a function that returns the sort of the variable that is the value of feature *action* or *objectType* of the variable of sort s , i.e., the meaning used for the sign represented by sort s in feature term ψ_j . Then, the unambiguity value is defined to be inversely proportional to the cardinality of Pol .

$$\text{Unambiguity}(\psi_b) = \begin{cases} 1/|\text{Pol}(\psi_b)| & \text{if } |\text{Pol}(\psi_b)| \neq 0 \\ 1 & \text{otherwise} \end{cases}$$

Values play a different role depending on the target or audience towards which the creation is headed. Audiences are characterised by the values and by a preferences among these values. Given a set of values \mathcal{V} , there are potentially as many audiences as there are orderings on \mathcal{V} .

Definition 5 (Audience) An audience is a binary relation $\mathcal{R} \subseteq \mathcal{V} \times \mathcal{V}$ which is irreflexive, asymmetric, and transitive.

We say that v_i is preferred to v_j in the audience \mathcal{R} , denoted as $v_i \succ_{\mathcal{R}} v_j$, if $\langle v_i, v_j \rangle \in \mathcal{R}$. We say that a value v_j covers v_i in the audience \mathcal{R} , denoted as $v_i \succ_{\mathcal{R}} v_j$, if $v_i \succ_{\mathcal{R}} v_j$ and $\nexists v_i'$ such that $v_i \succ_{\mathcal{R}} v_i' \succ_{\mathcal{R}} v_j$.

Given a blend, an argument is generated for each value. The degree of the value characterises the ‘polarity’ of the argument which can be *pro* or *con* a blend. Arguments pro promote a blend whereas arguments cons demote it. Given a set of blends \mathcal{B} , the tuple $\langle \mathcal{B}, \mathcal{V}, \Delta \rangle$ will be called a theory.

Definition 6 (Argument) Let $\langle \mathcal{B}, \mathcal{V}, \Delta \rangle$ be a theory.

- An argument pro a blend b is a tuple $\langle (v, \delta), b \rangle$ where $v \in \mathcal{V}$, $\delta \in \Delta$ and $0.5 \leq \delta \leq 1$
- An argument con b is a pair $\langle (v, \delta), b \rangle$ where $v \in \mathcal{V}$, $\delta \in \Delta$ and $0 < \delta < 0.5$

A function Val returns the value v associated with an argument and a function Deg returns δ .

The blend evaluation can be formulated as a decision problem in which one has to decide an order relation $\geq_{\mathcal{B}}$ on the set of candidate blends \mathcal{B} . The definition of this relation is based on the set of arguments pros and cons associated with the candidate blends. Depending on the kind of arguments that are considered and how they are handled, different decision criteria can be defined (Amgoud and Prade, 2009):

- **Unipolar decision criteria:** they focus either only on arguments pros or arguments cons;
- **Bipolar decision criteria:** they take both arguments pros and cons into account;
- **Meta-criteria:** they aggregate arguments pros and cons into a meta-argument.

In what follows, we denote the set of arguments pros and cons as $\mathcal{A}_p = \{\alpha_1, \dots, \alpha_n\}$ and $\mathcal{A}_c = \{\alpha_1, \dots, \alpha_m\}$ respectively. Besides, we assume to have the following functions: $\mathcal{M}_p : \mathcal{B} \rightarrow 2^{\mathcal{A}_p}$ and $\mathcal{M}_c : \mathcal{B} \rightarrow 2^{\mathcal{A}_c}$ that return the set of arguments pros and the set of arguments cons associated with a blend respectively; $\mathcal{M} : \mathcal{B} \rightarrow 2^{\mathcal{A}_p \cup \mathcal{A}_c}$ that returns all arguments associated with a blend.

A basic decision criterion for comparing candidate blends can be defined by comparing the number of arguments pros associated with them.

Definition 7 Let $b_1, b_2 \in \mathcal{B}$. $b_1 \geq_{\mathcal{B}} b_2$ if and only if $|\mathcal{M}_p(b_1)| \geq |\mathcal{M}_p(b_2)|$.

Notice that the above criterion guarantees that any pair of blends can be compared.

When the audience is taken into account, one may think of preferring a blend that has an argument pro whose value is preferred to the values of any argument pro the other blends.

Definition 8 Let $b_1, b_2 \in \mathcal{B}$. $b_1 \geq_{\mathcal{B}} b_2$ if and only if $\exists \alpha \in \mathcal{M}_p(b_1)$ such that $\forall \alpha' \in \mathcal{M}_p(b_2)$, $\text{Val}(\alpha) \succ_{\mathcal{R}} \text{Val}(\alpha')$.

In the above definition, $\geq_{\mathcal{B}}$ depends on the relation $\succ_{\mathcal{R}}$. Since $\succ_{\mathcal{R}}$ is a preference relation, some of the values of the arguments can be incomparable. Consequently, b_1 and b_2 will not be comparable neither. This definition can be relaxed, for instance, by ignoring these arguments.

The counter-part decision criteria of Definitions 7-8 for the case of arguments cons can be defined in a similar way and we omit them.

In the case of bipolar decision criteria, we can combine the criterion dealing with arguments pros with the criterion dealing with arguments cons.

Definition 9 Let $b_1, b_2 \in \mathcal{B}$. $b_1 \geq_{\mathcal{B}} b_2$ if and only if $|\mathcal{M}_p(b_1)| \geq |\mathcal{M}_p(b_2)|$ and $|\mathcal{M}_c(b_1)| \leq |\mathcal{M}_c(b_2)|$.

Unfortunately, the above definition does not ensure that we can compare all the blends.

Finally, meta-criteria for deciding which blends are preferred can be defined by aggregating arguments pros and cons into a meta-argument. Then, comparing two blends amounts to compare the resulting meta-arguments. A simple criterion can be defined by aggregating the degrees of the arguments associated with a blend.

Definition 10 Let $b_1, b_2 \in \mathcal{B}$. $b_1 \geq_{\mathcal{B}} b_2$ if and only if

$$\sum_{\alpha \in \mathcal{M}(b_1)} \text{Deg}(\alpha) \geq \sum_{\alpha' \in \mathcal{M}(b_2)} \text{Deg}(\alpha')$$

This definition can be extended to take the audience into account. To this end, we consider a rank function that maps each value of \mathcal{R} to an integer. The rank function is defined as follows:

$$\text{Rank}_{\mathcal{R}}(v) = \begin{cases} 1 & \text{if } \nexists v' \text{ s.t. } v' \succ_{\mathcal{R}} v \\ \max_{v' \succ_{\mathcal{R}} v} \{\text{Rank}_{\mathcal{R}}(v')\} + 1 & \text{otherwise} \end{cases}$$

Essentially, Rank counts how many values a certain value covers. This ranking is then used to define the following audience-based aggregation decision criterion.

Definition 11 Let $b_1, b_2 \in \mathcal{B}$. $b_1 \geq_{\mathcal{B}} b_2$ if and only if

$$\sum_{\alpha \in \mathcal{M}(b_1)} \frac{\text{Deg}(\alpha)}{\text{Rank}_{\mathcal{R}}(\text{Val}(\alpha))} \geq \sum_{\alpha' \in \mathcal{M}(b_2)} \frac{\text{Deg}(\alpha')}{\text{Rank}_{\mathcal{R}}(\text{Val}(\alpha'))}$$

This definition also guarantees that all the blends are comparable.

The Model at Work

Let us imagine an agent that has access to a Rich Background $\mathcal{C} = \{\psi_1, \psi_2, \psi_3, \psi_4\}$ consisting of four of the icons depicted in Figures 2b(I-II-III-IV). As previously described, ψ_1 is a feature term representing an icon with meaning SEARCH-HARDDISK. ψ_2 represents an icon that consists of two sorts of type sign, an ARROW and a CLOUD, whose meaning is DOWNLOAD-CLOUD. ψ_3 represents an icon with two sorts of type sign, a PEN and a DOCUMENT, whose meaning is EDIT-DOC; finally, ψ_4 is a feature term that consists of three sorts, ARROW, DOCUMENT and CLOUD with the intended meaning of DOWNLOAD-DOC-CLOUD.

The agent receives as input a query asking for an icon with meaning SEARCH-DOC, ψ_q (Eq. 1), and an audience, that is, a preference order over the values. For the sake of this example, we assume that Simplicity $\succ_{\mathcal{R}}$ Unambiguity.

The discovery retrieves the following pairs of concepts:

$$\{ \langle (\psi_1, 0.36), (\psi_3, 0.36) \rangle \}, \{ \langle (\psi_1, 0.36), (\psi_2, 0.27) \rangle \}$$

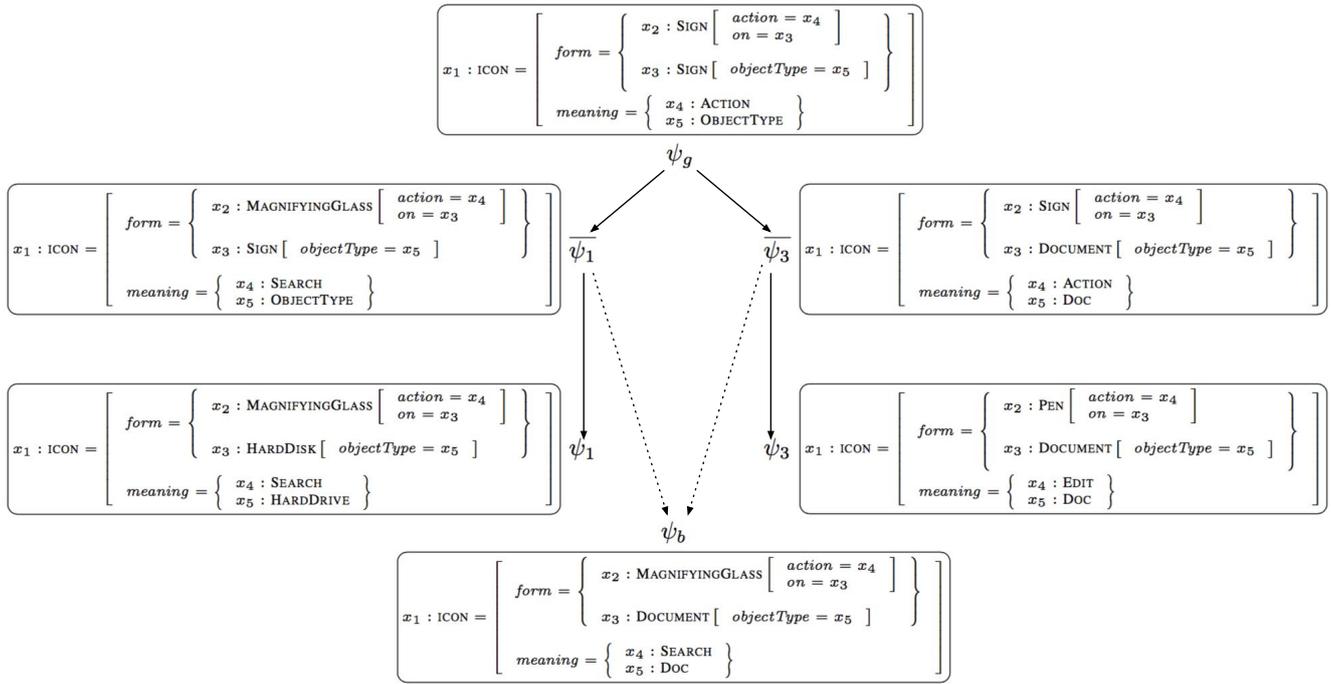


Figure 4: Amalgam-based blending of feature terms ψ_1 and ψ_3 .

$$\{ \langle (\psi_3, 0.36), (\psi_2, 0.27) \rangle \}, \{ \langle (\psi_1, 0.36), (\psi_4, 0.25) \rangle \}$$

$$\{ \langle (\psi_3, 0.36), (\psi_4, 0.25) \rangle \}, \{ \langle (\psi_2, 0.27), (\psi_4, 0.25) \rangle \}$$

The agent proceeds to blend the first pair in the list. To this end, it applies the amalgam-based blending. The least general generalisation of ψ_1 and ψ_3 is an icon with two sorts of type SIGN, one *on* the other one, and with meaning ACTION and OBJECTTYPE respectively. The agents explores the space of generalisations and finds two maximal blends; a blend ψ_{b_1} describing an icon with two sorts of type MAGNIFYINGGLASS and DOCUMENT whose meaning is SEARCH-DOC; another blend ψ_{b_2} describing an icon with sorts of type PEN and HARDDISK whose meaning is EDIT-HARDDRIVE. Since ψ_{b_2} does not satisfy the query, is discarded, and only ψ_{b_1} is kept. The creation of ψ_{b_1} is illustrated in Figure 4.

The agents repeats the above procedure for each pair discovered. Finally, it finds another blend, which satisfies ψ_q , by blending the pair ψ_1 and ψ_4 . It is a blend describing an icon with three sorts of type MAGNIFYINGGLASS, DOCUMENT, and CLOUD whose meaning is SEARCH-DOC-CLOUD. Intuitively, this blend can be obtained by generalising HARDDISK from ψ_1 and ARROW from ψ_4 , and by keeping the other input icons' specifics. We denote this blend as ψ_{b_2} . The set of blends is $\mathcal{B} = \{ \psi_{b_1}, \psi_{b_2} \}$. A representation of ψ_{b_1} and ψ_{b_2} is given in Figures 2b(V-VI).

The agent evaluates these blends by means of the arguments and values described in the previous section. The blend ψ_{b_1} contains 10 variables whereas ψ_{b_2} contains 14. Therefore, the simplicity value's degrees of ψ_{b_1} and ψ_{b_2} are 0.1 and 0.07 respectively. Their unambiguity, on the other hand, is 1, since the Rich Background does not contain icons

with the same signs used in ψ_{b_1} and ψ_{b_2} , but with a different meaning. The arguments built by the agent are:

	Simplicity	Unambiguity
ψ_{b_1}	0.1	1
ψ_{b_2}	0.07	1

Therefore, both blends have an argument pro regarding their simplicity and an argument con w.r.t. their unambiguity value. It is easy to see that the blends are ranked in different ways when using the criteria we defined. For instance, ψ_{b_1} and ψ_{b_2} are equally preferred when counting their arguments pros (or cons) (Definition 7), and when considering both arguments pros and cons (Definition 9). Instead, ψ_{b_1} is preferred to ψ_{b_2} when using the criteria that take the audience into account (Definitions 8 and 11).

Conclusion and Future Work

In this paper, we described a process model for concept invention that is based on and extends the conceptual blending theory of Fauconnier and Turner (2002). According to this process, concept invention is characterised by different sub-processes—discovery, blending, and evaluation—that together account for concept invention. We proposed its computational model in terms of feature terms, a formal knowledge representation language. This allowed us to capture the concept invention process in terms of well-defined operators such as anti-unification—for computing a generic space—and unification—for computing a blend. Pairs of input concepts are retrieved from a Rich Background by means of a discovery process that takes a similarity measure into account. Blending is realised according to the notion

of amalgam, and blend evaluation is achieved by means of arguments, values and audience.

We exemplified the computational framework in the domain of computer icon design, but the framework is general enough to be used in other domains such as music or poetry generation. We plan to explore the use of arguments, values and audiences as a means to evaluate concept blends in such domains as future work.

We also aim at extending the process model by including the notion of coherence by Thagard (2000). Coherence theory, when used to explain human reasoning, proposes that humans accept or reject a cognition depending on how much it contributes to maximising the constraints imposed by situations or other cognitions. In the case of concept invention, coherence can be defined and used, for instance, to measure to what extent a blend coheres or incoheres with the Rich background and other blends.

Acknowledgments

This work is partially supported by the COINVENT project (FET-Open grant number: 611553).

References

- Amgoud, L., and Prade, H. 2009. Using arguments for making and explaining decisions. *Artificial Intelligence* 173:413–436.
- Atkinson, K.; Bench-Capon, T.; and McBurney, P. 2004. Justifying practical reasoning. In *Proc. of the Fourth Workshop on Computational Models of Natural Argument (CMNA'04)*, 87–90.
- Bench-Capon, T. J. M. 2003. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation* 13(3):429–448.
- Besold, T. R., and Plaza, E. 2015. Generalize and Blend: Concept Blending Based on Generalization, Analogy, and Amalgams. In *Proc. of the 6th Int. Conf. on Computational Creativity, ICCCI5*.
- Bonet, B., and Geffner, H. 1996. Arguing for decisions: A qualitative model of decision making. In *Proc. of the 12th Conf. on Uncertainty in Artificial Intelligence (UAI'96)*, 98–105.
- Carpenter, B. 1992. *The Logic of Typed Feature Structures*. New York, NY, USA: Cambridge University Press.
- Confalonieri, R.; Corneli, J.; Pease, A.; Plaza, E.; and Schorlemmer, M. 2015a. Using Argumentation to Evaluate Concept Blends in Combinatorial Creativity. In *Proc. of the 6th Int. Conf. on Computational Creativity, ICCCI5*.
- Confalonieri, R.; Eppe, M.; Schorlemmer, M.; Kutz, O.; Peñaloza, R.; and Plaza, E. 2015b. Upward Refinement for Conceptual Blending in Description Logic —An ASP-based Approach and Case Study in \mathcal{EL}^{++} . In *Proc. of 1st Int. Workshop of Ontologies and Logic Programming for Query Answering*. Co-located with IJCAI-2015.
- Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence* 77:321–357.
- Eppe, M.; Confalonieri, R.; Maclean, E.; Kaliakatsos-Papakostas, M. A.; Cambouropoulos, E.; Schorlemmer, W. M.; Codescu, M.; and Kühnberger, K. 2015a. Computational Invention of Cadences and Chord Progressions by Conceptual Chord-Blending. In *IJCAI 2015*, 2445–2451.
- Eppe, M.; Maclean, E.; Confalonieri, R.; Kutz, O.; Schorlemmer, W. M.; and Plaza, E. 2015b. ASP, Amalgamation, and the Conceptual Blending Workflow. In *Logic Programming and Nonmonotonic Reasoning - 13th Int. Conf., LPNMR 2015, Lexington, KY, USA, September 27-30, 2015.*, 309–316.
- Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending And The Mind's Hidden Complexities*. Basic Books.
- Goguen, J. A., and Harrell, D. F. 2005. Foundations for active multimedia narrative: Semiotic spaces and structural blending. Manuscript to appear published in the journal “Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems”.
- Goguen, J. 1999. An introduction to algebraic semiotics, with application to user interface design. In Nehaniv, C. L., ed., *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *LNCS*. 242–291.
- Harrell, D. F. 2005. Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation. *6th Digital Arts and Culture Conference*.
- Harrell, F. 2007. *Theory and technology for computational narrative: an approach to generative and interactive narrative with bases in algebraic semiotics and cognitive linguistics*. Ph.D. Dissertation, University of California, San Diego.
- Malcolm, G. 2000. *Software Engineering with OBJ: algebraic specification in action*. Kluwer.
- Ontañón, S., and Plaza, E. 2012. Similarity measures over refinement graphs. *Machine Learning* 87(1):57–92.
- Ontañón, S., and Plaza, E. 2010. Amalgams: A Formal Approach for Combining Multiple Case Solutions. In *Proc. of the Int. Conf. on Case Base Reasoning*, volume 6176 of *Lecture Notes in Computer Science*, 257–271. Springer.
- Pereira, F. C. 2007. *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Mouton de Gruyter.
- Pollock, J. 1992. How to reason defeasibly. *Artificial Intelligence Journal* 57:1–42.
- Smolka, G., and Ait-Kaci, H. 1989. Inheritance hierarchies: Semantics and unification. *Journal of Symbolic Computation* 7(3–4):343–370.
- Thagard, P. 2000. *Coherence in thought and action*. The MIT Press.
- Veale, T., and Donoghue, D. O. 2000. Computation and blending. *Cognitive Linguistics* 11(3-4):253–282.
- Veale, T., and Keane, M. 1997. The competence of sub-optimal theories of structure mapping on hard analogies. In *IJCAI*, 232–237.

Optimality Principles in Computational Approaches to Conceptual Blending: Do We Need Them (at) All?

P. Martins¹, S. Pollak², T. Urbančič^{3,2}, A. Cardoso¹

¹ CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

² Jožef Stefan Institute, Ljubljana, Slovenia

³ University of Nova Gorica, Nova Gorica, Slovenia

Abstract

Optimality principles are a key element in the Conceptual Blending (CB) framework, as they are responsible for guiding the integration process towards 'good blends'. Despite their relevance, these principles are often overlooked in the design of computational models of CB. In this paper, we analyse the explicit or implicit presence and relevance of the optimality principles in three different computational approaches to the CB, known from the literature. The approaches chosen for the analysis are Divago, Blending from a generalisation-based analogy model, and blending as a convolution of neural patterns. The analysis contains a discussion on the relevance of the principles and how some of absent principles can be introduced in the different models.

Introduction

Fauconnier and Turner (2002) proposed *Conceptual Blending* (CB) as a general and basic cognitive mechanism that leads to the creation of new meaning and insight. It integrates (or blends) two or more *mental spaces* in order to produce a new mental space, the *blend(ed) space*. Here, mental space means a temporary knowledge structure created for the purpose of local understanding as opposed to *frames*, which are more stable knowledge structures (Fauconnier 1994).

CB is not a simple combination of the initial mental spaces; it involves a network of mental spaces in which knowledge is transferred and meaningfully integrated (see Figure 1). At least two of the mental spaces correspond to the *input spaces* (the initial spaces). A partial matching between the input spaces is constructed (*cross-space mapping*). The matching between elements is then reflected in another mental space, the *generic space*, which contains elements common to the different input spaces. The latter space captures the conceptual structure that is shared by the input spaces. The outcome of the blending process is the *blend*, a mental space that simultaneously maintains partial structures from the input spaces and has an emergent structure of its own.

Integration of input elements in the blend space results from three operations: *composition*, *completion*, and *elaboration*. Composition occurs when the elements from the input spaces are projected into the blend space, allowing for

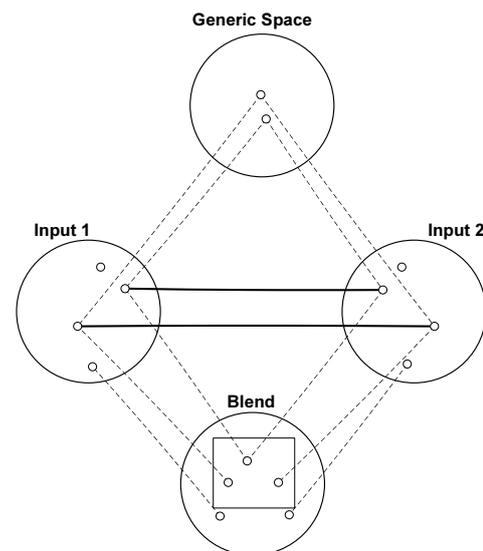


Figure 1: The original four-space CB network (Fauconnier and Turner 2002).

new relations to become available in the blended space. This implies not only the matched elements, but also other neighbouring elements to be projected into the blend. Completion occurs when existing knowledge in long-term memory, i.e., knowledge from *background frames*, is used to create meaningful structures in the blend. Elaboration is an operation closely related to completion; it involves cognitive work to perform a simulation of the blended space.

The possibilities for blending are apparently infinite and the quality of blends can be quite diverse. The *optimality principles* (also known as *optimality constraints*) have a key role in blending, namely in the integration process. They are responsible for providing guidance towards highly integrated, coherent and easily interpreted blends.

Despite the challenge in designing a computational model of the CB mechanism, several formalisations and computational models of the CB mechanism have been proposed. The inclusion of the optimality principles in formal and computational models has arguably been one of the most challenging tasks, mainly due to the subjectivity and the

computational inefficiency associated with these principles.

Bou et al. (2014) have presented a survey of computational approaches to conceptual blending where the presence of the optimality principles was assessed. To the best of our knowledge, the work of Bou et al. offers the most detailed discussion on the computational modelling of the optimality principles. In this paper, we analyse the presence and relevance of the optimality principles in three considerably different approaches to the CB mechanism. Instead of solely basing our analysis on the assessment of the presence or absence of the optimality principles, we discuss the relevance of the principles and how some of the absent principles can be introduced in the different models. We include in our discussion suggestions from our previous study on the quality of blends (Martins et al. 2015). It is particularly relevant to analyse the importance of optimality principles in scenarios where creative blends are a goal.

The remainder of this paper is structured as follows. In the upcoming section, we describe the optimality principles of CB theory. Then, we analyse optimality principles in computational models of CB. Finally, we draw the main conclusions of this study and suggest lines of further research.

Optimality principles

Originally, Fauconnier and Turner (1998) have presented a list of five optimality principles (integration, topology, web, relevance, and unpacking). Later, the same authors have extended the list by including three more principles (maximisation of vital relations, intensification of vital relations, and pattern completion) (Fauconnier and Turner 2002). This paper focuses on the latter.

The Principles

Integration The *integration* principle states that the blend must be perceived and manipulated as a unit. Every element in the blend structure should have integration.

Topology *Topology* acts as a force that attempts to maintain the topological structure of the input spaces in the resulting blend. For any input space and any element in that space projected into the blend, it is optimal for the relations of the element in the blend to match the relations of its counterpart.

Intensification of Vital Relations A key characteristic of the blending process is the ability to *compress* a diffuse conceptual structure into more intelligible and manipulable human-scale situations in the blended space (Fauconnier 2005; Turner 2006). Such compression is likely to occur when mental spaces are connected by *vital relations*, such as time, space, cause-effect, analogy or a part-whole relation. The principle known as *intensification of vital relations* states that diffuse structures should be compressed by scaling a single vital relation (e.g. scale down an interval of time) or transforming vital relations into others.

Maximisation of Vital Relations The *maximisation of vital relations* principle states that the number of vital relations in the blended space should be maximised in order to create human scale.

Pattern Completion The *pattern completion* principle forces the introduction of integrated patterns either from the input spaces or from frames. The elements in the blend should be completed using existing integrated patterns as additional inputs. The principle dictates the use of a completing frame having relations that can be the compressed versions of the important *outer-space vital relations* (space, time, etc.) between the inputs.

Web The *web* principle states that manipulating the blend as a unit must maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation.

Relevance (or Good Reason) The *relevance* principle dictates that an element in the blend should be relevant, which includes being relevant to establish links to other spaces and for running the blend.

Unpacking The *unpacking* principle imposes the ability to ‘deconstruct’ the whole blending process starting from the blended space. This principle takes exclusively the perspective of the ‘blend reader’, who is expected to recognise the input spaces and the results of intermediate operations, namely the cross-space mappings.

Optimal blends vs. creative blends

All of the listed principles try to ensure an easy interpretation of the blend and trigger a prompt cognitive response. Additionally, they intend to provide integrity and coherence, namely by the integration, web, and topology principles. However, there is a tension among the principles, which includes different levels of incompatibility between them (Grady, Oakley, and Coulson 1999). For example, an intensification of vital relations might hinder the ability to reconstruct the entire blending network (unpacking principle).

The aforementioned tension among principles makes the construction of a blend satisfying all the principles impossible. However, we cannot simply regard these principles as ‘rigid laws’, but as something with a reasonable degree of flexibility (Kowalewski 2008). Furthermore, the optimality of a blend depends on its purpose: different purposes imply distinct levels of priority for each principle.

While the optimality principles can provide guidance towards consistent, useful, and easily interpreted blends, we cannot ensure that they contribute to defining novel and surprising blends. Thus, the criteria conveyed by the optimality principles cannot dictate whether a blend is creative or not. Nonetheless, they help defining other ‘good characteristics’ of a creative blend.

Optimality Principles and Computational Approaches to Conceptual Blending

‘Conceptual blending is not a compositional algorithmic process and cannot be modeled as such for even the most rudimentary cases. Blends are not predictable solely from the structure of the inputs. Rather, they are highly motivated by such structure, in harmony with independently available background and

contextual structure; they comply with competing optimality constraints ... and with locally relevant functional goals. In this regard, the most suitable analog for conceptual integration is not chemical composition but biological evolution. Like analogy, metaphor, translation, and other high-level processes of meaning construction, integration offers a formidable challenge for explicit computational modeling.' (Fauconnier and Turner 1998).

Despite the challenge in computationally modelling the CB mechanism, several formalisations and computational models of the CB mechanism have been proposed. The inclusion of the optimality principles in such models has arguably been one of the most challenging tasks, mainly due to the subjectivity and the computational inefficiency associated with these principles. According to Goguen (1999), who proposed one of the first formalisations of CB theory, the optimality principles are one of the components of CB theory that cannot be formalised and straightforwardly implemented, as they require human judgment.

In this section, we analyse the role implicitly or explicitly played by the optimality constraints in three different computational models: (1) Divago (Pereira 2005), which is strongly inspired by CB theory and contains quantitative metrics for the optimality principles; (2) a model that follows a neuro-computational approach (Thagard and Stewart 2010), with blending being performed via the convolution of mental representations; and, finally, (3) a model constructed using a generalisation-based approach to analogy (Guhe et al. 2011).

We have opted for these three models because they simultaneously illustrate the heterogeneity and the maturity of computational approaches to CB. For an updated and a more complete overview of computational approaches to CB, we refer the reader to the works of Martins et al. (2014), Bou et al. (2014), or Li et al. (2012).

When analysing the relevance of some principles, we take into account also suggestions from our previous study in which we investigated the quality of blends as perceived by humans in a web-based questionnaire (Martins et al. 2015). The participants were asked to rate criteria related to the optimality principles (e.g., coherence) and creativity (e.g., novelty and surprise).

Divago

The Divago system (Pereira 2005) is one of the earliest computational approaches to CB and is, to the best of our knowledge, the only system to date that uses a thorough formalisation of the optimality principles. The architecture of the system is depicted in Figure 2.

In Divago, the first step corresponds to selecting a pair of input spaces (domains) from the *Knowledge Base*. The input spaces are represented as concept maps, i.e., graphs where vertices are concepts and edges represent relations. The selection of such spaces is performed by the user or randomly generated. Then, the *Mapper* module performs the selection of elements for projection. Such selection is achieved by means of a partial mapping between the input

spaces using *structural alignment*. This operation looks for the largest isomorphic pair of sub-graphs contained in the input spaces.

For each mapping provided by the *Mapper*, the *Blender* performs a projection into the blend space. At this stage, all the possible projections resulting from each mapping must be represented in the blend space. The whole set of projections summarises the *Blendoid*, which is the set of all possible blends.

The *Factory* module is responsible for exploring the space of all possible blends provided by the *Blender*. The *Factory* interacts both with the *Elaboration* and *Constraints* modules: it is based on a genetic algorithm (GA) that looks for the elaborations that best fulfill the requirements dictated by the *Constraints* module. At each iteration, the GA sends each blend to the *Elaboration* module, which is responsible for applying context-dependent knowledge, and then sends the result to the *Constraints* module, which applies the optimality principles in order to evaluate the elaborated blend. When the GA finds an adequate solution (or a pre-defined number of iterations is reached), the *Factory* stops the execution of the genetic algorithm and returns the best blend.

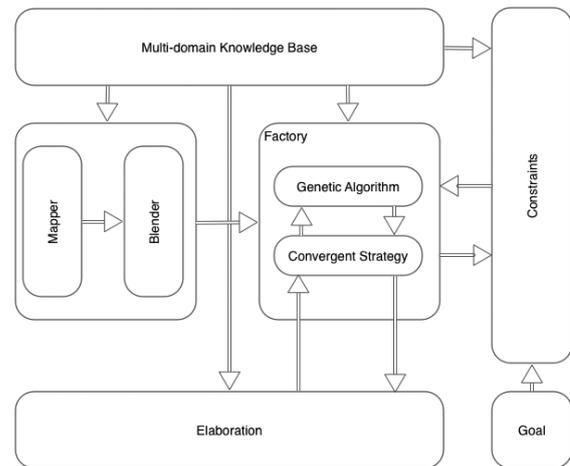


Figure 2: Divago architecture.

The *Constraints* module contains an implementation of the optimality principles based on quantitative metrics.

Integration

The measure of integration is based on the idea of *frame coverage*. If F is the set of frames that are satisfied in a blend, frame coverage corresponds to the set of relations from its concept map that belong to the set of conditions of one or more frames in F .

Definition 1 (*SingleFrameIntegration*). For a frame f with a set C of conditions, a blend b , with a concept map CM_b , its blendoid with a concept map, CM_{B^+} , and VI , the set of integrity constraints that are violated in the frame, the

integration value, I_f , is defined by:

$$I_f = \frac{\#C}{\#CM_b} \times (1 - \iota)^{\#VI} \times (1 + \frac{\#CM_b}{\#CM_{B^+}}) / 2, \quad (1)$$

where ι is a penalty factor between 0 and 1, a value that penalises a frame for each violation of integrity constraints. An integrity constraint is violated if its premises are true. In the context of the integration measure of frame f above, f violates integrity ic if the conditions C_{ic} of ic are met and $C_{ic} \cap C \neq \emptyset$.

Integration is estimated through the following equation:

Definition 2 (Integration). Let $F_b = \{f_1, f_2, \dots, f_i\}$ be the set of the frames that have their conditions (C_i) satisfied in the blend b , α , the disintegration factor (with $0 < \alpha < 1$), and I_{f_i} , the single frame integration value, as in Eq. (1).

$$Integration = I_{\bigcap_0^i C_i} + \alpha \times Uncoverage \times \sum_0^i I_{f_i}. \quad (2)$$

The *Uncoverage* value consists of the ratio of relations that do not belong to the intersection of all frames w.r.t. the total number of relations considered in the frames:

$$Uncoverage = \frac{\#\bigcup_0^i C_i - \#\bigcap_0^i C_i}{\#\bigcup_0^i C_i}. \quad (3)$$

Topology

The topology measure follows the principle that if a pair of concepts x and y are associated in the blend by a relation r , then the same relation must exist in the inputs between the elements from which x and y were projected. In this case, the relation $r(x, y)$ is *topologically correct*. The topology measure corresponds to the ratio of topologically correct relations in the concept map of the blend:

Definition 3 (Topology). For a set $TC \subseteq CM_b$ of *topologically correct relations*, defined as

$$TC = \{r(x, y) : r(x, y) \in CM_1 \cup CM_2\}, \quad (4)$$

where CM_1 and CM_2 correspond to the concept maps of inputs 1 and 2, respectively. The topology measure is calculated by the ratio:

$$Topology = \frac{\#TC}{\#CM_b}. \quad (5)$$

Maximisation/Intensification of Vital Relations

In Divago, intensification is treated as maximisation, i.e., there is only one measure for the principles related to the vital relations. To define the maximisation measure, the impact of the vital relations to the blend is given by the ratio of vital relations w.r.t. the whole set of possible vital relations, contained within the blendoid:

Definition 4 (Maximisation_VR). Let Υ be a set of *vital relations*. From the concept map of the blend b , we may obtain the set of vital relations in b , B_{VR} :

$$B_{VR} = \{r(x, y) : r(x, y) \in CM_b \wedge r \in \Upsilon\}.$$

From the blendoid (the union of all possible blends), B^+ , we have B_{VR}^+ :

$$B_{VR}^+ = \{r(x, y) : r(x, y) \in CM_B^+ \wedge r \in \Upsilon\}.$$

Finally, the Maximisation of Vital Relations measure is calculated by the ratio

$$Maximisation_VR = \frac{\#B_{VR}}{\#B_{VR}^+}.$$

Pattern Completion

In Divago, pattern completion is viewed as frame completion, as a pattern is described by a frame. The act of completing a frame consists in asserting the truth of the ungrounded premises, a process that happens only after a sufficient number of premises is true (*completion threshold*). The measure that indicates the conditions that are actually satisfied by a frame f in a blend b is called *completion evidence* of f , $e(f, b)$. (Frame) completion can only happen when the completion evidence is higher than the completion threshold.

Definition 5 (Completion Evidence). The Completion Evidence e of a frame f_i with regard to a blend b is calculated according to the following:

$$e(f_i, b) = \frac{\#Sat_i}{\#C_i} \times (1 - \iota)^{\#VI}, \quad (6)$$

where Sat_i contains the conditions of each f_i that are satisfied in b , C_i contains the conditions of f_i , ι is the integrity constraint violation factor and VI the set of violated integrity constraints.

In the end, pattern completion is computed by finding the union of all the conditions contained within the patterns and estimating its own completion evidence:

Definition 6 (Pattern Completion). The Pattern Completion measure of a blend b with regard to a set of frames F is calculated by

$$PatternCompletion = e(\bigcup_{f_i \in F} f_i, b). \quad (7)$$

Web

The web principle is not treated as an independent principle; it is co-related to topology and unpacking. As a result, it is given as an estimation of the strength of the web of connections to the inputs:

Definition 7 (Web).

$$Web = \lambda \times Topology + \beta \times Unpacking, \quad (8)$$

with $\lambda, \beta \geq 0$ and $\lambda + \beta = 1$.

Relevance

The idea of relevance is strongly associated with the goal of blending:

Definition 8 (Relevance). Assuming a set of goal frames, F_g , the set F_b of the satisfied frames of blend b and the value PCN_F for the pattern completion of a set of frames F in blend b , relevance is given by:

$$Relevance = \frac{\#(F_g \cap F_b) + \#F_u \times PCN_{F_u}}{\#F_g}, \quad (9)$$

where F_u , the set of unsatisfied goal frames, consists of $F_u = F_g - F_b$. This formula gives the ratio of satisfied and partially satisfied goal frames w.r.t. the entire set, F_g of goal frames.

Unpacking

Unpacking is reduced to the ability to reconstruct the input spaces. To measure it, the definition of *defining frame* is required:

Definition 9 (DefiningFrame). Given a blend b and an input space d , the element x (which is the projection of the element x_d of input concept map d to b) has a defining frame $f_{x,d}$ consisting of

$$f_{x,d} = C_0, C_1, \dots, C_n \longrightarrow true, \quad (10)$$

where $C_i \in \{r(x, y) : r(x_d, y) \in CM_d\}$. Assuming that k is the number of conditions (C_i) of $f_{x,d}$ that are satisfied in the blend, the unpacking value of x with regard to d (represented as $\xi(x, d)$) is

$$\xi(x, d) = \frac{k}{n'}, \quad (11)$$

where n' is the number of elements to which x is connected. The *total estimated unpacking value* of x as being the average of the unpacking values with regard to the input spaces:

$$\xi(x) = \frac{\xi(x, 1) + \xi(x, 2)}{2}. \quad (12)$$

Definition 10 (Unpacking). Let \mathcal{X} be the set of m elements of the blend b , generated from input concept maps 1 and 2. The *Unpacking* value of b is calculated by

$$Unpacking = \frac{\sum_{i=0}^m \xi(x_i)}{m}, x_i \in \mathcal{X}. \quad (13)$$

Blending as a convolution of neural patterns

Thagard and Stewart (2010) propose a neuro-computational approach based on a mechanism that combines neural activity patterns by a process of *convolution*, a mathematical operation that interweaves structures. The main idea behind such approach is to build combinations of neural activity patterns that are probably useful and novel. The work aims at modelling the so-called *AHA! moment*, which occurs when humans discover surprising relations between apparently unrelated pieces of information.

Concepts are represented as activity patterns of vectors of neurons, which are convoluted in order to combine patterns (the use of convolution to combine neural representations is based on the assumption that any representation can be treated as a vector). Although the authors do not explicitly claim that their approach models the CB mechanism, they highlight the similarities between the proposed account of creativity and the blending mechanism. A key feature of this model is the ability to combine several multimodal representations, including information that can be sensorial, kinesthetic, and verbal, as well as emotional (see Figure 3). As for the latter, it is worth mentioning that emotional reactions play a key role in creative thought; in particular, the

reaction of pleasure/approval that is associated with the generation of novel and surprising ideas. As a result, the AHA! experience is presented as a convolution of a novel combined representation with patterns of brain activity for emotion.

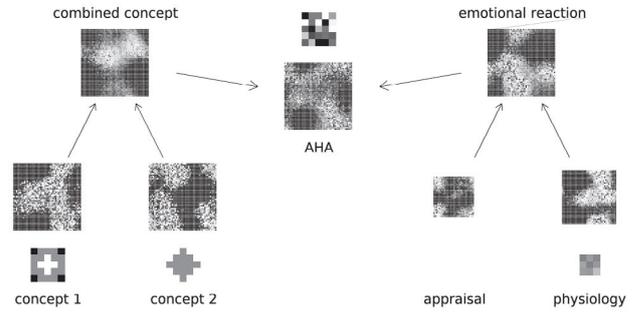


Figure 3: The AHA! experience as a convolution of neural patterns (combination of four representations into a single one). The arrows indicate the flow of information although many reentry feedback loops may occur (Thagard and Stewart 2010).

Another relevant feature of this model is the ability to reverse the process of convolution, using neural connections similar to those required for performing the convolution. This reverse process, which is known as *deconvolution*, implies loss of information: the output is an approximation of the original patterns.

Blending from a Generalisation-Based Analogy Model

Guhe et al. (2011) present an account of blending based on the *Heuristic-Driven Theory Projection* (HDTP) (Schwering et al. 2009; Gust, Kühnberger, and Schmid 2006), which was originally proposed as a framework for analogy making. HDTP represents knowledge about the domains as *first-order logic theories*, whose analogical mapping is established via *anti-unification*, i.e., an analogical relation is built by associating terms with a common generalisation.

In the HDTP framework, knowledge is mapped and transferred from a source domain S to a target domain T . To create an analogy, two stages are required: *mapping phase* and *transfer phase*. In the former, the two domains are compared to find structural commonalities, leading to the creation of a generalised description G that contains the matching parts of both domains. In the final phase, unmatched knowledge in S can be mapped to the target domain to create new hypotheses.

The first phase is similar to the cross-space mapping and the generation of the generic space in the CB framework. In fact, the authors turn the HDTP framework into a CB framework by modifying the second phase: the knowledge transfer is replaced by a process that creates a new knowledge domain B , the blend. Knowledge from S and T is merged to create B based on the following mapping: *‘in the ideal case, B respects the shared features of S and T (those with common generalisations), and inherits independently the other*

features of S and T '.

Since unmatched parts of the domains will be transferred into the blend, which may introduce incoherence, the framework has the ability to either discard conflicting knowledge or reduce the coverage of the generalisation.

Figure 4 depicts a diagram illustrating the extension of the HDTP framework to CB.

In this model, the mental spaces are represented by *many-sorted first-order theories*. To blend two theories, three steps are required: (i) definition of *core (blend) laws*, which unite input signatures to generate new signatures ; (ii) addition of *preferred conjectures* (generation and addition of laws that concern equality of analogous entities, functions and relations) ; (iii) definition of *extra conjectures* (addition of laws from the input spaces) (Bou et al. 2014).

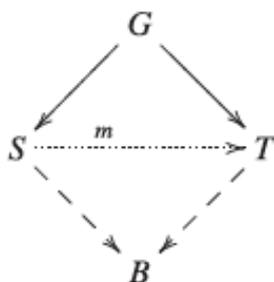


Figure 4: HDTP as a blending framework. Top arrows denote substitutions resulting from the computation of the analogical relation m . Dashed arrows indicate the heuristic-driven construction of the conceptual blend (Guhe et al. 2011).

Discussion

Among the three models previously described, Divago is the one that encapsulates more elements of CB theory. Despite the inherent subjectivity involved, the Constraints module in Divago tries to be consistent with theory regarding the optimality principles. This gives Divago a certain modularity, as different principles and weights can be considered as a function of the task at hand. The inclusion of metrics to assess the presence of vital relations is probably the most noteworthy characteristic of the constraints module. However, it is important to note that Divago does not perform compression (of vital relations).

The neuro-computational approach based on the convolution of neural patterns is not directly inspired by CB theory. Its inspiration comes from findings in the field of neuroscience that can be related to blending (neural combination and binding). This kind of approach does not take into consideration the optimality principles. However, we believe that the inclusion of those principles in the model of emotional reactions would make the whole model more complete, as it would define emotional reactions that are associated not only with novelty and surprise but also with the coherence and interpretation of ideas. Here, the challenging

task is to model neural processes that can generate inputs to assess the presence of the optimality principles.

The blending model developed from a generalisation-based analogy model is particularly suitable for generating blends that are mostly analogical constructions. The preferred and extra conjectures that can be added during the generation of blends share some similarities with the optimality principles in terms of role in the blending process, as they help discard unwanted blends.

While some models try to include the optimality principles, others do not take them into consideration. But are all the principles relevant? And, when they are not an obvious part of the model, could they be implicitly defined? We present our view on these questions for each one of the optimality principles and for each one of the computational approaches described herein.

Integration

Integration is a principle that most contributes to the integrity of a blend and it cannot be completely disregarded. Our experiments with visual blends showed the importance of integrity to the quality of a blend; there was a high correlation between integrity (or coherence) and the overall impression of the blend (Martins et al. 2015). Figure 5 depicts two examples of visual blends (fictional hybrid animals) used in our survey: *Guorse* and *Pengwhale*. The former was among the blends with the lowest overall impression and coherence scores, whereas the latter was among the favourite blends (with high overall impression and coherence scores).

Integration is present in each one of the models but in apparently varying degrees. Divago performs integration both in an explicit and implicit way. The former corresponds to the maximisation of the criterion given by Equation (2). However, integration is also achieved to some extent through the strategy used to perform cross-space mapping, as Divago basis its mapping on structural alignment, which ensures a certain degree of integration.

The blending model based on the HDTP framework tries to ensure integration by constructing the generalisation model and in subsequent stages the integrity criterion is still taken into consideration.

In the neuro-computational model, the convolution of neural activity patterns is by definition an integration operation. However, to ensure a higher level of integration, it is fundamental to assess integration through the emotions module.

Topology

As for the Topology principle, we can argue that its relevance is somewhat relative. On one hand, it can ensure consistency to some extent, as it contributes to the external coherence of the blend. On the other hand, it tends to inhibit the inclusion of more uncommon associations. However, as observed by Pereira (2005), the importance of maintaining the same topological arrangement depends on the type of the blend we are aiming at. For example, if our construction pursues an analogy, then topology becomes crucial; if we are pursuing less strict combinations, then it should become secondary.



Guorse
(guinea pig, horse)



Pengwhale
(penguin, whale)

Figure 5: Two examples of fictional hybrid animals used in the online questionnaire (Martins et al. 2015). Each sub-caption contains the corresponding name of the blend as well as the input spaces. The author of both blends is Arne Fredriksen (<http://gyyp.imgur.com/>).

The model based on the HDTP framework follows an approach that tries to build externally coherent blends, despite the absence of an implementation of the optimality principles. With regard to the neuro-computational approach, topology could be assessed via the emotions module.

Maximisation/Intensification of Vital Relations

A maximisation or an intensification of vital relations contributes to make the blend easier to understand and to trigger a prompt cognitive response. However, maximisation (or intensification) is not always possible, as the mental spaces are not always connected by vital relations. Furthermore, computationally modelling the phenomenon of compression, i.e., bringing appropriate relations from different inputs to the blend can be challenging.

Pattern Completion

We do not view pattern completion as a fundamental principle. It can enrich the blend, but it is not the type of constraint

that, by itself, contributes more to the integrity and easy understanding of a blend. However, we believe it cannot be completely disregarded, especially when there is some incompleteness associated with the blend. Additionally, frame pattern completion can increase the capabilities and relevance of the blend.

Any of the computational models described in this paper can accommodate an implementation of this principle. In the HDTP-based approach, this can be achieved via the definition of extra conjectures. In the neuro-computational model, pattern combination could be assessed by inputs related to the incompleteness of patterns.

Web

The web principle ensures that elaboration is performed without removing links to the input spaces. This constraint has a direct relation with the topology and unpacking principles, as they try to maintain the connections to the input spaces. More particularly, the topology principle tries to maintain the web of relevant connections to the input spaces, whereas unpacking tries to reduce the cognitive work associated with the reconstruction of the input spaces.

In our view, this is a relevant principle in most of the scenarios, as it promotes the easy understanding of the blended space and tends to produce an immediate cognitive effect. However, it does not have to be directly applied, as it depends on the topology and unpacking principles.

Relevance

Relevance is a principle that is associated with the usefulness of the blend. Since the quality of a blend depends on its purpose, it is fundamental to understand the usefulness of the various elements of a blend. An inexistent blending goal can be detrimental to the assessment of the relevance. It is therefore advantageous to have additional knowledge regarding the blending goal and how it relates to the elements of the blend. This principle is usually implicitly present. For example, the HDTP-based model can use the preferred and extra conjectures to define goals.

Unpacking

Our previous series of experiments on the evaluation of visual blends suggested that unpacking is relevant in order to better understand the blend. The participants tended to emphasise the importance of recognising the input spaces (Martins et al. 2015). However, there was also a generalised opinion that the favourite blends were those whose unpacking took some time to occur. The unpacking act can give a hint on the level of surprise of a blend: a longer unpacking tends to suggest a higher level of surprise. However, too much surprise can be detrimental to the quality of the blend.

As for the external coherence of the blend, we believe that unpacking is a fundamental criterion. However, for some approaches, it can become challenging to evaluate the easiness of reconstructing the integration network or simply determining the input spaces. In those cases, a topology measure is required to account for external coherence.

Since the convolution of neural patterns can be reverted (via deconvolution), we can say that the neuro-computational approach follows the unpacking principle.

We also argue that HDTP-based CB model tries to follow this principle, as it tries to add the maximum number of symbols of the input spaces to the blends.

Conclusions and further work

The optimality principles are a fundamental element in CB theory. They are responsible for guiding the integration process towards highly integrated, coherent and easily interpreted blends. While several computational models of the CB mechanism have been proposed and successfully used as creative systems, the inclusion of the optimality principles in those models has been overlooked, mainly due to the subjectivity and the computational inefficiency associated with this element of CB theory.

In this paper, we have analysed the presence as well as the relevance of the optimality principles in three different approaches to the CB mechanism. Three substantially different computational models were studied: Divago (explicit presence), CB as a convolution of neural patterns (implicit presence) and CB from a Generalization-Based Analogy Model (implicit presence).

From our analysis, we believe that not all principles are relevant. Integration, topology, unpacking, relevance, and intensification/maximisation of vital relations appear to be the most crucial ones. In fact, principles such as integration and topology tend to be implicitly present in models that apparently overlook the optimality principles.

Integration is the most vital principle to establish the integrity of the blend. Topology and unpacking are responsible for defining the external coherence of the blend. However, if we want to favour the introduction of uncommon associations, topology and unpacking can be treated as secondary principles.

Maximisation (and intensification) of vital relations can contribute to an easier understanding of the blend and create a more immediate cognitive effect. As such, they are also fundamental principles.

Relevance is related to the usefulness of the blend and, as a result, we believe it cannot be disregarded in most cases.

As future work, we will continue our investigation on the relevance of the optimality principles. We also plan to reimplement the Constraints module in the Divago framework using some of the discoveries made during our study.

Acknowledgments

This research was partly funded through EC funding for the project ConCreTe (grant number 611733) that acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission.

References

Bou, F.; M., E.; Plaza, E.; and Schorlemmer, M. 2014. D2.1 - reasoning with amalgams. Public deliverable, Concept Invention Theory (FP7 - 611553).

Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive Science* 22(2):133–187.

Fauconnier, G., and Turner, M. 2002. *The Way We Think*. New York: Basic Books.

Fauconnier, G. 1994. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. New York: Cambridge University Press.

Fauconnier, G. 2005. Compression and emergent structure. *Language and Linguistics* 6(4):523–538.

Goguen, J. 1999. An introduction to algebraic semiotics, with applications to user interface design. In *Lecture Notes in Artificial Intelligence*, volume Computation for Metaphor, Analogy and Agents, 242–291. Springer.

Grady, J. E.; Oakley, T.; and Coulson, S. 1999. Blending and metaphor. In Steen, G., and Gibbs, R., eds., *Metaphor in Cognitive Linguistics*.

Guhe, M.; Pease, A.; Smaill, A.; Martinez, M.; Schmidt, M.; Gust, M.; Kühnberger, K.-U.; and Krumnack, U. 2011. A computational account of conceptual blending in basic mathematics. *Cognitive Systems Research* 12(3–4):249–265. Special Issue on Complex Cognition.

Gust, H.; Kühnberger, K.-U.; and Schmid, U. 2006. Metaphors and heuristic-driven theory projection (hdtp). *Theoretical Computer Science* 354(1):98 – 117. Algebraic Methods in Language Processing Third International {AMAST} Workshop on Algebraic Methods in Language Processing 2003.

Kowalewski, H. 2008. Conceptual blending and sign formation. *The Public Journal of Semiotics* 2(2):30–51.

Li, B.; Zook, A.; Davis, N.; and Riedl, M. 2012. Goal-driven conceptual blending: A computational approach for creativity. In Maher, M. L.; Hammond, K.; Pease, A.; Pérez, R.; Ventura, D.; and Wiggins, G., eds., *Proceedings of the Third International Conference on Computational Creativity*, 9–16.

Martins, P.; Cardoso, A.; Urbančič, T.; Pollak, S.; Perovšek, M.; and Lavrač, N. 2014. Study and design of methods for concept blending. Public deliverable, Concept Creation Technology (FP7 - 611733).

Martins, P.; Urbančič, T.; Pollak, S.; Lavrač, N.; and Cardoso, A. 2015. The good, the bad, and the aha! blends. In *Proceedings of the 6th Int. Conference on Computational Creativity, ICC-15*.

Pereira, F. C. 2005. *Creativity and AI: A Conceptual Blending approach*. Ph.D. Dissertation, University of Coimbra.

Schwering, A.; Krumnack, U.; Kühnberger, K.-U.; and Gust, H. 2009. Syntactic principles of heuristic-driven theory projection. *Cognitive Systems Research* 10(3):251–269. Special Issue on Analogies - Integrating Cognitive Abilities.

Thagard, P., and Stewart, T. C. 2010. The AHA! experience: Creativity through emergent binding in neural networks. *Cognitive Science* 35(1):1–33.

Turner, M. 2006. Compression and representation. *Language and Literature* 15(1):17–27.

Learning to Blend Computer Game Levels

Matthew Guzdial, Mark Riedl

Entertainment Intelligence Lab, School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 303 USA
mguzdial3@gatech.edu, riedl@cc.gatech.edu

Abstract

We present an approach to generate novel computer game levels that blend different game concepts in an unsupervised fashion. Our primary contribution is an analogical reasoning process to construct blends between level design models learned from gameplay videos. The models represent probabilistic relationships between elements in the game. An analogical reasoning process maps features between two models to produce blended models that can then generate new level chunks. As a proof-of-concept we train our system on the classic platformer game Super Mario Bros. due to its highly-regarded and well understood level design. We evaluate the extent to which the models represent stylistic level design knowledge and demonstrate the ability of our system to explain levels that were blended by human expert designers.

Introduction

Concept blending is a powerful tool for problem solving in which two independent solutions combine into a novel solution referred to as a *blend*. It has been presented as a fundamental cognitive process and linked to the creation of creative artifacts (e.g. a griffin can be described as a blend between a lion and a bird) (Fauconnier and Turner 2002). Concept blending has traditionally appeared in expert systems applications, where a human expert encodes concepts from a particular field such as architecture, engineering, or mathematics (Goel 1997; Bou et al. 2015). Despite the concept blending’s creative potential, it has not appeared in the domain of video games to any large extent, even though games are well-suited to computational creativity research (Liapis, Yannakakis, and Togelius 2014). This is likely due to concept blending—and many other computational creativity techniques—relying on high quality knowledge bases. The quality of the “knowledge base” determines the quality of the blends a system is capable of constructing, meaning that a human expert often has to iterate over a knowledge base multiple times. In addition to the knowledge base, many concept blending systems require a means of evaluating blends, requiring human-authored heuristics.

Concept blending systems take a significant amount of human effort to construct. Machine learning could in theory derive a knowledge base from a corpus of examples, thus

reducing the requirement of human input. However, knowledge learned from machine learning techniques tends to be noisy, full of inconsistencies and mistakes that could thwart typical approaches to concept blending.

We present an unsupervised approach to concept blending video game levels, informed by a knowledge base learned from gameplay videos. The use of gameplay video is key to the unsupervised nature of our system as the system can infer human knowledge about exemplar game without requiring explicit human authoring. The learned knowledge base takes the form of probabilistic graphical models that are robust to the noisiness of machine learning with sufficient data. The models learn the likelihood of relationships between level elements, and can therefore evaluate the relative likelihood of a level, meaning that the blended models can evaluate blends without a human authored heuristic. We make use of Super Mario Bros. as a proof-of-concept game for our system, due to its popularity and highly-regarded level design.

Our contributions are as follows: (1) a novel concept blending approach to blend models capable of generation and evaluation, (2) a human evaluation of our system’s ability to evaluate how stylistically similar an input level is to exemplar gameplay levels, and (3) a case study of our blended models’ evaluation of human expert blended levels.

Background

Fauconnier and Turner (1998) formalized the “four space” theory of concept blending. In this theory they described four spaces that make up a blend: two *input spaces* represent the unblended elements, input space points are projected into a common *generic space* to identify equivalence, and these equivalent points are projected into a *blend space*. In the blend space, novel structure and patterns arise from the projection of equivalent points. Fauconnier and Turner (1998; 2002) argued this was a ubiquitous process, occurring in discourse, problem solving, and general meaning making.

Concept blending systems tend to follow some variation of the four spaces theory, but there exists a great variety of techniques to map between the concepts present in the various spaces (Falkenhainer, Forbus, and Gentner 1989). Analogical reasoning has traditionally been one of the leading conceptual mapping approaches, as it maps concepts based on relative structure instead of surface features.

This type of structural mapping has proven popular as it tends to better match human problem solving (Goel 2015; Bou et al. 2015). However, such analogical reasoning systems require a non-trivial amount of human input, as a human author must encode concepts in terms of their structure and how to compare structural information within a domain.

Due to the large amount of authorship required, it is unclear if the creative output of such a system arises from concept blending algorithms or the creativity of a human author when encoding structures. Recently O'Donoghue et al. (2015) have looked into deriving this knowledge automatically from text corpora, producing graphical representations of nodes and their verb connections. Our own work runs parallel to O'Donoghue et al., but in the domain of two dimensional video games levels and without the dependency rules that exist in the english language.

Concept blending, based on analogy or any other mapping technique, is not commonly used in video games. Prior work has looked into knowledge intensive concept blending systems to create new elements of video games such as sound effects and 3D models (Ribeiro et al. 2003; Martins et al. 2004). The Game-O-Matic system made use of concept mapping to match verbs onto game mechanics to create arcade-style games based on human-authored mapping knowledge (Treanor et al. 2012). Gow and Corneli (2015) proposed a system to generate small games via amalgamation (Ontañón and Plaza 2010). Permar and Magerko (2013) presented a system to produce novel interactive narrative scripts via concept blending, using analogical processing. The work presented in this paper focuses on a two-dimensional platformer game, a very different domain.

Our work is inspired by work in the computer graphics field on probabilistic graphical models that encode style from scene and object exemplars (Kalogerakis et al. 2014; Guerrero et al. 2015; Emilien et al. 2015). In these approaches, 3D scenes are broken into individual objects and parts, with each part and important relationships tagged by a human expert. Categories of these tagged exemplars are then used to train a probabilistic graphical model, representing style as the probability of seeing certain object pairs and their relative relationships. Our approach thus avoids much of the human effort of these systems: categorizing the exemplar input via a clustering technique, tagging individual elements via machine vision, and probabilistically determining important relationships rather than explicitly encoding them. There has been work in blending individual tagged exemplars together based on surface level features of components (Alhashim et al. 2014). Our work focuses on blending the models learned from exemplars rather than individual exemplars, and makes use of structural information for concept mapping.

System Overview

The goal of our work is to develop a computational system capable of generating novel game levels by blending different concepts from the game together. For example, we may wish to generate a level of Super Mario Bros. in which Mario swims through an underwater castle. Our system as

a whole can be understood as containing three parts, operating sequentially. First, our system automatically derives sections of levels from gameplay video and categorizes these sections according to their features. Second, the system derives probabilistic graphical models from each category. At this point in the process, our system can be used to generate game level sections similar to, but different from existing game levels (Guzdial and Riedl 2016). Lastly, our system can blend these learned models together using structural information to produce a final model that can produce creative, novel game levels. We chose the highly regarded, classic platformer Super Mario Bros. to test our approach.

We begin by supplying our system with two things: a set of videos to learn from and a sprite palette as seen in the top of Figure 1a. By sprite palette we indicate the set of "sprites" or individual images used to build levels of a 2D game. For this proof-of-concept we found nine videos representing entire playthroughs of Super Mario Bros. and a fan-authored spritesheet. With these elements the system makes use of OpenCV (Pulli et al. 2012), an open-source machine vision toolkit, to determine the number and placement of sprites in each frame of the video. It then combines frames into *level chunks*, the actual geometry that a frame sequence represents. Level chunks include both the sprite geometry and the length of time the player stays in that chunk. These chunks are then clustered into categories of chunk types as seen in Figure 1b.

Each learned level chunk category is used as the basis for training a probabilistic model, visualized in Figure 1c. The system learns what possible sprite shape "styles" exist in a given category of level chunk, and the probability of relative positions between these shapes. This probabilistic approach makes up for the imperfect nature of machine vision, as mistakes disappear with sufficient data. These learned models are very large, and so the system generates an abstracted graph called an *S-structure graph* for blending as seen at the top of Figure 1d. The structure between sprite shape styles are then mapped from one S-structure graph to another in order to conceptually map elements from one model onto another. These mappings are then used to transform the lower-level, more detailed model into a blended model.

Model Learning

Our system learns a generative, probabilistic model of shape to shape relationships from gameplay videos. Given this paper's focus on blending we give a brief description of the model learning process here, for further detail please see (Guzdial and Riedl 2016). These types of probabilistic graphical models, common in the object and scene modeling field, require a set of similar exemplars as input. These sets are typically categories of 3D models, decided on by a human expert. Given that the input to our system is gameplay video, we must determine (1) what input a probabilistic model should learn from and (2) how to categorize this input in an unsupervised fashion to ensure the required similarity. For the input to our system we define the level chunk, a short segment of a level. For the categorization we make use of K means clustering with K estimated with the distortion ratio

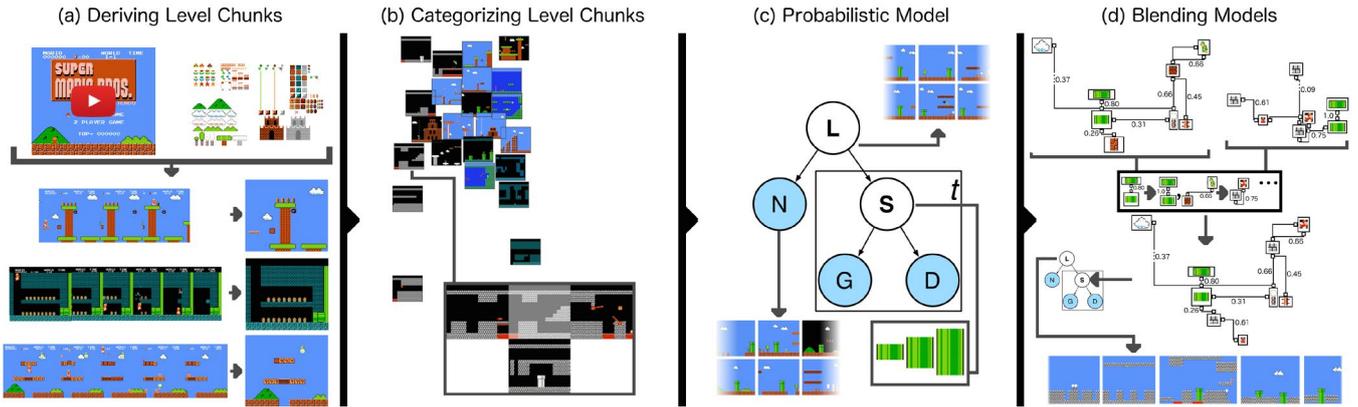


Figure 1: Visualization of the entire process of model building.

(Pham, Dimov, and Nguyen 2005). Each category is then used as input to learn a generative, probabilistic model.

Probabilistic Model

The system builds a probabilistic graphical model from each of the level chunk categories. The intuition for this per-category learning is that different types of level chunks have different relationships, and therefore different models must be learned on an individual category basis. The model extracts values for latent variables to represent probabilistic design rules of a level chunk category. Figure 1c contains a visual representation of the probabilistic model, along with visualizations of three node types. White nodes represent hidden variables, with the blue node values derived directly from the level chunks in a category. Figure 2 represents a final learned model for an individual category along with a representative level chunk.

The three observable nodes are the G node, D node, and N node. The G node represents the sprite “geometry”, an individual sprite shape of sprite type t . Sprite shapes in this case are built by connecting all adjacent sprites of the same type t (e.g. ground, block, coin). These shapes can differ considerably, Figure 3 contains two “block” shapes differing in both orientation and size. The D node represents the set of all relative relationships between a given G node and all other G nodes in its level chunk. The D node in Figure 3 is the set of vectors capturing relative orientation and direction between the question block shape and all other G nodes in the chunk (two block shapes, one goomba shape, and one ground shape). The vectors connect at the cardinal points in order to better represent symmetry in the design. Each D node is paired to a specific G node, as in Figure 3 that visualizes the question block shape’s D node. The N node is the last directly observed variable. It represents the number of individual atomic sprite values in a particular level chunk. In the case of Figure 3 there are two goombas, seventeen ground sprites, etc. Worth noting that the names of these shapes are applied retrospectively, the system reasons over them as images.

The first latent variable is the S node, it represents “styles” of sprite types. These styles can vary either in geometry or

relative position. For example, there are a variety of possible arrangements and positions of pipe bodies, as seen in the lower right of Figure 1c. They can come in groups ranging in *size* from one to four, and can differ in *position*, appearing on top of the ground, on stairs, or out of the bottom of the screen. The system learns the values and number of S nodes by clustering G and D node pairs. By pairs of G and D nodes we mean that each shape is paired with the set of connections from it to everything in its chunk. This process is accomplished by sprite type, meaning that there is at least one S node for each type of sprite. With a fully formed S node we can now determine the probability of an S node shape of a specific type at a given relative distance, given another S node shape. More formally: $P(g_{s_1}, r_d | g_{s_2})$ or the probability of a G node from *within* a particular S node, given a relative distance to a second G node. For example in Figure 3, goomba shapes have a high probability of co-occurring with ground shapes at those same relative positions.

The L Node represents a specific style of level chunk, the intuition behind it is that it is constituted by the different styles of sprite shapes (S) and the different kinds of chunks that can be built with those shapes (N). Once again the system represents this as a clustering problem, this time of S nodes. Each S node tracks the N node values that arose from the same chunk as it’s G and D nodes. Essentially, each S node knows the level chunks from the original Mario that represented its “style” of shape. Figure 2 represents a final learned L Node and all of it’s children. Notice the multiple S nodes of the “block” type, with the singular “ground” S node.

Generation of Novel Level Chunks

L nodes can be used to generate novel level chunks. The generation process is a simple greedy search algorithm, attempting to maximize the following scoring function:

$$1/N * \sum_{i=1}^N \sum_{j=1}^N p(g_i | g_j, r_{i-j}) \quad (1)$$

Where N is equal to the current number of shapes in a level chunk, g_i is the shape at the i th index, g_j is the shape at

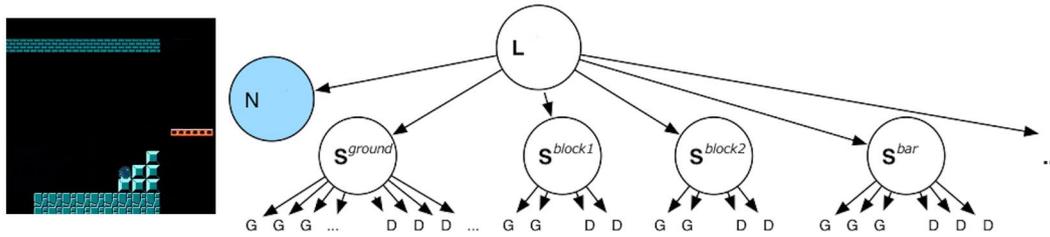


Figure 2: Visualization of a final L Node and one of the example chunks used to train it.

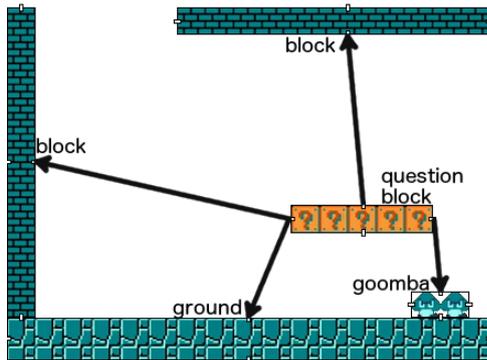


Figure 3: Example of a D Node

the j th index, and r_{i-j} is the relative position of g_i from g_j . This is equivalent to the average of the probabilities of each shape in terms of its relative position to every other shape or the *average sprite probability* of the chunk.

The generation process begins with two things: a single shape chosen randomly from the space of possible shapes in an L node, and a random N node value to serve as an end condition. The N nodes hold count data of sprites from the original level chunks in a category. For example in Figure 4 the top image is a level chunk that informed an N node value with: “blocks: 10”, “pipeTop: 1” and so forth. This N node value can therefore serve as an end condition to the process as it can specify how many of each sprite type a generated chunk needs to be complete.

In every step of the generation process, the system creates a list of possible next shapes, and tests each addition, choosing the one that maximizes its scoring function. These possible next shapes are chosen according to two metrics: (1) shapes that are still needed to reach the N node value-defined end state and (2) shapes that are required given a shape already in the level chunk. For example in Figure 4 from step 1 to step 2 the “pipeBody” shape is added in order to get closer to the end state, while from step 3 to step 4 the “lakitu” enemy is added as the system deems it to be required with the style of pipeTop shape added in step 3. The system defines a shape to require another shape if $p(s_1|s_2) > 0.95$, or if the two shapes co-occur more than 95% of the time. The process ends either because the chunk reaches a sufficient number of sprites as determined by the

N node, or the probability of adding any further shapes is too low ($p < 0.05$).

Blending

The levels generated from a learned probabilistic model tend to resemble the original Super Mario Bros. levels, and while novel, may not be considered creative or surprising. While our model may generate a unique configuration of elements, the types of elements and their individual relationships are drawn directly from the original game. Concept blending serves as a well-regarded approach to produce creative artifacts, but the learned models extracted from gameplay videos are not suited to concept mapping. Instead of using these models directly, our system takes the common concept blending approach and transforms our detailed model into a more abstract model in order to find mappings (Goel 1997). We define this *S-structure graph* as the set of S nodes, styles of sprite shapes in a model, and a set of edges representing probabilistic relative positions between them as seen in Figure 5. In most concept blending systems the abstraction knowledge (e.g. a door and a cabinet are both “openable furniture”) is encoded by a human expert. Instead we can make use of the learned probabilistic relationships

Figure 5 gives an example of a final S-structure graph on the left derived from an L Node trained on level chunks like that on the right. Each box and image represents an S node, the lines between them are D node connections, vectors connecting the cardinal points of the shape styles. The D node connections also have a probability [0...1] corresponding to how likely they are to appear. The S-structure graphs form the basis of structural comparisons between different types of level chunks.

Each S Node has many more D node connections than appear in the S-structure graph. The system uses a subsection of connections equal to the minimum number of connections with the maximum probability to create a fully-connected graph. The system defines a threshold Θ_s for each S node, with a starting value of 1. The system decreases this value iteratively for the current most unconnected node, then adding all the connections of equal or greater probability than Θ_s for each S node to a potential graph. When the graph is fully connected, the process stops.

Concept blending systems typically have a concept of a *source* space and *target* space. Our approach is the same, in that an L node to blend from (source) and an L node to blend to (target) must be selected. Each relationship—D node connection—from the source graph is mapped to

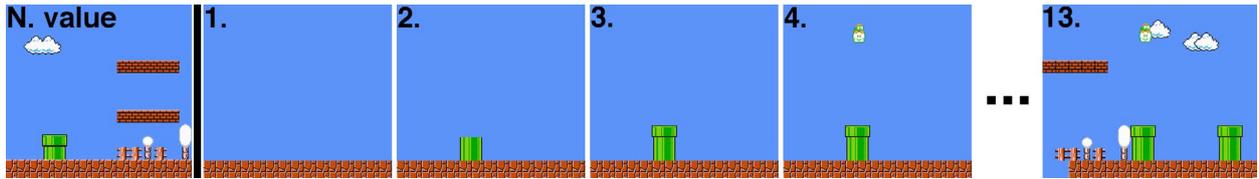


Figure 4: A visualization of the chunk generation process, beginning with an N node value and a single “ground” shape.

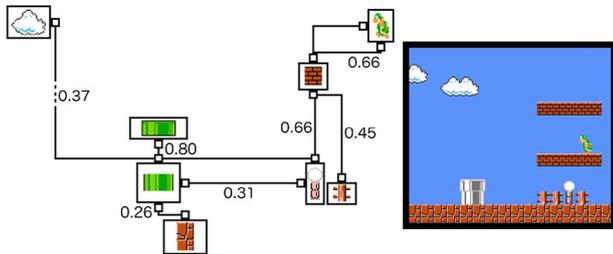


Figure 5: S-structure graph and an example instantiation.

the closest relationship on the target graph. This mapping is a simple closest match, based on a function that equally weights differences in probability with the cosine distance. This list of D node connection mappings can be transformed into a list of S node mappings via referencing the S nodes the relationship exists between. The structural mapping between these relationships therefore serves as a basis for potential S node mappings, with the final S node mappings determined according to the greatest evidence and the target of the blend.

Consider two mapped D node connections from two different S-structure graphs, one representing the relationship between “ground” and “goombas” and the other the relationship between “sea blocks” and “squids”. From these the system can derive the mappings “ground to sea blocks” and “goombas to squids”. The system then takes the final mappings with the greatest evidence. Rather than map *all* of the S nodes from one probabilistic model to another, the system can specify a *target* for the blend, a desired final set of S nodes, and the system can choose only the mappings that fit this final set. For example, if our desired final set was “sea blocks, squids, and goombas” then the system could accept the mapping ground to sea blocks, but not goombas to squids. This final mapping is then used to transform the *source* L node, which means changing N, and S node values within the L node. For example, if previously there existed a relationship between goomba and ground, there would now exist a relationship between goomba and seablock.

Evaluation

In this section we present results from two distinct evaluations meant to demonstrate the utility of our system. The first evaluation is a human study that demonstrates that our probabilistic graphical model captures humans’ intuitions of level design style. The second evaluation is a case study, demonstrating that our system’s blended models can explain

Table 1: The results of comparing our system’s rankings and participant rankings per question.

Category	r_s	p
Style	0.6115095	2.2e-16
Design	0.51948	2.2e-16
Fun	0.2729658	3.745e-5
Frustration	-0.4393904	6.79e-12
Challenge	-0.387222	2.351e-09
Creativity	-0.1559725	0.02007

human-created expert blends significantly better than the un-blended model.

Model Evaluation

The first evaluation shows that the models learned by our system capture human design intuition. We do this by showing that Equation 1 scores Super Mario Bros. levels similarly to humans.

We ran a human subjects study in order to obtain human level rankings to compare to our system rankings. In the study, individuals played through a series of three levels in the vein of Super Mario Bros., the first of which was always a level from the original Super Mario Bros., while the other two levels were chosen randomly from a set of fifteen novel levels. The fifteen novel levels were generated from three generators: the Snodgrass and Ontañón (2014) generator, the Dahlskog and Togelius (2014) generator, and our own generative system. After playing all three levels subjects were asked to rank the three levels they played on measures of style (defined as more “mario-like”), design, fun, frustration, challenge, and creativity. If our hypothesis is correct, we’d expect to see the human ranking of levels correlate strongly with our system’s predicted rankings of these levels based on our system’s level chunk scoring function (Equation 1: the “average shape probability”).

In order to use the scoring function in Equation 1 for entire levels we broke each into chunks of uniform length, randomly selected from these chunks to ensure equally sized distributions, and then used the maximum scoring L node to score each chunk. This gave a distribution of scores over an entire level, and we then determined an absolute ranking of levels according to the median values of these distributions. Our system used this total ordering to predict how a human subject might rank any triplet of levels.

We ran this study for two months and collected seventy-five respondents. We compared the participant rankings and our system’s predicted rankings with Spearman’s Rank-



Figure 6: World 9-1 from the game Super Mario Bros.: Lost Levels

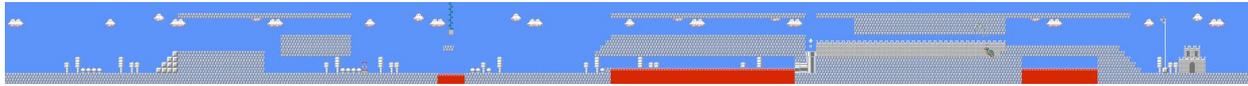


Figure 7: World 9-3 from the game Super Mario Bros.: Lost Levels

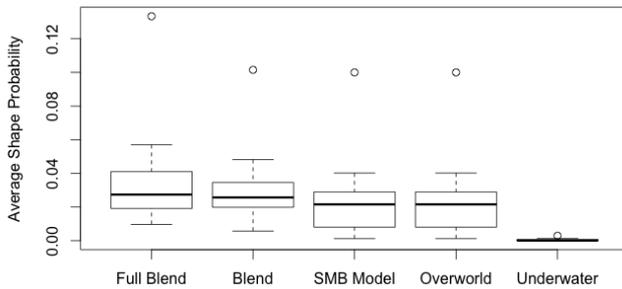


Figure 8: Score distributions from evaluating World 9-1

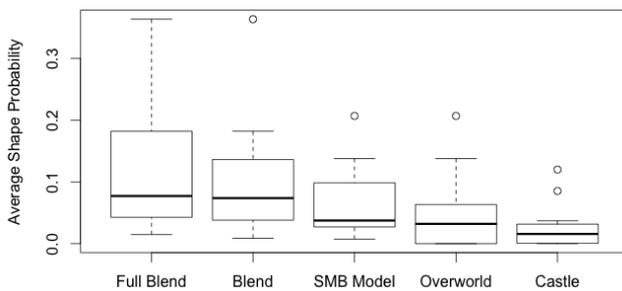


Figure 9: Score distributions from evaluating World 9-3

Order Correlation. Table 1 summarizes the results with significant p-values and correlations in bold.

The strongest correlation present is for the style rankings, which provides strong evidence that our model captures stylistic information. The other correlations can be explained as a side-effect of our model training on the well-designed Super Mario Bros. levels. The very weak correlation between the creativity rankings and our system’s rankings is likely due to the lack of a strong cultural definition of creativity in video game levels. The respondent ranking distributions on a per-generator basis did not differ significantly, further suggesting that this interpretation is accurate, as otherwise we’d expect to see some generators creating more “creative” levels than others.

Blending Evaluation: Lost Levels

The evaluation of blending techniques is a traditionally difficult problem due to the subjective nature of blend quality. Given that our blended models are generative, we could run

a human study on levels generated from these models. However, our initial human study demonstrated that human subjects do not tend to agree on the creativity of a level, indicating that this type of study would be inconclusive. Learned models can also be used for *evaluation*, thus an alternative way to determine the quality of our blended models is to determine how well they account for human-expert blends.

In the case of Super Mario Bros., the designer Shigeru Miyamoto designed a second game known as Super Mario Bros.: Lost Levels based on the original game. The game includes levels that can be understood as blends of Super Mario Bros. level types. World 9-1 (Figure 6) uses a combination of sprites found otherwise only separately in “underwater” and “overworld” levels. The level includes castles, clouds, and bushes that only appear in overworld levels appearing with coral and squids. World 9-3 (Figure 7) on the other hand uses a combination of sprites otherwise found only separately in “castle” and “overworld” levels. The level includes elements from overworld levels alongside lava and castle walls. Due to their “blended” nature, we hypothesize that our blending technique can create models that explain these human blends significantly better than our original, unblended models trained on the Super Mario Bros. levels. That is, how well do the actual relationships between sprites in Lost Levels match the predicted relationships in our models. To rank these levels with our system we used the same strategy as our earlier model evaluation, sectioning off each level into uniform chunks and evaluated each chunk with a set of learned models.

We created four different versions of our system to create four distinct types of learned model:

- **SMB Model:** The Super Mario Bros. (SMB) model represents the set of L nodes learned from gameplay video of the original game.
- **Blended Model:** To construct a blended model the system first chooses what of the original L nodes to blend. The system constructs this initial set by choosing the L node that maximally explains each uniform chunk of the blended level. The system then blends each each pair of L nodes in the set as both the source and target L node using the blended level as the target for the blend. This model can be thought of as an unsupervised model.
- **Level Type Model:** We constructed additional models via hand-tagging each L node with it’s level type. For example, “Overworld” to represent the above ground lev-



Figure 10: A high-quality blended level according to the model targeting World 9-1.



Figure 11: A high-quality blended level according to the model targeting World 9-3.

els, “Underwater” and “Castle”. These models represent subsections of the larger SMB model. We parsed each blended level with the level type models that made up its blend. World 9-3 (Figure 7) was therefore parsed with the “Overworld” and “Castle” models.

- **Full Blended Model:** We constructed the largest possible blended model for each level as a “full” blended model. We constructed this model by taking all of the L nodes tagged with the two level types for each blended level, and blending all of the L nodes together for all possible pairs, leading to a massive final blended model. This model served as an upper-bound of performance for our blending technique given human knowledge of level types, and can therefore be considered a supervised model.

Figure 8 summarizes the results of the evaluation for World 9-1. While 9-1 is made up of a combination of “overworld” and “underwater” level sprites, it is much more overworld than underwater with a 6:1 ratio of sprites from each type. The models reflect this, with the Underwater level type model doing very poorly at explaining the level, while the SMB and Overworld level type models behave essentially the same. Despite this low quality blend, the blended model’s distribution differs significantly from the SMB Model distribution according to the paired Wilcoxon-Mann-Whitney test ($p=0.03327$). In addition the blended model and full blended model distributions do not differ significantly ($p > 0.05$), indicating that the system’s choice for L nodes to blend is as good as creating all possible blended L nodes in this case. It is worth noting that the SMB model typically finds median scores for actual Super Mario Bros. levels between 0.1 and 0.2, with the lowest median score for any level being 0.05. None of these models reaches even the lowest point, but we contend this is due to the fact that the level does not represent a strong blend.

Figure 9 summarizes the results of this evaluation for World 9-3. World 9-3 represents a much more even blend than World 9-1 with an overworld to castle sprite ratio of 3:1. This is reflected in the relative distributions of the Castle and Overworld level type models. Once again the blended model distribution differs significantly from the SMB Model distribution ($p=0.0008308$). In this case the full blended model also differs significantly from the blended model ($p=0.002961$). However, despite the overall higher distribution, the full blended model’s median value rose only a small amount compared to the blended model’s median (0.077 vs 0.074). The full blended model is also made up of over two-

hundred L nodes as opposed to our system’s blended model of twenty-four L nodes. We therefore contend that our system picked out the most important L nodes to blend. In addition, both blended models’ distributions fell into the range of an actual Super Mario Bros. level. We contend this is due to the level being a more even blend, indicating that our blending technique leads to blended models close in quality to those models trained directly on exemplar levels.

Example Output

We present a set of illustrative generated levels from our system. To create full levels our system determines the sequence of L nodes that best explains the sequence of uniform chunks of a target level. Each L node in this sequence is then prompted to generate a novel level chunk and the sequence of generated chunks constitutes a level. Figure 10 and Figure 11 represent high-quality levels (according to our system) using a blending target of World 9-1 and 9-3 respectively. In comparison we present Figure 12 representing a lower quality blended level, and Figure 13 representing a high-quality level generated by the *full* blend model. The difference between the low and high scoring levels should be clear from their structure, with Figure 12 including individual, oddly placed blocks and a floating castle. We further identify a lack of difference between the blend and *full* blend models, with Figure 11 and 13 appearing very similar.

Conclusions

In this paper we’ve presented techniques to learn probabilistic models from gameplay video and to blend these models to produce novel level types. We ran a human subjects study to evaluate our model’s ability to capture level design style as a measure of structural likelihood. We found strong evidence for this in the form of a strong correlation between participant’s ranking of style and our system’s rankings. We demonstrated via two case studies that our system is able to explain human expert blended levels, and is able to blend models that evaluate these levels significantly better than the unblended models. Taken together, these represent a system that is able to learn about design, evaluate design like a human, and is able to extend this knowledge to explain new domains via concept blending.

Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. IIS-1525967.



Figure 12: A lower quality blended level according to the model targeting World 9-3.



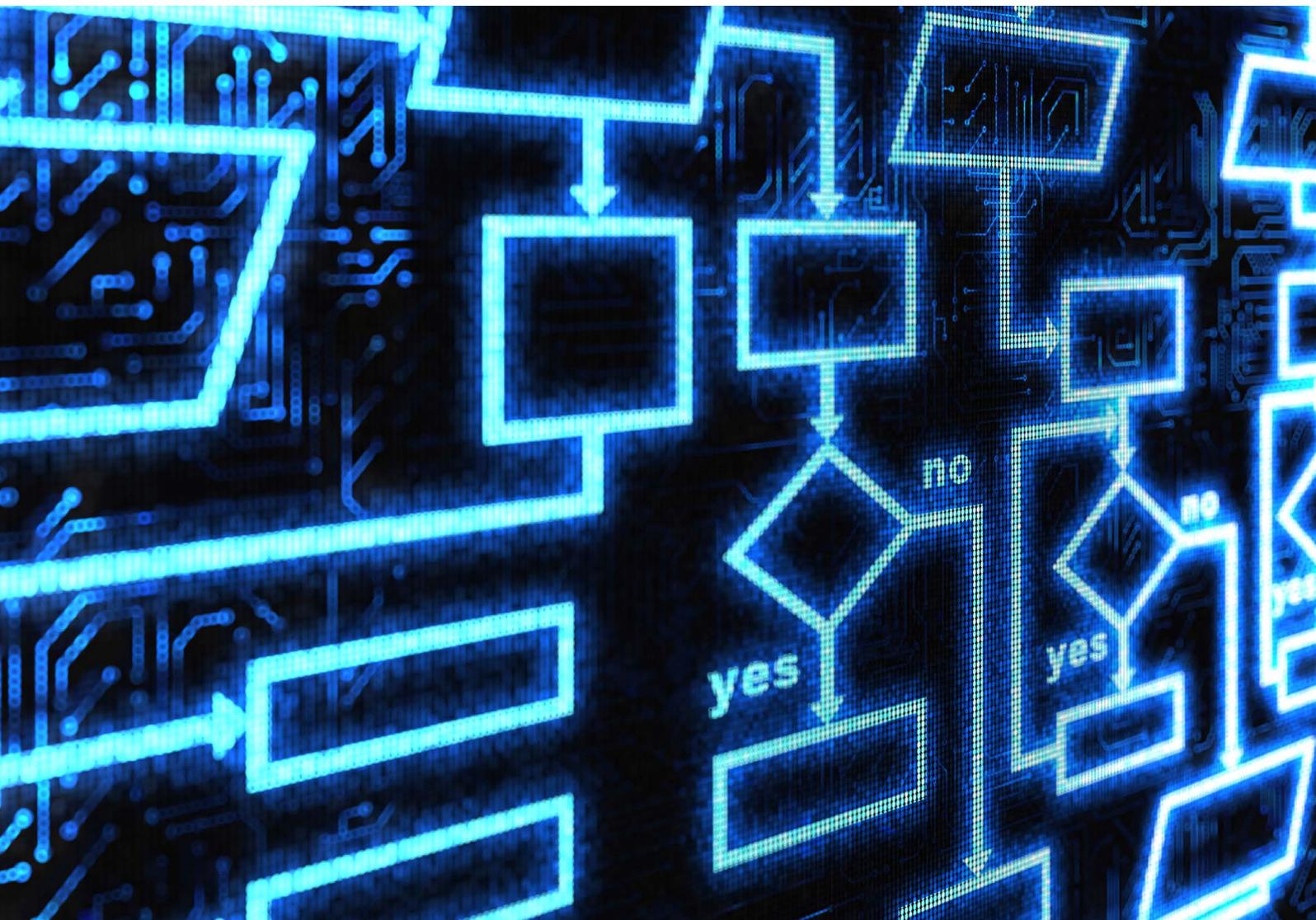
Figure 13: A high quality blended level generated by the *full* blend model targeting World 9-3.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

- Alhashim, I.; Li, H.; Xu, K.; Cao, J.; Ma, R.; and Zhang, H. 2014. Topology-varying 3d shape creation via structural blending. *ACM Transactions on Graphics (TOG)* 33(4):158.
- Bou, F.; Schorlemmer, M.; Corneli, J.; Gómez-Ramírez, D.; Maclean, E.; Smaill, A.; and Pease, A. 2015. The role of blending in mathematical invention. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 55.
- Dahlskog, S., and Togelius, J. 2014. A multi-level level generator. In *2014 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE.
- Emilien, A.; Vimont, U.; Cani, M.-P.; Poulin, P.; and Benes, B. 2015. World-brush: Interactive example-based synthesis of procedural virtual worlds. *ACM Transactions on Graphics* 34(4):106:1–106:11.
- Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial intelligence* 41(1):1–63.
- Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive science* 22(2):133–187.
- Fauconnier, G., and Turner, M. 2002. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.
- Goel, A. K. 1997. Design, analogy, and creativity. *IEEE expert* 12(3):62–70.
- Goel, A. K. 2015. Is biologically inspired invention different? In *Proceedings of the 6th ICCG*, 47.
- Gow, J., and Corneli, J. 2015. Towards generating novel games using conceptual blending. In *Proceedings of the 11th AIIDE Conference*.
- Guerrero, P.; Jeschke, S.; Wimmer, M.; and Wonka, P. 2015. Learning shape placements by example. *ACM Transactions on Graphics* 34(4):108:1–108:13.
- Guzdial, M., and Riedl, M. 2016. Toward Game Level Generation from Gameplay Videos. *arXiv:1602.07721*.
- Kalogerakis, E.; Chaudhuri, S.; Koller, D.; and Koltun, V. 2014. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics* 31(4):55:1–55:11.
- Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *Proceedings of the 5th International Conference on Computational Creativity*, volume 4.
- Martins, J.; Pereira, F.; Miranda, E.; and Cardoso, A. 2004. Enhancing sound design with conceptual blending of sound descriptors. In *Proceedings of the 1st joint workshop on computational creativity*.
- O'Donoghue, D. P.; Abgaz, Y.; Hurley, D.; and Ronzano, F. 2015. Stimulating and simulating creativity with dr inventor. In *Proceedings of the 6th ICCG*.
- Ontañón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *Case-Based Reasoning. Research and Development*. Springer. 257–271.
- Permar, J., and Magerko, B. 2013. A conceptual blending approach to the generation of cognitive scripts for interactive narrative. In *Proceedings of the 9th AIIDE Conference*.
- Pham, D. T.; Dimov, S. S.; and Nguyen, C. D. 2005. Selection of k in k-means clustering. In *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 103–119.
- Pulli, K.; Baksheev, A.; Korniyakov, K.; and Eruhimov, V. 2012. Real-time computer vision with OpenCV. *Commun. ACM* 55(6):61–69.
- Ribeiro, P.; Pereira, F. C.; Marques, B.; Leitao, B.; and Cardoso, A. 2003. A model for creativity in creature generation. In *Proceedings of the 4th GAME-ON Conference*, 175.
- Snodgrass, S., and Ontañón, S. 2014. Experiments in map generation using markov chains. In *Proceedings of the 9th International Conference on Foundations of Digital Games*.
- Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-O-Matic: Generating videogames that represent ideas. In *Proceedings of the 3rd FDG working on Procedural Content Generation*.

SOFTWARE PLATFORMS



The FloWr Online Platform: Automated Programming and Computational Creativity as a Service

John Charnley, Simon Colton, Maria Teresa Llano and Joseph Corneli

Computational Creativity Group, Department of Computing,
Goldsmiths, University of London, UK
ccg.doc.gold.ac.uk

Abstract

We present recent developments in the Flowchart Writer (FloWr) project, where we have built a framework for implementing creative systems as flowcharts of processing nodes. We describe how the system has been migrated from a desktop application to a web portal and document the various features that the portal provides to support Computational Creativity research and development. This includes a node development package and automated chart development assistants. We detail how we have supplemented the online graphical platform with a web service API to enable developers to remotely access the features of FloWr through a programming language of their choice. This encompasses developing systems as flowcharts, together with running flowcharts remotely and also allows developers to publish flowcharts as web services. Importantly, the API allows Computational Creativity researchers to experiment with the automated development of creative software systems. To encourage this, we have also introduced simple models for automated software development into the FloWr API itself, providing a novel system for unsophisticated users to experiment with. We demonstrate the potential benefits of using FloWr, with case studies showing how the web portal has been used for both node and chart development by novice and expert users.

Introduction

In the FlowChart Writer (FloWr) project¹, we have built a platform for all Computational Creativity researchers to produce novel creative systems via GUI-based writing of *flowcharts* which pass information through processing *nodes*. The ultimate aim of the project is to form a community of users contributing to a corpus of nodes and flowcharts, which will enable the automatic generation of flowcharts, hence modelling creativity at the process level.

Since its introduction as a desktop application in (Charnley, Colton, and Llano 2014), there have been many developments with FloWr. One of the most significant changes has been the migration to an online version. FloWr users no longer have to download a huge Java desktop application, no updates are required and a variety of devices can be used to access the system. Nor are users restricted by the computing power of their device as processing is performed

on our servers. Our main motivation for this migration, however, is a desire to provide a platform for interaction between researchers in the Computational Creativity community, by letting them share ideas, processes and resources, and collaborate on creative system development.

The main flowchart writing platform allows high-level creative systems to be developed. Users can also create new flowchart nodes to add novel functionality to the portal, increasing the power and scope of the systems that can be created. Our platform also provides Computational Creativity system design as a service through the FloWr API. This means that researchers are not restricted to the visual GUI and can access the full power of FloWr how they like, from whichever programming environment they choose. We are particularly interested in the possibilities for automated programming that this affords. Another new feature lets users expose flowcharts as standalone web services to allow other systems or users to access them. In the next section, we give details of the portal and highlight improved features of the online system over the previous desktop version. We follow this with two case studies highlighting the potential value of FloWr for novices and experts alike. We conclude with a discussion of future developments for the FloWr platform.

This work has similarities with the ConCreTeFlows project (Žnidarsic et al. 2016), where concept creation workflows have been implemented in a flowchart paradigm. That project uses ClowdfloWS² which, like FloWr, provides a portal for developing and sharing flowchart-based systems. ClowdfloWS was chiefly developed for algorithmic programming in machine learning and data-mining, with appropriate nodes. By contrast, FloWr has been developed for Computational Creativity collaboration and research and we are unaware of any other projects with these specific aims.

The FloWr Web Portal

In addition to improved facilities for writing flowcharts, users have access to a great deal of additional meta-level information describing what nodes and charts do. There is also an *Admin* area which gives access to numerous other enhancements, such as the *API* and *Node Development Package*. We describe the various aspects of the portal below.

¹<http://ccg.doc.gold.ac.uk/research/flowr/>

²<http://clowdfloWS.org/>

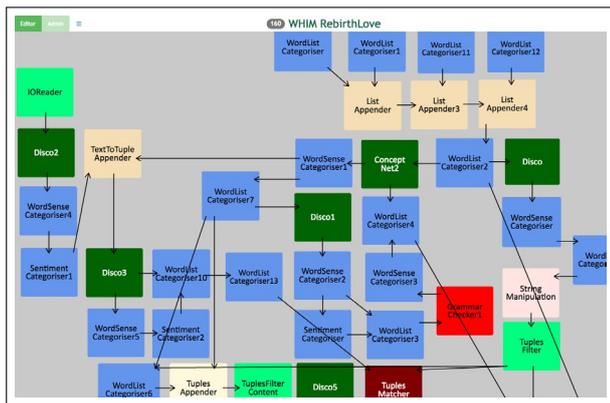


Figure 1: The online FloWr flowchart building interface.

Writing Flowcharts

The main interface, shown in Figure 1, is where users manually craft flowcharts. It is an improved version of the FloWr desktop application (Charnley, Colton, and Llano 2014). Flowcharts consist of nodes, representing self-contained processing elements, and arrows, which indicate how data is passed between nodes during processing. The core features of the desktop version have been retained including adding, removing, moving and re-sizing nodes, defining variables to highlight output, setting parameters, running charts and viewing output. Nodes are still written as stand-alone Java-wrapped code modules and the system still makes use of an underlying script syntax to describe the functional aspects of a chart that are independent from the graphical representation. General usability, feel and use-of-space has been improved by using modern front-end web development frameworks, as has cross-device/platform support. There are also improvements to GUI interaction, such as single-axis resizing and improved chart runtime feedback.

Node Information We have enhanced the information panel that appears on double-clicking a node, with additional information and the ability to open multiple instances. The *information panel* for a node shows the unique node id, its type, a bespoke label and a description of what the node does in the specific context of this chart, as well as tools to alter the node colours. The *input panel* shows, for each parameter, its name (with type tooltip), links to information about that parameter (see below) and a type-specific input control for setting the parameter. Static-value parameter setting has been enhanced to make it simpler and more fault-tolerant. Type-appropriate controls are displayed, such as checkboxes for Boolean values. Automated pre-validation ensures that, for example, numerical parameters are only passed numbers. Node developers can specify additional validation checks, such as maximum values, and bespoke data-types can be used to enforce regex-based validation. Developers can also choose which type of control, should be used for each input parameter, such as a textarea or textbox.

The *output panel* shows a tabbed list of all the defined output variables for the node (elements of its output that have been given specific labels). Output and variable definitions operate in the same manner as for the desktop version, using

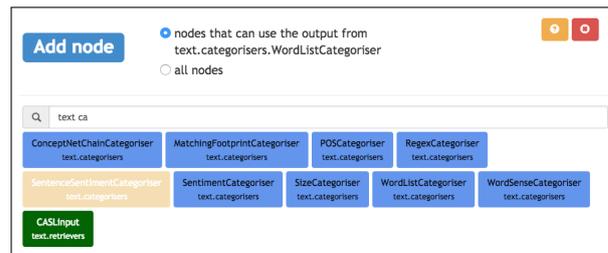


Figure 2: The add-node wizard.

the same syntax. If the user has sufficient access rights – i.e. they own that chart – they can add, delete and amend variables, and we have made error notification clearer and more robust. Once run, any node output can be downloaded in JSON-encoded format. This is particularly useful for testing new nodes in the node development environment, as we describe below. The user can also inspect the output from the Java console, which is useful for debugging node errors and for applications where flowcharts are to be used from shell commands. As in the desktop version, the output of nodes can be locked so that outputs are cached which is useful for chart development and debugging.

Smart Assistance We have introduced wizards to help users create charts, as shown in Figure 2. For example, when the first node is added to a blank chart, the user will, initially, be shown only those nodes that have been tagged by the node developer as suitable for starting charts. Similarly, when a user wishes to create a link between two nodes, by holding down the control key to drag a new arrow between them, a wizard will suggest a subset of nodes which can, in some way, make use of the source node output. Similarly, when a user draws an arrow in to a node from empty space, or vice-versa, the new node wizard will consider which nodes might be appropriate to take/provide data from/to the target. These wizards use Java reflection to find potential data-type matches. Restricting nodes on this basis is very useful, given how many nodes are available, especially when dealing with certain artefact types, e.g., dragging an arrow out of an image retrieval node brings up a wizard showing only image-manipulation nodes. The wizards also provide a comprehensive node text search facilities to find particular nodes or domain-specific packages.

A new *MapHelper* wizard, shown in Figure 3, helps to create and manage data links between nodes, i.e., how data is passed between them at runtime. This appears whenever a new node has been chosen via the new-node wizards, above, or when an existing arrow is selected. This wizard shows the source, or output, node on the left and the target, or input, node on the right. Existing data maps are shown at the top of the dialog. To create new data maps, the user selects a parameter from the right-hand panel, whereupon FloWr uses Java reflection to review all the output of the source node to find any output or variables that match the data-type of that parameter. If the user clicks on one of these suggestions, the wizard establishes the data map, creating a new output variable where necessary.

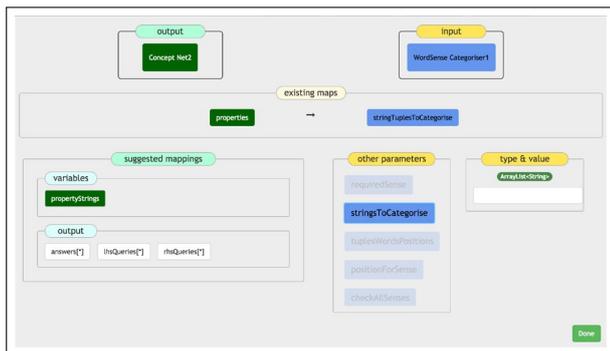


Figure 3: The map helper wizard.

Data Types The desktop version could already handle a large number of different data types and we have continued to expand this. For example, we have recently added images, which are represented internally as Java *Buffered-Images*. These can be large, depending upon the resolution, so thumbnails are used in output panels which give users the option to download the image at any resolution they choose, up to the full version. We have also introduced the notion of bespoke FloWr data-types. These are designed to enforce data-consistency by providing methods to support seamless front-end validation. For example, we have introduced a new *Float01* data-type which can only be instantiated with a string value that represents a valid float value between 0 and 1. The node provides a regular expression for validating front-end input strings accordingly. Node developers can use this to define bespoke data types and take assurance that values input to their node will be valid and consistent.

Menu The menu bar contains a number of new features. For instance, chart loading now includes lists per user and for recent charts. New *history* functionality takes regular snapshots of the chart and allows users to return their chart to a previous state. This also includes a facility whereby the user can take a named snapshot to better control versioning. There are other menu items for clearing the chart output, restarting the user-specific server (useful in debugging), highlighting run nodes and removing locks. The user can view the script underlying the chart and export the whole chart as an XML file. They can also view the supplementary information about the chart provided by the chart owner, as described below. Some of the charts that FloWr users have developed are quite large and contain many nodes. So, we have improved the way in which FloWr handles node positioning and introduced user-specific view profiles, which persist between sessions. The menu provides commands to re-centre the chart and help to find nodes that have been moved off-screen. Auto-layout using Graphviz (www.graphviz.org) has also been implemented.

Help and Information

Every FloWr node or chart has a specific **owner** attribute, which is used to control access and editing rights. Only the owner of a chart can edit it and they can, optionally, lock it to prevent accidental changes. Charts can be *private*, visible to the owner alone, or *public*, where all other users can

view and run, but not amend, that chart. To amend another person's chart, a user must take a copy and use that. Similarly, only node owners may download and make changes to a node's code and information (using the node development package described below). So, only node owners can download and upload new code or rollback versions. Chart Owners can provide an overview of the chart, bespoke node labels and context-specific descriptions for nodes using in-situ editors. Node Owners can provide additional information about nodes, which is available from various buttons next to node types and parameters. This includes an overview of the node, its default colours and whether it can be used to start charts, i.e. generates data from scratch (to inform the first-node wizard). It provides information about the parameters and allows specific input control, default value and validation options to be set. In particular, the node owner can specify drop-down options for a parameter, together with user-friendly replacement labels, if desired. This has replaced clunky source code constructs, which had a number of issues. Information can also be provided about the bespoke output objects, or sub-objects, that the node owner has created for their node.

Implementation

The portal uses a mixture of front-end web technologies, PHP, a relational SQL database and Java. To avoid cross-user data contamination, often caused by Java static variables, each user is given their own java server instance which handles their current chart. This also provides a load-balancing system. User-state is maintained server-side between sessions. Currently users must have a Google account to log in and their account must be unlocked using a code provided by the FloWr team. To maintain state consistency, users can only log in from one browser session at a time. We have taken steps to improve responsiveness by minimising client-server communication, e.g. by bundling calls in specific client use-cases. For speed of execution, chart runs are handled entirely by the back-end Java server with only minimal updates passed to the browser. Output for display is sent to the client piecemeal, with elements transferred only when viewed by the user. The system uses a mixture of sockets and file-system tools to transfer data around and minimise lag.

Admin Area

The admin area handles aspects of the portal that aren't concerned directly with flowchart writing. There are tabs for searching and managing charts, including moving them between the API and GUI or importing from XML. A tab for managing nodes, including downloading/uploading code, rolling back to previous versions and purging unwanted nodes entirely. Note that any changes to nodes must, currently, be carefully managed by node developers to ensure that existing charts aren't broken. The admin area also includes an ever-changing tutorial section. The developer section provides a link to download the developer package (see below). Other tabs in the admin area include recent news and developments. There is a place to provide feedback, a bug tracker for superusers and instructions for using the API.

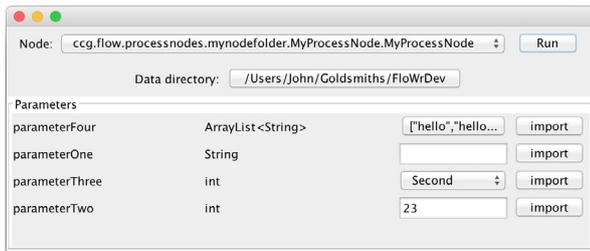


Figure 4: The NodeTester application.

Node Development

We have created a developer package to help users create their own nodes. It includes the `ProcessNode` and `ProcessOutput` Java classes that underlie all nodes and their output (as described in (Charnley, Colton, and Llano 2014)). The package also provides a `NodeTester` application, shown in Figure 4, which lets developers parameterise a node, run it and inspect its output. `NodeTester` allows you to select a node to focus on, from those that you have in the local *processnodes* package. Below this is a `DataDirectory` selector. In the desktop version, this contained large libraries of static data, such as dictionaries or newspaper article archives. Previously users had to download these multi-gigabyte archives to their local machine if they wanted to use all the nodes. The online version keeps all this data on the server, so during development, node owners use a local `DataDirectory` and, when ready for release, an administrator installs the node's static data files on the server.

There is also a panel for setting parameters which uses a JSON format. Whilst not as user-friendly as the online GUI, it is as powerful and allows other data-types, such as lists-of-lists to be defined. Once parameter values have been set, they are stored locally as text. So, they can be edited manually and saved between development sessions. `NodeTester` allows users to import downloaded output from charts (see above) for use as values for parameters. Hence, developers can debug their nodes as though they were part of a larger flowchart, without having to continually upload and test their code changes.

Once a developer is happy with their node, they can upload it through the admin area of the portal. The meta-level node information that owners can provide about their nodes, as described above, is stored in XML format alongside the source code. If the owner desires, they can upload changes to this directly rather than using the online edit functions.

The FloWr Web API

One of the most exciting developments in the FloWr project is the API. This allows users to access the power of FloWr from within a programming language of their choosing. We describe this as Computational Creativity as a Service. Via the API, users can perform almost any of the actions that would be available to the online developer. As with the underlying script syntax, there is no need for notions of chart layout, colours, labels etc., and the API is predominantly functional rather than visual. So, for example, it is possible to set values for parameters but not, say, to re-position a

node in some notion of a screen. Charts created under the API are kept separate from those created through the online GUI (this is changeable in the Admin Charts area). This is chiefly to remove the potential for portal to be swamped by a large number of automatically-created API charts.

The API is accessed by POST requests, which must specify the user's temporary 24-hour access token. As well as creating and editing charts, the API allows the caller to run them, as they would in the online GUI, and download the output they produce. Some of the commands available include all those for manipulating charts, parameters, variables, data maps etc. They also provide lists of available nodes, parameter information, output information and current chart state including run progress. Chart states are provided in JSON format, listing all the nodes and their parameters in a machine-readable format, similar to both the XML export and the underlying script syntax.

We are particularly interested in encouraging the API to be used for automated programming via automatic flowchart construction. To this end, the experiments in automatic flowchart writing presented in (Charnley, Colton, and Llano 2014) have now been run entirely separately to the core FloWr system, via the API. To foster such research, many of the API commands provide meta-data about the available nodes, their parameter types, output etc., which allows automatic programming approaches to make informed decisions about chart manipulation. In addition, the API has functions for users to upload and syntax-check flowchart scripts and XML charts, which provides another approach to automated programming that researchers could use.

Automation Features

In addition to providing API functionality to encourage automated programming, we have begun to add some automated programming features into the main FloWr GUI portal. In particular, users can call up the automated programming dialog box where they can re-run the simple flowchart construction experiments presented in (Charnley, Colton, and Llano 2014), where we asked FloWr to generate charts for creating poetic couplets from scratch. In the dialog box, the user can select the nodes to be placed into particular places in a template flowchart, sets of values to consider for particular parameters and they can specify a minimum level of output from the chart. They can then ask the system to generate a working flowchart. We are hoping that this demonstration of how FloWr might be used for automated programming could encourage researchers to perform their own experiments via the API, and we plan to greatly expand the online automation aspects.

Case Study 1: A novice user adds a new node

This case study describes the experience of a relatively new user to FloWr (author 4) who is familiar with the general features of the web API, and who wants to contribute to node development. His objective is to wrap one of the commands from the Microsoft Web Language Model API (www.projectoxford.ai/webml) in a FloWr node, and use it in a sample flowchart. The command to be encapsulated is

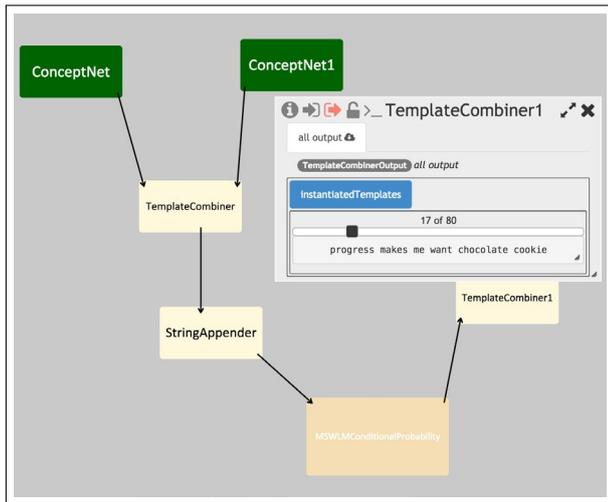


Figure 5: “Progress makes me want chocolate cookie”: Simple interaction with a flowchart containing the new MSWLMConditionalProbability node

Conditional Probability that rates how likely it is that a particular word/phrase will follow a given sequence of words.

The user easily finds the FloWrDev.zip file inside the development area of FloWr. This contains the NodeTester toolkit, including FloWrDev.jar, a file of INSTRUCTIONS, and an outline of a sample node. The sample node does not have a lot of detail, so the user navigates to the *Nodes* tab and types “api” into the search box to track down code he had written earlier. There is a suitable existing node available as a template for modification, namely the node that wraps FloWr’s own API. He downloads FloWrAPI.zip, which contains detailed and relevant sample code.

The sample files are renamed, placed in an appropriate local directory, then easily compiled and run under Java within the FloWrDev.jar environment. Java then displays a minimal panel for setting parameters and running the toolkit (Figure 4). After some further investigation, the user decides that he needs to track down another worked example that formats its output in a way that is more suited to the problem at hand. He returns to the web API and locates a node that is known to have output with suitable format, but this was created by another user so the *Get code* option is greyed out and unavailable. This time, he had another way to track down the code, but he submits a feature request asking for more public example nodes. The web UI has a *Feedback* where this sort of request can be made.

Adapting the FloWr API node to work with the Microsoft API, rather than the FloWr API itself, is straightforward. With the additional example in hand, formatting the node’s output correctly is also easy. Following the strategy used in the example, a third file is added to support a supplementary class that organises output variables. During the adapting of the node, the toolkit gives access to standard Java debugging information and FloWr-specific runtime error messages.

The next step is to upload the code. On the *Nodes* tab in the web interface there is a button for this. Files are selected one by one using a file chooser. After some processing, the new node becomes available via the dropdown node chooser. Clicking on the *Description* button opens up an interface whereby documentation for the node and each of its input parameters and output variables can be added. Along with a plain-text description, this interface also displays the Java type information (which cannot be altered at this stage), and an interface that provides the ability to add or modify default settings and multiple drop-down options for the node. In this case, the user just fills in the plain text descriptions.

The node is now available for further experimentation. Although the NodeTester allows the user to try out various parameter settings for a single node, this is the first experiment with the node in context. Hints are provided (e.g., FloWr knows that a chart should typically begin with a node that retrieves text for further processing). It takes the user about half an hour to explore the available nodes and choose some that make a convincing demo (Figure 5). This flowchart uses a ConceptNet (Liu and Singh 2004) node to come up with food combinations using the template: [x HasProperty edible] + [x Any eat]. It concatenates these, and then uses the new node to decide which word combinations are likely to follow the word “eat”. It then uses another ConceptNet node to select a putative cause: [x IsA change]. Finally, it combines the answers using a template combiner node: b1Texts[*] makes me want b2Texts[*]. Note that b1Texts and b2Texts here refer to the inputs to the template combiner. Along with “*progress makes me want chocolate cookie*” (Figure 5), other output with highly probable food items, as rated by by the new node, included “*grow makes me want cheese plate*” and “*become makes me want steak egg*”. Output ranked with low probability included “*become makes me want chocolate chinese restaurant*” and “*progress makes me want dandelion goat*”. In the process of writing this flowchart, the user learns more about how the node-connection Wizard works: `ctrl` + `mouse1` establishes connections between the nodes, and then the Wizard points out which available variables from a given upstream node can be connected to a selected input parameter in the downstream node.

Along the way to this result, a few further ideas came to mind. Firstly, the node could be improved by allowing more input parameters, in order to make more expressive use of Microsoft’s API. Secondly, additional text processing nodes could be created that quickly concatenate ArrayLists together to form n-grams for testing using the API (rather than using TemplateCombiner nodes). These ideas are easily addressed following the patterns described above. In total, the writing of the new node, deploying it within the online FloWr portal, then building, debugging and running a flowchart containing the new node took around 2 hours, which we believe is a reasonable time for a novice user to write a simple generative system. The new node contains 208 lines of code in total, of which approximately 40 are new. The experience was quite satisfactory to the user, who felt it was a good way for him as a novice Java programmer to get practice in the language.

Case study 2: An expert user’s flowcharts

Here we present an account of the use of FloWr to develop a complex flowchart from the perspective of an expert user (author 3) who is not a developer of the core FloWr system (author 1). We also summarise this user’s broader experience with the system. The flowchart we focus on is shown in Figure 1. This flowchart generates fictional ideas to be used, in this instance, in the context of generating original concepts for musical theatre pieces. Research using FloWr for fictional ideation has been carried out for around 2.5 years as described in (Llano et al. 2016). During this time, the expert user has added 41 new nodes to FloWr, which range over utility nodes, natural language processing, and domain specific nodes for fictional ideation and theory formation (Colton, Ramezani, and Llano 2014). An overview of nodes this user added to the system, and an example in each category, is presented in Table 1. A total of 17 flowcharts have been built to support fictional ideation, 9 of which have been released into production to be used in the European WHIM project (www.whim-project.eu) and 8 of them being more experimental. These flowcharts use an average of 35 nodes each, with the most complex flowchart composed of 73 nodes and the most simple one composed of 6 nodes.

Type	New	Example
Utility	11	RunShellScript: runs shell scripts
Natural Language Processing:		
Categorisers	2	POSCategoriser: annotates text with part-of-speech information
Combiners	5	ListAppender: appends two lists
Extractors	1	NamesExtractor: extracts proper names from text
Language	1	GrammarChecker: e.g. converts nouns from singular to plural
Manipulators	2	StringManipulation: changes text case
Matchers	2	TuplesMatcher: matches tuples in specified positions
Retrievers	9	WordNet: retrieves WordNet data
Theory formation	3	HR3: text to HR3 format (Colton, Ramezani, and Llano 2014).
Ideation	5	AudienceModel: ranks fictional ideas

Table 1: New nodes added by the expert user.

The starting point for our examination of the flowchart in Figure 1 is a set of templates that describe pre-defined, general scenarios that are to be completed by specifying either locations, attributes, or characters, etc., that form the foundation for the fictional ideas. These templates also form the building blocks of the flowchart and their structure guides the development effort. An instance of such a template is:

What if a PERSON_TYPE had to learn how to ACTIVITY in order to find true love?

The building blocks here are place-holders for the *person_type* and the *activity*. A ConceptNet node is used as the source of knowledge to retrieve a list of suitable concepts to fill the *person_type* place-holder. This is shown in Figure 6(a) block 1. As can be seen in this figure, various ConceptNet nodes are invoked, which retrieve ConceptNet facts of the form: [x, IsA, human], [x, IsA, occupation], [x,

IsA, person] and [x, IsA, profession], where *x* is the *person_type*. Outliers are removed using *WordListCategoriser* nodes, and the results are appended to a common list through the *ListAppender* nodes. Building block 2 of Figure 6(a) retrieves facts of the form [x, Any, y], where *y* is then filtered to restrict only to verbs, through a *WordSenseCategoriser* node; the results can then be used in the *activity* place-holder. Finally, building block 3 combines these tuples into larger tuples of the form: [x_1 , IsA, \rightarrow , x_2 , \rightarrow , y], where $x_1 \neq x_2$. The template is finally filled in through the *TemplateCombiner* node, producing ideas such as:

What if a banker had to learn how to fix a cat in order to find true love?

After a first version of a flowchart is finished, the output is analysed to decide if further work is required, which is usually the case. In the particular example followed here, two additional modifications were performed. These are shown in Figure 6(b). The nodes in blocks 4 and 5 retrieve representative qualifiers of the x_1 and x_2 concepts – for which there was not much data in ConceptNet. To achieve this, the *Disco* node, a linguistic tool (www.linguatools.de/disco/disco_en.html) that extracts related words using co-occurrences, was used. In particular, the 50 most common collocations for each concept were retrieved, and consequently filtered to keep only adjectives. Finally, block 6 in Figure 6(c) handles the evaluation of the ideas. This is achieved by connecting to an external web service that analyses their narrative potential through a set of measures; the results are subsequently fed into another external web service that contains an audience model that provides a ranking of the ideas. A possible expansion of the idea above, that is ready for this evaluation is:

What if a wealthy banker had to learn how to fix a cat in order to captivate an accomplished veterinarian?

Having access to different linguistic tools as nodes that support the retrieval and analysis of information has provided a useful framework for experimentation. In particular, being able to connect different tools (as illustrated above for the *Disco* and *ConceptNet* nodes) enables easy formation of a dynamic knowledge base to enrich generated output.

As can be seen from Figure 1, we have only explained a small subset of nodes used in the flowchart (we have focused on one of the possible templates). The flowchart deals with a total of 4 templates, each of which produces a different set of fictional ideas using common standard building blocks, contextual information blocks and evaluation blocks. This way of working provides a lot of flexibility. However, it also has the drawback that flowcharts can become cluttered with multiple nodes that could otherwise be encompassed in only one. For instance, if the *ConceptNet* node had either (a) an input parameter that takes a list of RHS or LHS queries in the form of text – it currently accepts these query parameters only in the form of an *ArrayList* of strings – or (b) if an alternative *ConceptNet* node existed with a text parameter, then all of the nodes in block 1 of Figure 6(a) could be replaced by a single node. Such functionality could be easily developed; however, following the line set out in alternative (a),

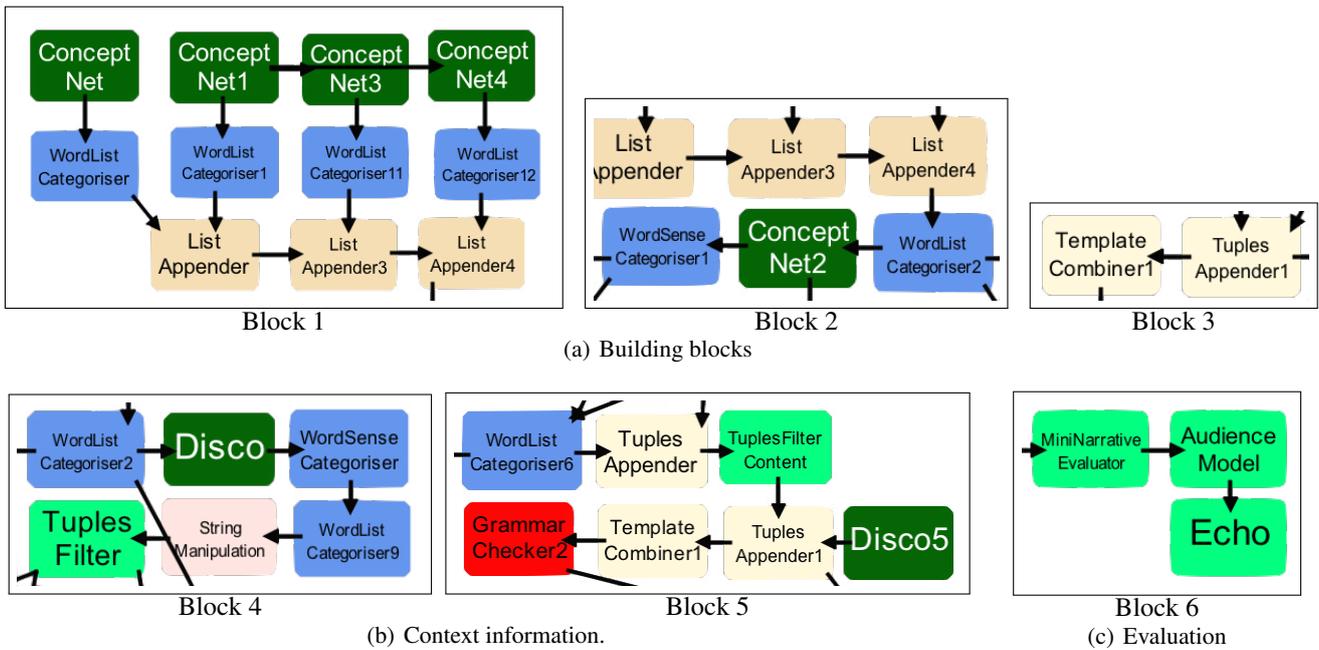


Figure 6: Expert user processes in the fictional ideation flowchart, split into blocks.

nodes can become too complex, with an unmanageable variety of optional parameter combinations, and following the route set out in alternative (b), FloWr would end up with several ConceptNet nodes that only differ in the way the information is retrieved, so it could become difficult to track all of the variants. In any case, complex flowcharts are likely to be difficult to read because of the sheer number of nodes. A future improvement can be accomplished by having nodes that represent entire flowcharts, so for instance, block 1 in Figure 6(a) could be represented as a single node, which could be expanded at will to show the lower layers.

In summary, the experience with FloWr by the expert user has been very satisfactory in the sense of reuse and ease of experimentation. Challenges are present with all frameworks, and from experience of using FloWr in the context of fictional ideation, it is clear that the maintenance of large flowcharts (as in Figure 1) can be difficult, since the connections become confusing. Being able to use hierarchies of flowcharts would be facilitative. The format of data exchange is another challenge. As illustrated above with the ConceptNet node, the situation can be complex either for the developer or the user. We plan to implement a mechanism to handle the complexity of nodes in a flexible framework that can be tailored to different users with different priorities.

Conclusions and Future Work

We have described progress in the FloWr project, the long term aim of which is to study automating creativity at process level through the automatic construction of flowcharts for generative purposes. We have migrated FloWr to an online portal, with all the benefits that this affords, with an aim of creating a platform for interaction between Computational Creativity researchers. In addition, we have described

new powerful additional features of the platform, such as the API, the node developer package and the ability to publish flowcharts as stand-alone web services. We are particularly interested in how the API will help and encourage researchers to experiment with automated programming. To illustrate the potential value of FloWr to Computational Creativity researchers, we presented two case studies. In the first, a novice user created a new node and a flowchart which used it in two hours, showing that the learning curve for the platform is not too steep. In case study 2, the accomplishments and experiences of an expert user showed that FloWr has real potential for building sophisticated creative systems.

FloWr has reached an important milestone in its development. The platform is now mature and performs all of the main functions for which it was built. We still foresee development work for the core system. However, at this point we have decided that, rather than second-guess which new features to introduce, we will focus our efforts on raising awareness of the system and supporting its adoption by new FloWr users by helping them to learn how to use various aspects of the system and contribute to the platform. We will then respond to feedback and direct our efforts accordingly.

Future feature improvements include better looping and conditional processing plus the introduction of new data-handling approaches such as tagging and a global data store during chart running. Although such enhancements should not be of major concern to API users, who can decide to run parts of the chart, and handle output, as they wish. Further improvements could include parameter validation using owner-defined regex checking or cross-parameter checks and better support for pausing and stopping chart runs.

Our automation experiments will continue. In particular, we are hoping that, as the corpus of nodes and flowcharts

grows, we will have an opportunity to machine-learn regularities in the structure of flowcharts, and apply that knowledge to the generation of novel flowcharts.

In order to further support automated programming, it is likely to be important to expand on the bespoke data-type and supplementary meta-information, e.g. minimum values, that FloWr currently supports. Following the methodology of “Design by Contract” (Mitchell and McKim 2002), node authors would be able to make explicit statements of pre-conditions, post-conditions, and invariants. Sophisticated automatic programming clients could then reason about these specifications. There are a range of Java libraries that support this sort of annotation, e.g., the Java Modelling Language (Leavens, Baker, and Ruby 1998). Using such an approach, we plan to see whether an automatic programming tool could rediscover, for example, the patterns used in Figure 6(a)–6(c), and the way they fit together. This would be informed by work on reasoning about formal specifications. There is an established body of work in this area, much of it carried out in connection with theorem provers such as Coq (Bove and Capretta 2007; Tollitte, Delahaye, and Dubois 2012). Reasoning about program syntax and semantics is a recognised challenge “the pragmatic questions in this domain are far from settled” (Chlipala 2011, p. 340), and we hope to contribute to this field with reasoned automation over the FloWr system.

We intend to build on our initial work in encapsulating charts as nodes. We have developed a node which makes calls to the chart-run feature of the API and, so, we have been able to use entire flowcharts as single nodes in other charts. The next stage will be to resolve the issue of interface – i.e. how to re-describe specific parameters of the encapsulated chart as named parameters of the encapsulating node – so that traditional nodes and encapsulating nodes are indistinguishable. This is likely to require some development work in the core FloWr system, as the ability to pass in a list of encapsulated parameters with potentially differing data-types is not clear. Python and Clojure clients implementing the current API functions are available, and will be maintained and extended as the API evolves (<https://github.com/holtzermann17/FloWrTester>).

We want to enhance chart-sharing by introducing a system which, like view profiles, allows viewers of public charts to alter a subset of parameters, rather than having to take a copy. The subset of changeable parameters will be determined by the chart owner so that they can be changed to produce new chart output without breaking the chart. This will have some overlap with our work to encapsulate charts as nodes by introducing a notion of interface. In addition, we would like to enable users to switch between a number of pre-set chart parameterisations. We also plan to further extend the media types that FloWr can handle. The placeholder-viewer approach used for images is very effective and we expect to be able to easily expand FloWr to other multimedia domains, such as audio and 3D models.

The community features of FloWr will evolve. We will introduce sub-groups of users, context-based messaging, and comments and responses. We will supplement the basic public/private visibility system with the ability to open charts up

to subgroups and introduce a similar system for node visibility and versioning. So, for instance, novice node developers will be able to ask for feedback from a select group before releasing their code to a wider audience. We hope that FloWr becomes a hub for Computational Creativity research, education and practice. Illustration with examples is very powerful and the ability to quickly set up a chart to demonstrate concepts with a natural language tool like Porterstemming (Porter 1980), could be highly effective in a range of scenarios. As per case study 1 above, we imagine a researcher who hears about a new piece of research or tool and is able to easily share it with the broader community by implementing it as a node and demonstrating its operation in a flowchart. Such illustrations have the power to be far more informative than, for example, writing up a research note or providing a link to a paper

Acknowledgments

This work has been supported by EPSRC Grant EP/J004049/1 (Computational Creativity Theory), and EC FP7 Grants 611560 (WHIM) & 611553 (COINVENT).

References

- Bove, A., and Capretta, V. 2007. Computation by prophecy. In *Typed Lambda Calculi and Applications*. Springer.
- Charnley, J.; Colton, S.; and Llano, M. T. 2014. The flow framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the International Conference on Computational Creativity*.
- Chlipala, A. 2011. *Certified programming with dependent types*. MIT Press.
- Colton, S.; Ramezani, R.; and Llano, M. T. 2014. The HR3 discovery system: Design decisions and implementation details. In *Proceedings of the AISB symposium on computational scientific discovery*.
- Leavens, G.; Baker, A. L.; and Ruby, C. 1998. JML: a java modeling language. In *Formal Underpinnings of Java Workshop (at OOPSLA'98)*.
- Liu, H., and Singh, P. 2004. Commonsense reasoning in and over natural language. In *Proc. 8th Int. Conf on Knowledge-Based Intelligent Information and Engineering*.
- Llano, M. T.; Colton, S.; Hepworth, R.; and Gow, J. 2016. Automated fictional ideation via knowledge base manipulation. *Cognitive Computation* 1–22.
- Mitchell, R., and McKim, J. 2002. *Design by Contract, by Example*. Addison-Wesley.
- Porter, M. 1980. An algorithm for suffix stripping. *Program* 14(3).
- Tollitte, P.-N.; Delahaye, D.; and Dubois, C. 2012. Producing certified functional code from inductive specifications. In *Certified Programs and Proofs*. Springer. 76–91.
- Žnidarsic, M.; Cardoso, A.; Gervás, P.; Martins, P.; Hervás, R.; Oliveira Alves, A.; Gonçalo Oliveira, H.; Xiao, P.; Linkola, S.; Toivonen, H.; Kranjc, J.; and Lavrač, N. 2016. Computational creativity infrastructure for online software composition: A conceptual blending use case. In *International Conference on Computational Creativity*.

Computational Creativity Infrastructure for Online Software Composition: A Conceptual Blending Use Case

Martin Žnidaršič¹, Amílcar Cardoso², Pablo Gervás⁴, Pedro Martins²,
Raquel Hervás⁴, Ana Oliveira Alves², Hugo Gonçalo Oliveira², Ping Xiao³,
Simo Linkola³, Hannu Toivonen³, Janez Kranjc¹, Nada Lavrač¹

¹Jožef Stefan Institute, Ljubljana, Slovenia

²CISUC, DEI, University of Coimbra, Coimbra, Portugal

³Department of Computer Science and HIIT, University of Helsinki, Finland

⁴Universidad Complutense de Madrid, Spain

Abstract

Computational Creativity is a subfield of Artificial Intelligence research, studying how to engineer software that exhibits behaviors which would reasonably be deemed creative. This paper shows how composition of software solutions in this field can effectively be supported through a Computational Creativity (CC) infrastructure that supports user-friendly development of CC software components and workflows, their sharing, execution and reuse. The infrastructure allows CC researchers to build workflows that can be executed online and be reused by others with a single click on the workflow web address. Moreover, it allows building of procedures composed of software developed by different researchers from different laboratories, leading to novel ways of software composition for computational purposes that were not expected in advance. This capability is illustrated on a workflow that involves blending of texts from different domains, blending of corresponding images, poetry generation from texts as well as construction of narratives. The paper concludes by presenting plans for future work.

Introduction

Computational creativity (CC) systems use as their basic ingredients different types of resources, including musical, pictorial and textual, to name a few. This paper focuses on infrastructure support to CC systems that base their creativity on textual resources. Such CC systems include poetry generation, metaphor creation, generation of narratives, creation of fictional ideas and conceptual blending, which all represent CC tasks which request manipulation of text resources that are provided as inputs.

Infrastructures supporting text-based creative systems are scarce. Ideally, a text-based CC system would automatically build creative artefacts from the given text resources, which the end user would then inspect and potentially adapt to their needs. An attempt in this direction is the FloWr system for automated flowchart construction, optimisation and alteration (Charney, Colton, and Llano 2014). While getting software to write CC code directly is a long-term research goal, that line of research is—with the exception of FloWr—still in its infancy stage. A substantially more mature area of research concerns the development of infrastructures supporting modular development, sharing and execution of code

used in text mining tasks. Text mining has numerous open source algorithms and natural language processing (NLP) software libraries available (such as NLTK (Bird 2006) and scikit-learn (Pedregosa et al. 2011)). However, even text mining and NLP experiments are still difficult to reproduce, including the difficulty of systematic comparison of algorithms. To this end, a number of attempts have been made to develop easy-to-use workflow management systems, allowing users to compose complex processing pipelines in a modular visual programming manner.

Related work As regards the work related to the platform presented in this paper, we first mention myGrid¹ which is used primarily for bioinformatics research, having in mind experiment replication. It is currently probably the most advanced workflow management system, although, due to its complexity, not very easy to use. The most important part of myGrid is Taverna, which is conceived as a suite of tools used to design and execute scientific workflows. A multilingual Internet service platform Language Grid², which is based on a service-oriented architecture and supports a web-oriented version of the pipeline architecture typically employed by NLP tools, is open source, but it is quite complex to install and use. The ARGO platform³ is a more recent development, which enables workflows to have interactive components, where the execution of the workflow pauses to receive input from the user, but ARGO is not open source and does not have sophisticated utilities for cataloguing the available web services or workflows, nor a system of access permissions.

Our recently developed platform CloudFlows⁴ (Kranjc, Podpečan, and Lavrač 2012) is web-based thus requiring no local installation, is simple to use and install, and available as open source under the MIT Licence. While CloudFlows is mainly devoted to data mining, its fork TextFlows⁵ is focused on text mining and NLP workflows. A fork platform for facilitation and reuse of computational creativ-

¹<http://www.mygrid.org.uk/>

²<http://langrid.org/>

³<http://argo.nactem.ac.uk/>

⁴<http://clowdflores.org>

⁵<http://textflows.org>

ity software is called ConCreTeFlows⁶. It is an independent platform with a specific backend that is being continuously adapted to computational creativity tasks and tools. As forks of ClowdFlows, TextFlows and ConCreTeFlows benefit from its service-oriented architecture, which allows users to utilize web-services as workflow components. The distinguishing feature of these platforms is the ease of sharing and publicizing workflows, together with an ever growing roster of reusable workflow components and entire workflows. As completed workflows, data, and results can be made public by the author, the platform can serve as an easy-to-access integration platform for data mining, text mining or computational creativity processes. Each public workflow is assigned a unique URL that can be accessed by anyone to either replicate the experiment, or use the workflow as a template to design new similar workflows.

Contributions In this paper we present ConCreTeFlows and illustrate its use in a specific use case of conceptual blending (introduction to blending theory is provided on page 3). This example employs multiple software components that are being developed by various members of the computational creativity community. The presented composition of software aims to conduct conceptual blending conceptually, textually and visually. Given two descriptions of arbitrary concepts in natural language, the presented approach provides conceptual graph representations of both concepts and their blend, a textual description of the blended concept and even a set of possible visual blends.

The paper is structured as follows. The first section presents ConCreTeFlows as a special purpose workflow management platform aimed at supporting computational creativity tasks. In the next section is the core of this paper. It provides a description of the use case and the basics of its theoretical foundations, followed by presentation of all the important methods and software components that are applied for its purpose. Last part of this section is devoted to critical discussion and ongoing work on the presented components. The paper concludes with a brief summary and plans for further work.

Software Infrastructure

This section briefly describes the main components of the ConCreTeFlows. It is a special purpose workflow management platform, aimed at supporting (primarily text-based) computational creativity tasks.

Like ClowdFlows, ConCreTeFlows can also be used in a browser, while the processing is performed in a cloud of computing nodes. The backend of ConCreTeFlows uses Django⁷, which is an open source web framework. The graphical user interface is implemented in HTML and JavaScript, using jQuery⁸ and jQuery-UI⁹ libraries. ConCreTeFlows is easily extensible by adding new packages

and workflow components. Workflow components of several types allow graphical user interaction during run-time, and visualization of results by implementing views in any format that can be rendered in a web browser. Below we explain the concept of workflows in more detail and describe the basic concepts of ConCreTeFlows.

The workflow model is the main component of the ConCreTeFlows platform and consists of an abstract representation of workflows and workflow components. The graphical user interface for constructing workflows follows a visual programming paradigm which simplifies the representation of complex procedures into a spatial arrangement of building blocks. The basic unit component in a ConCreTeFlows workflow is a processing component, which is graphically represented as a widget. Considering its inputs and parameters every such component performs a task and stores the results on its outputs. Different processing components are linked via connections through which data is transferred from a widget's output to another's input. An alternative widget input for a widget are parameters, which the user enters into widget's text fields. The graphical user interface implements an easy-to-use way of arranging widgets on a canvas to form a graphical representation of a complex procedure. Construction of new workflows thus requires no expertise, apart from knowing (usually from widget documentation) the inputs and outputs of the widgets to ensure their compatibility. Incorporation of new software components, on the other hand, requires basic programming skills in Python or SOAP web-service development in any programming language.

ConCreTeFlows implements its own workflow execution engine. Currently there are no ways to reuse the workflows using third party software. We plan to implement special widgets that will define inputs and outputs for REST API endpoints which will allow execution of workflows on variable inputs by any third party software.

The ConCreTeFlows graphical user interface is shown in Figure 1. On the top of the graphical user interface is a toolbar where workflows can be saved, deleted, and executed. Underneath on the left is the widget repository, which is a list of available widgets grouped by their functionality. Click on a widget in the repository adds it to the workflow construction canvas on the right. A console for displaying success and error messages is located on the bottom.

Workflows in ConCreTeFlows are processed and stored on remote servers from where they can be accessed from anywhere, requiring only an internet connection. By default each workflow can only be accessed by its author, although one may also chose to make it publicly available. ConCreTeFlows generates a specific URL for each workflow that has been saved as public. The users can then simply share their workflows by publishing the URL. Whenever a public workflow is accessed by another user, a copy of the workflow is created on the fly and added to his private workflow repository. The workflow is copied together with widgets' parameter settings, as well as all the data, in order to ensure the experiments can be repeated. In this way the user is able to tailor the workflow to his needs without modifying the original workflow.

⁶<http://concretedeflows.ijs.si>

⁷<https://www.djangoproject.com>

⁸<http://jquery.com>

⁹<http://jqueryui.com>

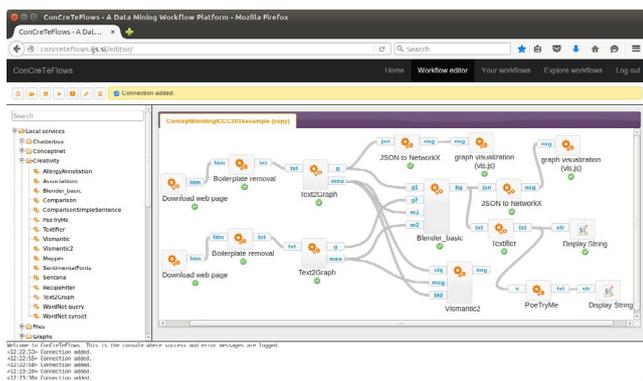


Figure 1: A screenshot of the ConCreTeFlows graphical user interface opened in the Mozilla Firefox Web browser, presenting a motivational CC use case.

Conceptual Blending Online

The elements of the conceptual blending (CB) theory (Fauconnier and Turner 2002) are an inspiration to many algorithms and methodologies in the field of computational creativity (Veale and O'Donoghue 2000; Pereira 2005; Thagard and Stewart 2010; Schorlemmer et al. 2014). A key element in the theory is the *mental space*, a partial and temporary structure of knowledge built for the purpose of local understanding and action (Fauconnier 1994). To describe the CB process, the theory makes use of a network of four mental spaces (Figure 2). Two of these correspond to the *input spaces*, i.e., the content that will be blended. The process starts by finding a partial *mapping* between elements of these two spaces that are perceived as similar or analogous in some respect. A third mental space, called *generic*, encapsulates the conceptual structure shared by the input spaces, generalising and possibly enriching them. This space provides guidance to the next step of the process, where elements from each of the input spaces are *selectively projected* into a new mental space, called the *blend space*. Further stages of the process elaborate and complete the blend.

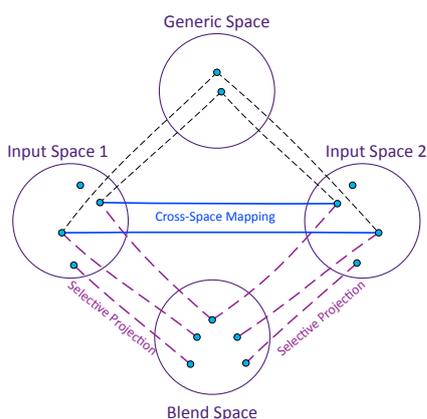


Figure 2: The original four-space conceptual blending network (Fauconnier and Turner 2002).

In most computational approaches to CB, the input and blended spaces are represented as computational versions of *Conceptual Maps* (Novak 1998), i.e., graphs where nodes are concepts and arcs are relations between them (see Figure 3).

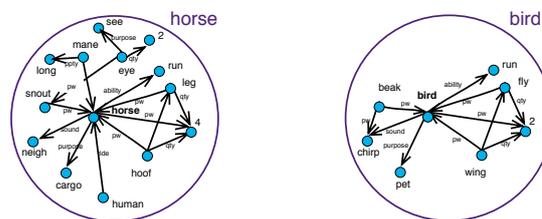


Figure 3: Concept maps of horse and bird.

Graph representations of concept blends are useful for automated analysis and further processing, but are not very suitable and appealing for human perception of the blended spaces. To improve on this aspect of conceptual blending, we have developed methodologies and algorithms for visual blending and for textual representation of concept graphs. Using these new techniques, we designed a CB process that results in conceptual blends that are described in natural language and enriched with visual representations. The process is sketched in Figure 4, where the boxes represent the main (software) components and the arrows indicate the flow of data from inputs to outputs.

Each of the main process components (that is, each box from the sketch in Figure 4) is implemented as an independent software solution and represented as a widget or a group of widgets in ConCreTeFlows.

In the following, we describe the process components and their implementations in detail, with a presentation of the whole workflow and some exemplary results at the end.

Construction of Conceptual Networks

The *Concept Network Builder* component from Figure 4 accepts a textual description of a concept in natural language and on its basis produces a conceptual graph. The set of possible concepts and relations in the resulting graph is open and not limited to a particular fixed set (such as relations in ConceptNet¹⁰) or linguistic characteristic. We decided to represent also relations as concepts, which allows treating a particular entity as both a concept or relation, depending on the context. For example, the concept of *eating* can be used to relate the concepts of *cows* and *grass*, but it can also be a concept related through *is a* with *animal activity*.

The software component that creates these conceptual graphs from text is implemented in ConCreTeFlows as the Text2Graph widget. This component first uses the Ollie triplet extractor (Mausam et al. 2012) to extract the triplets from the given text. The only text transformation before triplet extraction is uncapitalization of sentences. The resulting triplets are used to create a graph. In this process, the entities in the triplets can be lemmatized (this choice is

¹⁰<http://conceptnet5.media.mit.edu/>

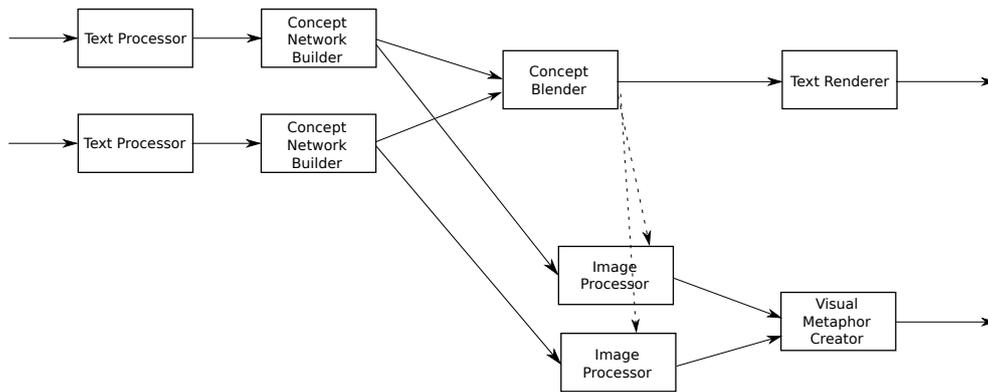


Figure 4: Sketch of the workflow for conceptual blending with visual and textual representations of blends.

left to the user). If the *Main size* parameter of the resulting graph is set, the graph is filtered to contain only a limited neighborhood around the *Main entity*, that is, the node in the graph that a user might be most interested in. The *Main entity* can be set by the user, but if it is not, it is selected automatically as the graph node with the highest out-degree.

Conceptual Blending

The *Concept Blender* component takes care of blending two elements that are represented as graphs. In our design of the process (Figure 4), the mapping between the elements from the input spaces is to be done by a human as an initial step (to select the two inputs to take part in blending) or to be done in the *Concept Blender* on all possible pairs of elements from the two input spaces. Following our representation, these would be pairs of conceptual graphs.

In the current baseline implementation, the *Concept Blender* expects only one pair of input elements, which it fully blends (merges the two conceptual graphs) without any influence of the *Generic Space*. An elaborate concept blending component, that is based on Divago framework, is in development as described in subsection on further work.

Visual Blending

In order to generate visual blends, the visual module, consisting of the *Image Processor* and the *Visual Metaphor Creator* in Figure 4, takes inputs from either the *Concept Network Builder* or the *Concept Blender*. From the first, it can take two concepts from two concept graphs (in this version, their main entities) as inputs to be visually blended. The resulting visual blend is not a representation of a blend created by *Concept Blender*, but an independent visual blend of the two concepts. For this purpose the visual module first finds photos tagged with the two concepts in Flickr, respectively. For ensuring the relevance and quality of the photos, we use a set of image analysis methods bundled together in QualiPy¹¹. The image processor separates the subject and the background of each photo, and inpaints the background

¹¹<https://github.com/vismantic-ohutuprojekti/qualipy>

to hide any marks of the subject. The visual metaphor creator implements three visual operations: juxtaposition, replacement and fusion, as described by Xiao and Linkola (2015). In effect, it puts one object in the context of another, or gives an object the texture of another object (see Figure 5 for an example).

The visual module can also take input from the concept blender, which indicates a specific way of blending. Specifically, the input may indicate a choice between the replacement and fusion operations. For instance, a frequent conceptual blend is placing an object in an unusual environment, which suggests that the replacement operator shall be used.

In ConCreTeFlows, such blending is available in two versions (generations) as Vismantic and Vismantic2 widgets.

Text Generation

In order to generate a textual description of the blends obtained, a *Text Renderer* widget called Textifier has been added to the workflow. Textifier is a natural language generation tool that transforms data represented in a graph into a natural language text. It carries out stages of content determination, document planning and surface realization (Reiter and Dale 2000) and then translates the result into plain text. Content determination processes input to select and adapt what might be rendered. The input graph must contain pairs of source and target nodes with information, and the system will create all possible paths and represent them in a tree. Textifier first groups related information that refers to the same concept by combining nodes that contain the same subject and discarding duplicated nodes. Nodes that represent information with granularity inappropriate for textual rendering – such as verb-preposition groups represented as single strings – are rewritten to make all information explicit in the knowledge structure. Lastly, Textifier can prune the tree if only branches of a certain length need to be considered. Currently we are working with branches that are three nodes long, after detecting that they tend to contain more promising information. Document planning is basic at present but will play a larger role once the graphs of blends are processed. The surface realization stage transforms the tree into text. Figure 6 shows an example of Textifier in operation over a graph constructed from a given input text.

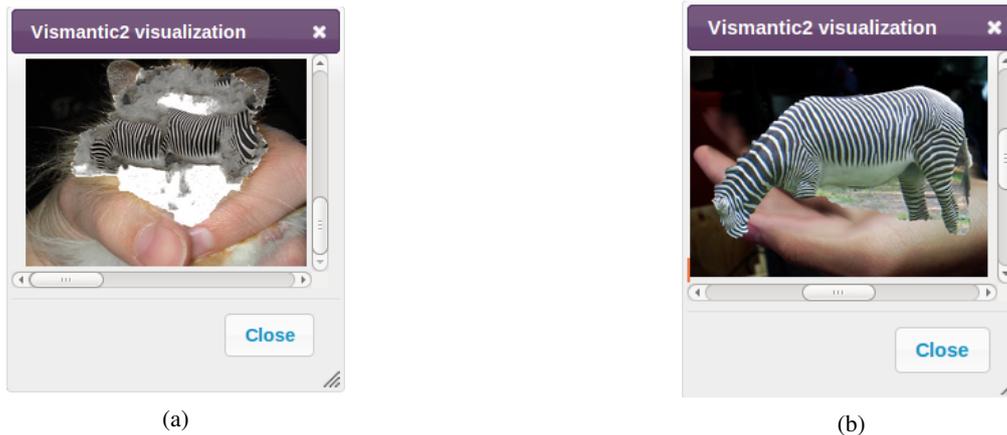


Figure 5: Two (out of four combinations of exchanging context and texture) outputs of the Vismantic2 widget for the example of blending the concepts of *hamster* and *zebra*. Figure 5a shows result of exchanging texture: hamster with a zebra's texture. In Figure 5b is an example of exchanging context: zebra is put in the usual visual context of a hamster.

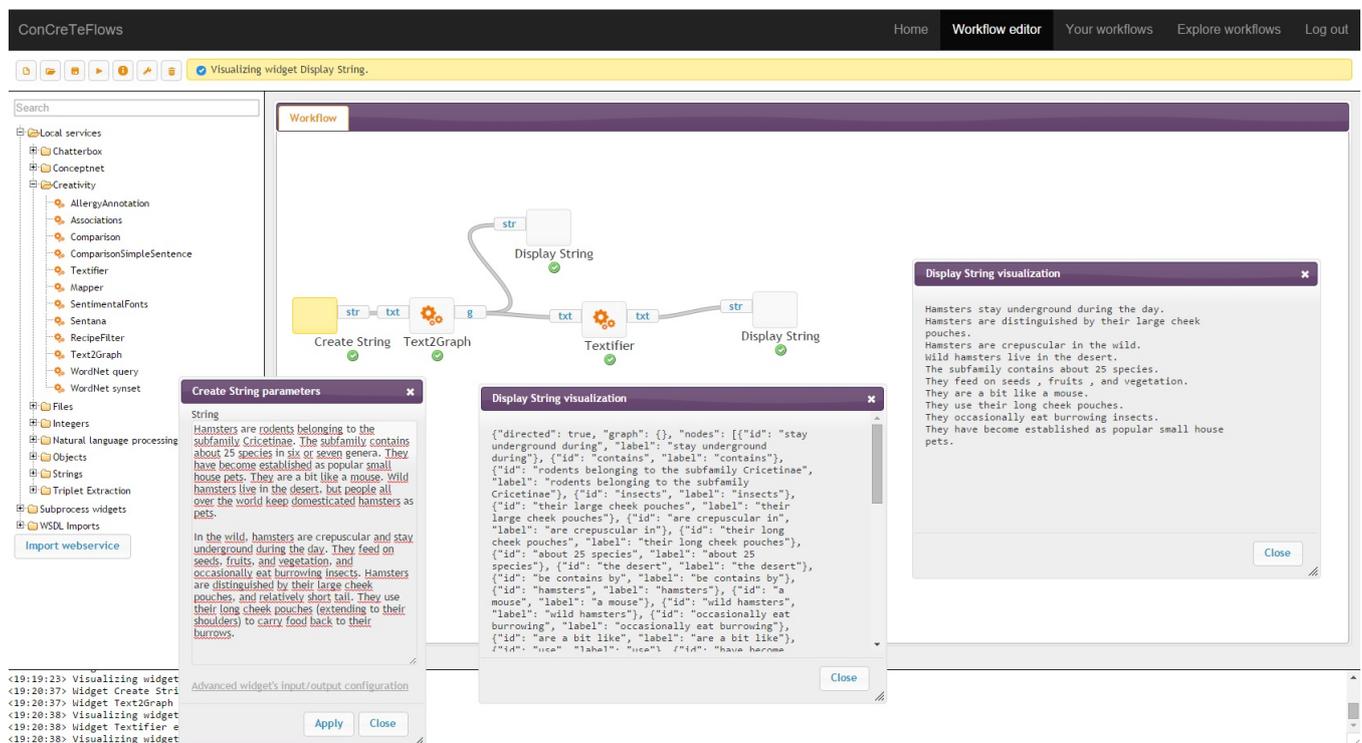


Figure 6: Example of Textifier working on input graph obtained from text.

Integration in a Workflow

The software components that implement the functionalities sketched in Figure 4 were implemented and integrated in ConCreTeFlows either as internal (Python) functions, wrapped standalone programs or as Web services. In addition, we implemented some additional components that support the user interaction and data processing. These are: (I) components for Web page content retrieval and filtering and (II) components for graph reformatting and visualization.

By connecting these software components, we composed a ConCreTeFlows workflow that conducts a basic conceptual, textual and visual concept blending. The workflow is presented in Figure 7 and is publicly available from: <http://concretflows.ijs.si/workflow/137/> where it can be executed, changed and appended with additional functionality.

In this workflow, two textual inputs are transformed into conceptual graphs by a series of the Download web page, Boilerplate removal and Text2Graph widgets. The first one

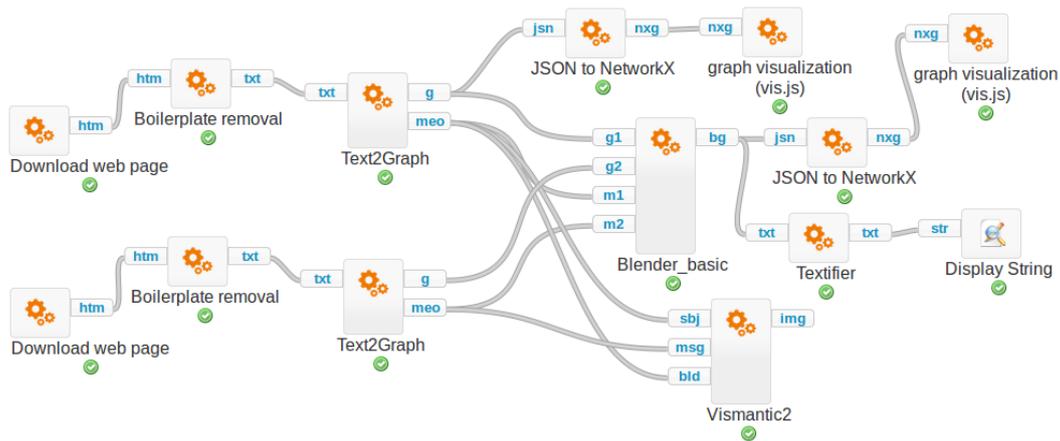


Figure 7: Workflow implementation in ConCreTeFlows (available at: <http://concreteflows.ijs.si/workflow/137/>).

obtains the Web page source from a given URL. In the example presented in this paper, these are the Wikipedia pages for two animals: hamster and zebra. The second widget removes the headers, menus, navigation and similar non-relevant content from the source. Finally, Text2Graph transforms the textual content into conceptual graphs (output *g*), which are available to other widgets with separately provided main entity (output *meo*). In the workflow, one of the graphs is reformatted and visualized with the graph visualization widget. All outputs of Text2Graph widgets enter the Blender_basic which blends the two graphs together and outputs a combined blended graph (output *bg*). This one gets served to the Textifier widget, which produces a textual description of the blend. Its output is presented by a standard Display String widget. The two main entities from Text2Graph widgets enter also the Vismantic2, which either changes the texture of one to the texture of the other (see Fig. 5a), or puts one in the usual surroundings of the other (Fig. 5b). This way it creates four candidates for visual blends. This widget takes somewhat longer to run, as it is in fact a call to a computationally intensive Web service. Upon completion, the outcome is shown in an output similar to the ones shown in Figure 5.

Workflow dissemination, reuse and extension

Any ConCreTeFlows workflow can either remain private or be made public for the purposes of dissemination and reproducibility of work. The workflow from the previous subsection is available from a public URL. This means that anyone can open it in ConCreTeFlows. Everytime this happens, a dedicated copy of the original workflow is made for that particular user. This allows any user not only to run the workflow with its original data and parameters, but also to change the inputs, parameters and redesign the structure of software components without affecting the original workflow.

Changing and extending a workflow is easy, but it requires some insight on the format of data that is exchanged among the widgets. This is usually made available in widget documentation, but can also be seen by observing the raw results of a widget (right-click and *Results*).

In the following we describe a simple exemplary extension of our workflow from Figure 7.

Exemplary addition: PoeTryMe widget The system named PoeTryMe (Gonçalo Oliveira and Cardoso 2015) is a poetry generation platform with a modular architecture that may be used to produce poetry in a given form, based on a set of seed words. Semantically-coherent lines are generated using the seeds or related words, and are produced by exploiting the knowledge in a semantic network and a grammar with textual renderings of the covered relations. A generation strategy selects some of the produced lines and organises them to suit the given form.

The PoeTryMe widget is limited to some of the features of the full system. Nevertheless, it can produce poetry in three languages (Portuguese, Spanish and English), given one of the available target forms (block of four, sonnet, ...), an open set of seeds, and a surprise factor, between 0 and 1, with implications on the selection of more or less semantically-distant words.

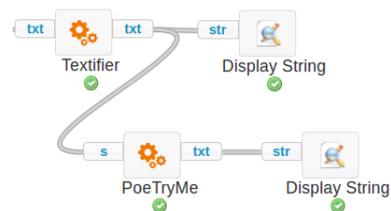


Figure 8: Addition of the PoeTryMe widget to the workflow.

In our workflow, the PoeTryMe widget can be appended to the Textifier widget (as shown in Figure 8) in order to get also a poem inspired by the resulting blend. Here is an example of a poem from a blend of *hamster* and *zebra*:

*when the coat paints the water white and black
stadiums here make song each stand has his
have not yet grown by the familiar crack
will mine and leave where the great love is*

Discussion and Future Work on Components

In the following, we discuss some of the encountered issues and shortcomings of the components and processes that are presented in this paper, as well as present some ongoing work on their improvements and additions.

Graph representations and formats The use of graphs for representing knowledge presents advantages in as much as it is a simple format with significant expressive power. In this sense it acts as a useful communication format for the various components in the flows envisaged. However, it has certain disadvantages in the sense that the graphs as considered at present do not have a unique semantic interpretation. Some of the modules produce graphs where relations are represented as edges between nodes representing objects, and others rely on graphs that represent relations as nodes occurring in the path of the graph between nodes representing objects. Even when the same approach to knowledge representation in a graph is used, problems may arise depending on the type of string used to label the nodes. Examples of problematic cases are: inflected verb forms used as well as verbs in infinitive, nouns used in singular and/or plural form, complex actions of the form *stay_at_home...* At present the content determination stage of the Textifier module is carrying out complex transformations to handle these various inputs in a uniform fashion when it comes to the final rendering. This requires the development of different version of the content determination stage for receiving input from different modules. It would be beneficial to make progress towards a unified approach to graph representation to allow blending operations to be carried out fruitfully between outputs generated by different modules. However, a certain flexibility is desirable in these content determination modules, so that they can tolerate inputs not altogether conforming to expectations. This is largely due to the open nature of the ConCreTeFlows platform, which may see the addition of new modules that do not conform to any standards set on graph representation, but also because the results of conceptual blending operations may not always produce output conforming to standards, even when the inputs to the conceptual blending process do conform.

TextStorm Conceptual Maps TextStorm (Oliveira, Pereira, and Cardoso 2001) is an NLP tool based on a Definite Clause Grammar (DCG) to extract binary predicates¹² from a text file using syntactic and discourse knowledge, not needing any preview knowledge about the discussed domain. The resulting set of predicates constitute a Conceptual Map. This tool can be used as an alternative to the Text2Graph.

TextStorm receives text as initial base of the open information extraction. After applying Part-of-Speech tagging and querying WordNet (Miller 1995), it builds predicates that map relations between two concepts from parsing of sentences. Its goal is to extract from utterances like Cows,

¹²These predicates have the common Prolog form: `funcor(Argument 1, Argument 2)`.

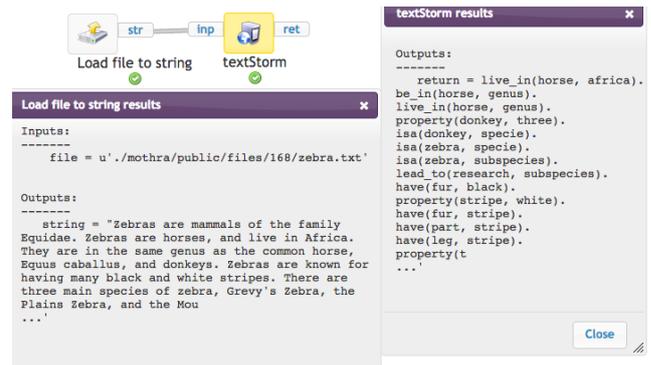


Figure 9: Imported Textstorm SOAP Web service used in ConCreTeFlows.

as well as rabbits, eat only vegetables, while humans eat also meat, the predicates `eat(cow, vegetables)`, `eat(rabbit, vegetables)`, `eat(human, vegetables)`, `eat(human, meat)` which will form its concept map. Since concepts in text are not named every time the same way, TextStorm uses WordNet's synonymy semantic relationship to identify the concepts that were already referred before with a different name.

Textstorm operates as a standards compliant SOAP Web service and as such can be imported on-the-fly to ConCreTeFlows (see Figure 9).

Divago concept blender The concept blending method that is currently used in the workflow from Figure 7 is very basic. We are currently working on the adaptation of a more elaborated blender, the pre-existing Divago (Pereira 2005), to offer its main functionalities as webservice in ConCreTeFlows. This blender adopts the same graph format as TextStorm, i.e., the Conceptual Map format, for the input and blended mental spaces.

The new blender, the *DivagoFlow*, is itself a flowchart composed of two modules, the *Mapper* and the *Blend Factory*. The first is responsible for finding analogy mappings between two Input Spaces using structural alignment. More precisely, it computes the largest isomorphic pair of sub-graphs contained in the Input Spaces. The output mapping is, for each pair of sub-graphs, the list of crossover relations between nodes of each of the input spaces. The *Blend Factory* takes these mappings as input, as well as the Input Spaces and a Generic Space. For each mapping, it performs a selective projection into the Blend Space, which leads to the construction of a *Blendoid*, an intermediate graph that subsumes the set of all possible blends. This Blendoid feeds an evolutionary process that explores the space of all possible combinations of projections of the Input Spaces taking into account the Generic Space. This module uses an implementation of the CB theory optimality principles (a set of principles that ensure a coherent and integrated blend) as fitness measure. When an adequate solution is found or a pre-defined number of iterations is attained, the Blend Factory stops the execution and returns the best Blend.

Conclusions

We have presented the ConCreTeFlows platform for online composition of computational creativity solutions. It is entirely Web based and does not require installation for its use. New processes in the platform can be designed as workflows of software components, which are either made available in the platform or even imported on-the-fly in case of SOAP Web services¹³. Workflows can be either private or shared, which makes for an elegant solution to dissemination and reuse of one's work and repeatability of experiments.

The main focus of the paper is on a use case, which shows how the platform can be used in practice and presents several computational creativity software components that were combined in a collaborative effort to implement an interesting conceptual blending solution. Namely, the resulting blends are not only conceptual but also visual and textual. The benefits of a unifying workflow for blending are twofold: a user can get blends of various kinds through the same user-interface and the components can affect one another to produce a more coherent and orchestrated set of multimodal blending results. While some of the presented components are currently being updated from implementing basic to more elaborate methods, the presented prototype solution is fully operational and serves as a proof of concept that such an approach to multimodal conceptual blending is possible. Potential for use of such an approach is for example in creation of news stories. Such a tool could form an entire automated article on a funny and humorous or a serious and thought-provoking blend of topics. All the components of an article are there: the text, the picture, as shown, one could even add a poem. Other potential uses of the approach could be in art, advertising and human creativity support.

To make these things possible, as described in section *Future Work on Components*, our future work will include improvement of the components and the workflow presented in this paper. We will also continue with development and improvement of the presented platform to make creation of this and other computational creativity solutions further more efficient, collaborative and fun.

Acknowledgments

This research was partly funded by the Slovene Research Agency and supported through EC funding for the project ConCreTe (grant number 611733) that acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission.

References

Bird, S. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, 69–72. Association for Computational Linguistics.

Charnley, J.; Colton, S.; and Llano, M. T. 2014. The FloWr Framework: Automated Flowchart Construction, Optimisation and Alteration for Creative Systems. In *Fifth Inter-*

¹³REST services can currently only be wrapped into native widgets.

national Conference on Computational Creativity (ICCC-2014), 315–323.

Fauconnier, G., and Turner, M. 2002. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Fauconnier, G. 1994. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. New York: Cambridge University Press.

Gonçalo Oliveira, H., and Cardoso, A. 2015. Poetry generation with PoeTryMe. In Besold, T. R.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity Research: Towards Creative Machines*, Atlantis Thinking Machines. Atlantis-Springer. chapter 12, 243–266.

Kranjc, J.; Podpečan, V.; and Lavrač, N. 2012. ClowdfloWS: A cloud based scientific workflow platform. In Flach, P. A.; Bie, T. D.; and Cristianini, N., eds., *ECML/PKDD (2)*, volume 7524 of *Lecture Notes in Computer Science*, 816–819. Springer.

Mausam; Schmitz, M.; Bart, R.; Soderland, S.; and Etzioni, O. 2012. Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Novak, J. 1998. *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. Mahwah, NJ: Erlbaum.

Oliveira, A.; Pereira, F. C.; and Cardoso, A. 2001. Automatic reading and learning from text. In *Proceedings of the International Symposium on Artificial Intelligence*, 69–72.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research* 12:2825–2830.

Pereira, F. C. 2005. *Creativity and AI: A Conceptual Blending approach*. Ph.D. Dissertation, Dept. Engenharia Informática da FCTUC, Universidade de Coimbra, Portugal.

Reiter, E., and Dale, R. 2000. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Schorlemmer, M.; Smaill, A.; Kühnberger, K.-U.; Kutz, O.; Colton, S.; Cambouropoulos, E.; and Pease, A. 2014. COINVENT: Towards a computational concept invention theory. In *Proceedings of the 5th Int. Conference on Computational Creativity, ICC-14, Ljubljana, Slovenia*, 288–296.

Thagard, P., and Stewart, T. C. 2010. The AHA! experience: Creativity through emergent binding in neural networks. *Cognitive Science* 35(1):1–33.

Veale, T., and O'Donoghue, D. 2000. Computation and blending. *Cognitive Linguistics* Special Issue on Conceptual Blending:253–282.

Xiao, P., and Linkola, S. 2015. Vismantic: Meaning-making with images. In *Proceedings of the Sixth International Conference on Computational Creativity, ICC-2015*, 158–165.

DANCE



Cochoreo: A Generative Feature in idanceForms for Creating Novel Keyframe Animation for Choreography

Kristin Carlson, Philippe Pasquier, Herbert H. Tsang, Jordon Phillips, Thecla Schiphorst and Tom Calvert

School of Interactive Arts and Technology, Simon Fraser University, Canada
Applied Research Lab, Trinity Western University, Canada
kca59, pasquier, htsang, jjp1, thecla, tom@sfu.ca

Abstract

Choreography is an embodied and complex creative process that often relies on ‘co-imagining’ as a strategy in generating new movement ideas. Technology has historically been used as a tool to augment creative opportunities in choreographic process, with multiple choreographic support tools designed to function as a ‘blank slate’ for choreography. However, few of these tools support creative authoring with interactive or generative components. Cochoreo is a sub-module for generating body positions as keyframes that catalyze creative movement, as part of the movement sketching tool idanceForms (idF). Cochoreo catalyzes movement sketching by using parameters from Laban Movement Analysis, an existing movement framework, to generate unique keyframes that are used as seed material for choreographic process. idF is a creativity support tool that engages with choreographers’ creative movement process by design. This paper presents the design of Cochoreo and evaluations from our pilot study with university dance students.

Introduction

Choreographers are artists who are always searching for new inspirations from which to design novel movement ideas. They derive inspiration by exploring movement physically on themselves, they view movement on others, they observe interactions between strangers, explore the physics of inanimate objects and manipulate existing technology to create new movement experiences. It is in these exploratory interactions that choreographers not only discover ideas but iterate them to develop larger pieces of creative movement material. The performance theorist Andre Lepecki developed the term ‘co-imagining’ for these specific kinds of interactions that require multiple participants to devise and develop ideas, but who are not necessarily co-authors in the composition process (Cunteanu 2016). This paper discusses the current state of choreographic support tools and how our system, titled Cochoreo, addresses existing gaps between the domains of creativity support tools and autonomously creative systems.

While there are a variety of digital systems designed to engage with choreographic process, few support inspiration of new movement ideas or the iterative process of developing movement material. Current tools fall on a spectrum of

possible choreographer interaction, with limited options for co-imagining systems. Creativity support tools aid a choreographer in their existing creative practice, yet do not offer new movement ideas to the choreographer. Autonomously creative systems generate novel movement options but do not have a way to iteratively interact with a live choreographer. Few co-imagining systems exist yet none support the choreographers personal exploration of novel movement.

To combine the functionality of a creativity support tool with an autonomously creative system we have designed Cochoreo to co-imagine novel movement with the choreographer. Cochoreo is a sub-module within the existing platform idanceForms, a sketching tool for movement based on creating and animating keyframes. Keyframes are single frames, taken from film terminology and used in animation to describe important start and stop points (see Figure 1). Cochoreo generates keyframes (as single frames of body positions) and interpolates between keyframes to animate choreographer-designed movement.

Cochoreo generates novel keyframes for body positions by using a fitness function designed and tested by a choreographer. The iterative design process by the choreographer ensured that generated body positions would be

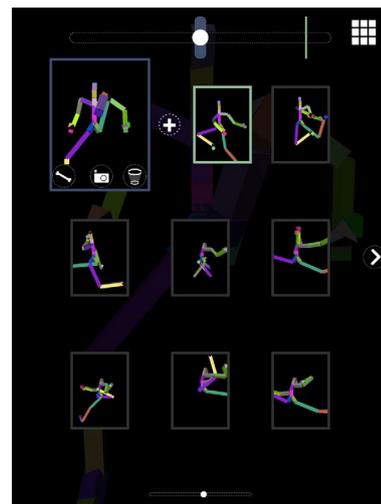


Figure 1: Keyframe Layout in idanceForms

unfamiliar, unstable and utilize complex movement understanding. There is also a parameterized fitness function option so that the choreographer can adjust generation options based on their personal preferences. The Cochoreo keyframes can then be edited and manipulated manually within the idanceForms framework.

Cochoreo was designed to leverage a co-imaginative approach to embodied choreographic process in technology. Cochoreo reflects the use of chance procedures made famous by world-renowned choreographer Merce Cunningham, who used the historical system DanceForms in his choreographic process. (Schiphorst et al. 1990). Our team has re-designed DanceForms (to idanceForms) to function on a mobile platform which utilizes affordances of a tablet for capturing and manipulating movement data.

This paper discusses the gap between creativity support tools and autonomously creative systems and illustrates an addressable gap in the design of choreographic support tools. We describe the system design of Cochoreo and present a pilot study with novice choreographers that explores their experience of choreography in relation to the integration of idanceForms, a platform for sketching movement and its generative feature Cochoreo.

Background

There are a variety of computational creativity projects that explore the generation or augmentation of movement material for choreography. Few of these systems are creative on their own and most involve some level of interaction with the human creator. However, these interactive systems often do not provoke creative compositional choices in the creator, and do not support the “sketching process”. For example, programs such as Adobe Photoshop or Microsoft Word give artists a “blank slate to put their ideas on but do not assist them artistically in their practice (Coughlan and Johnson 2009). We are interested in how an autonomous creativity component can support the creative process of the choreographer, to enable co-imaginative interaction. In order to implement techniques that engage the agency of choreographers, we illustrate a selection of existing systems that support choreographic process (see Figure 2). A deeper analysis of prior work includes the survey paper by (Fdili Alaoui, Carlson, and Schiphorst 2014) that described existing systems which have been developed to digitally reflect on movement material, to generate choreographic material, to provide real-time interaction with movement material, and to annotate movement material.

Systems that interactively support generative techniques and choreographic process include The Dancing Genome Project, Web3D Composer, Viewpoints AI and the standalone idanceForms.

The *Dancing Genome Project* developed a genetic programming model to explore sequences of movement in performance (Lapointe and poque 2005) (Lapointe 2005). The system analyses movement data and reorganizes it to create a new sequence with the same movements. This system was used to generate variations of movement phrases which were performed by a combination of live and digital performers. *Web3D Composer* creates sequences of ballet movements based on a predefined library

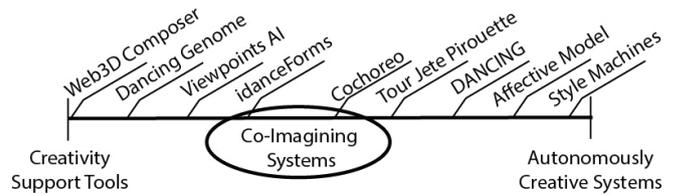


Figure 2: Co-Imagining Systems Scale

of movement material (Soga et al. 2006). The system allows the user to select movements from a pool of possibilities, which shift based on structural ballet syntax. This system is used mainly as a teaching tool to support the development of ballet structure knowledge. The Viewpoints AI project used the Viewpoints compositional framework to create a real-time interactive system exploring dance improvisation strategies (Jacob, M, Zook, A, and Magerko, B 2013). The system used kinect data and the SOAR reasoning framework to create a repository of short and long-term memory of the choreographers movements that select and apply different response modes and improvisational strategies. Both the system and the performer attend to each other’s movement choices by interactively improvising movement material. idanceForms enables choreographers to design movement poses as keyframes and then animates them, creating an iterative and reflective space for choreography design (Carlson et al. 2015a).

Systems that border on autonomous creativity include the Cochoreo, Tour, Jete, Pirouette, DANCING and Style Machine systems. Cochoreo generates body positions as keyframes within the idanceForms sketching application, to be used as catalysts for innovative movement design. Keyframes are animated and can be edited and sequenced within the idanceForms platform. Making use of the *DanceForms* framework, Yu and Johnsons system generates autonomous movement sequences through the use of a Swarm technique in their project titled *Tour, Jete, Pirouette* (Yu and Johnson 2003). This project used the existing libraries of movement within the DanceForms software to autonomously generate sequences from a series of individual movements onto a group of dance avatars. This created group movement sequences explored by choreographers who deemed the movement too challenging to perform exactly as the system did. *DANCING* used a series of music-related parameters, spatial pathway rules and a predefined library of traditional movements to generate Waltz choreography using a Genetic Algorithm (Nakazawa and Paezold-Ruehl 2009). By connecting the correct, predefined ‘steps’ in a domain-specific sequence that provides stage directions and orientations, this system generates syntactically correct movements in a complete choreography that are represented as ASCII (American Standard Code for Information Interchange) symbols on a birds eye view of the stage. It was noted that the generated choreography was able to be performed by ballroom dancers. Brand and Hertzmann developed a system called Style Machine that generates stylistic motion by using unsupervised learning techniques based

on a Stylistic Hidden Markov Model (SHMM) (Brand and Hertzmann 2000). This model learns patterns from a highly varied set of movement sequences recorded from motion capture data. The model then manipulates movement by identifying structure, style and accidental properties and applying style qualities to movement (such as modern dance style in ballet movements). Alemi, Li and Pasquier developed an interactive agent model that can capture and control the affective qualities of movement patterns (Alemi, Li, and Pasquier 2015). They trained a Factored, Conditional Restricted Boltzmann Machine (FCRBM) with a corpus of movement captured from two actors that was annotated based on their arousal and valence levels.

This selection of systems illustrates the developments towards interactively co-imagining choreographic material between a system and a human, yet there continues to be a gap.

DanceForms History

Using digital tools to support the creative process of choreography has a historical precedent. DanceForms (formerly Life Forms) is a human figure animation system that is optimized for dance (Calvert et al. 1991) including the same capabilities that are available in general purpose animation systems (e.g. Maya, MotionBuilder, 3D Studio Max, or Unity)(see Figure 3). While the system has been used by many choreographers, the most well known is Merce Cunningham. Cunningham used DanceForms to design movement as inspiration for constructing dances, exploring the random and procedural components of the system into his existing creative process using Chance Operations (a version of controlled randomization for content selection). Cunningham is a seminal figure in the history of choreography worldwide, and a unique 'user' of technology in dance, in particular the DanceForms software.

The Life Forms / DanceForms software was designed for use with desktop or laptop computers and normally requires a large screen (Calvert et al. 1993). Typically the user interaction requires that up to 5 windows be open at any time. The computer and the screen can be used in a studio but the computer is typically seen to be cumbersome and does

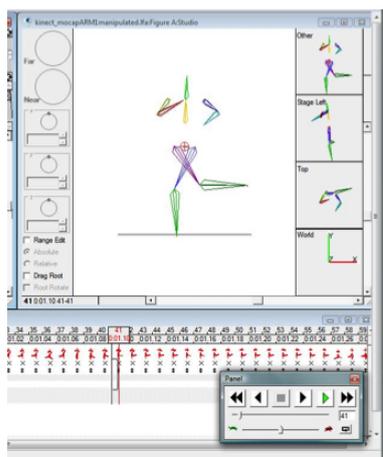


Figure 3: DanceForms Interface

not merge easily into a mobile, in-situ movement practice. The great advantage of mobile devices is just that: they are mobile. They can be carried onto the dance floor and the animated movements compare directly to the movements of the live dancers. There are also new affordances in mobile devices that we can take advantage of; the use of accelerometers to determine the acceleration, velocity and position of a limb and the use of an integral camera to capture the stance of a live dancer.

DanceForms has three views: space, time, and body-position. The space view allows the user to design movement pathways as spatial patterns. The timeline allows the choreographer to design sequences and timings of movement. The body-position view allows the user to design body positions using joint manipulation or to choose codified positions from pre-designed libraries. Libraries were designed by using the corpus of standard positions that define a movement language in techniques such as ballet or modern. These Danceforms libraries have been used as source material in generative composition using a Swarm algorithm to automatically compose sequences of movement (Yu and Johnson 2003). While DanceForms is the most articulate system available for computer-supported choreography, its precision-based design does not support rapid, portable, mobile, embodied or experiential forms of interaction. However, the rich foundation DanceForms provides for supporting movement design in software is highly useful as a step for mobile development and exploration of movement-based sensors for sketching choreography.

Cochoreo and the idanceForms Platform

Cochoreo is a sub-module of the idanceForms (idF) platform, a tablet-based mobile animation tool. This section will describe the idanceForms platform first, and the Cochoreo details second to illustrate the platform in which Cochoreo operates.

idF is a creativity support tool that allows the choreographer to sketch movement by creating, editing and viewing human figure animation on a tablet (Carlson et al. 2015b). idF differs from DanceForms in many ways: idanceForms is designed to support the sketching process of choreographers and is not meant to support the highly detailed traditional process in DanceForms. The shift in interaction from mouse-based to touch has dramatically changed the design to be more minimal but directed towards a choreographer working in an embodied way. This inspired the development of the Camera Keyframing feature, where a snapshot of a live dancer can be taken and used as a keyframe. idanceForms has been designed based on the epistemology of choreography; leveraging whole-body interaction as well as the playful and low-risk properties of sketching to create a mobile support tool for exploring creative movement in-situ (Blom 1982) (Studd and Cox 2013). By using an animation platform we can continue to provide an element of precision that the original DanceForms software maintains while opening to new opportunities for interaction, design and representation of movement. Our contribution with idF is its application to the live, in-situ creation and iteration of creative movement.

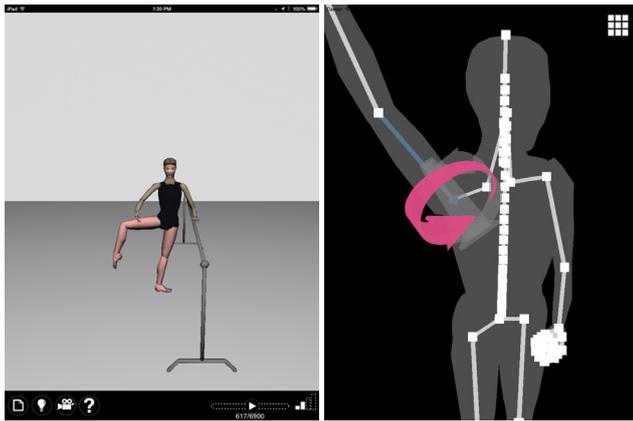


Figure 4: idanceForms Viewer and Skeleton Editing Tool

The 'home' screen is a playback screen that enables the choreographer to view the animation on a 'stage that they can move around using single finger touch to rotate around the space as well as pinch gestures to zoom (see Figure 4). The playback view is an important piece of the choreographic process, because it provides opportunities for viewing the animated movement, understanding the movement through the kinesthetically empathetic experience and reflection on the selection of and sequencing of still forms as keyframes. Playback is the portion of the creative process that provokes reflection and evaluation of choices made in the sketching process. Playback is the result of rapid prototyping: creating a space for choreographers to reflect in action and quickly continue working to create personal meaning.

Sequencing Keyframes

Once the choreographer has captured still poses to use as keyframes in their animation they have options for adjusting sequencing and timing of keyframes. Touching a keyframe once will select it and enable dragging and dropping to reorder keyframes for designing creative sequences. Because we are working with keyframes, there is built-in linear interpolation that takes the shortest path to move from one keyframe to the next. This creates a unique 'movement from the transition between a starting and ending still pose. The choreographer can control the timing of this 'movement by adjusting the timing into and out of a keyframe with the timing bar at the top of the editing screen.

Skeleton Editing Tool

Using the finger gesture the user can manipulate the skeleton in a joint and limbs level (see Figure 4). This fine control is facilitated by the gimbal ball visualization where the user can select the axis of the movement and then move the limbs accordingly.

Data Representation

The skeleton setup and skinning method we use is based on the COLLADA standard, using the CMU motion capture skeleton (cmu). We use a linked list of keyframes to

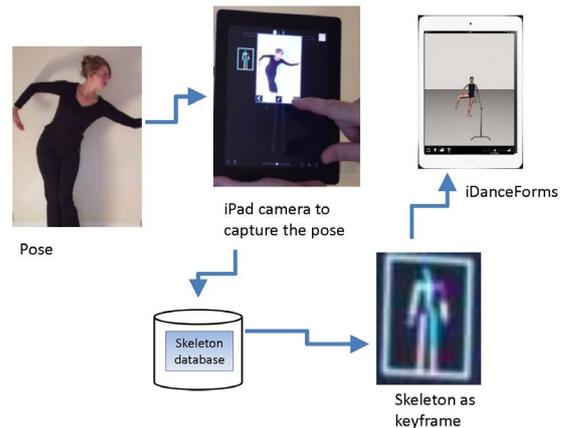


Figure 5: Camera Keyframing Feature

store our animations for two reasons a) this makes it easier to swap or move keyframes around during editing and b) it is also faster to play back the animation. Each keyframe stores a pose and an integer representing the number of in-between frames until the next keyframe. We do not store the explicit frame number in a keyframe, this is determined by the sum of the previous keyframes in the linked list added to the sum of inbetween frames for each keyframe, and this allows keyframes to be easily swapped / moved without recalculating their frame.

Camera Keyframing

idanceForms has developed a camera keyframing feature to enable embodied forms of interaction. Utilizing the 2D camera in mobile devices, the background is removed and the dancers still pose is compared to an existing database of images with existing skeletal data (see Figure 5). The built-in computer vision algorithm will then capture the pose and search through a database of pre-stored standard poses in order to try to find a corresponding skeleton pose. Once the skeleton pose has been found, it will be added to the list of keyframes. We have designed a database of movement using planar poses that can be easily detected from the front without occlusion. These poses include general and creative body positions as well as an imitation of alphabet letters that was used in a prior study with youth (Carlson et al., 2015). The existing skeletal data is used to create a keyframe that can be added to a sequence of keyframes to create an animation of movement and be further manipulated by the choreographer. This 'capture process is an exciting innovation for movement interaction which enables us to capture still forms as a wide range of potential planar figures.

Cochoreo System Design

Cochoreo is a sub-module of idanceForms, used to generate novel keyframes for creative movement (see Figure 7). Keyframes consist of a still body position, a human-shaped avatar with movement possibilities in all 3 axis. Cochoreo uses a Genetic Algorithm to evolve new keyframes from an

initial gene pool. Cochoreo is an extension of the Scuddle system (Carlson et al 2011), a generative system for movement catalysts. Scuddle generated movement catalysts as static positions with performative instructions to be interpreted by a choreographer and used as inspiration for designing new novel movement.

Cochoreo's implementation in the idanceForms animation platform enables a new parameterized fitness function, engaging the choreographer in the generative design process. idanceForms in return provides a sequencing and animation platform to iteratively view and design phrases of movement with the user, creating files that can be documented and used iteratively throughout the choreographic process. Currently Cochoreo operates using 2D data with the z-axis zeroed out. While we plan to move to 3D in the future, it will require another iterative design process to develop a constraint system for preferred creative catalysts.

Genetic Algorithm

We use a Genetic Algorithm to evolve movement catalysts. This approach enabled us to control fundamental components that problematize the choreographers process of creating movement, while generating novel inspirations for movement solutions. Genetic Algorithms are typically used to explore a wider range of potential solutions than other search algorithms can (Russell and Norvig 2010). We generate a population of 500 random individuals and give a score for their fitness against the prescribed goals for success. This initial population is then subjected to an iterative cycle of selection and breeding. Genes are bred using a two point cross over function with a 10 percent mutation percentage to create a new population that maintains diversity. Once a cycle is complete the new population is judged on its fitness once again and the process continues for a fixed number of five iterations or until a certain fitness threshold is reached (Floreano 2008) (Russell and Norvig 2010). More details on the generative process can be found in the Scuddle system paper (Carlson, Schiphorst, and Pasquier 2011).

Fitness Functions

Cochoreo has two fitness function options to evaluate novel body position criteria in keyframes. The options are: a pre-defined fitness function and a parametric fitness function based on Bartenieff Fundamentals movement constructs. The pre-defined fitness function uses a set of criteria specifically for provoking novel keyframes based on traditional dance movement. We have developed heuristic rules based on movement patterns discussed in Bartenieff Fundamentals and the authors expertise in contemporary dance practice to inhibit traditional habits when creating movement (Studd and Cox 2013). The fitness function evaluates each catalyst component separately (body symmetry, body position and levels) and then calculates the overall score. Preferred positions are those that highlight contralateral movement (body asymmetry), unstable levels with partially bent joints to create novel movement options. More on this function can be found in our prior paper (Carlson, Schiphorst, and Pasquier 2011). The parameterized fitness function allows the

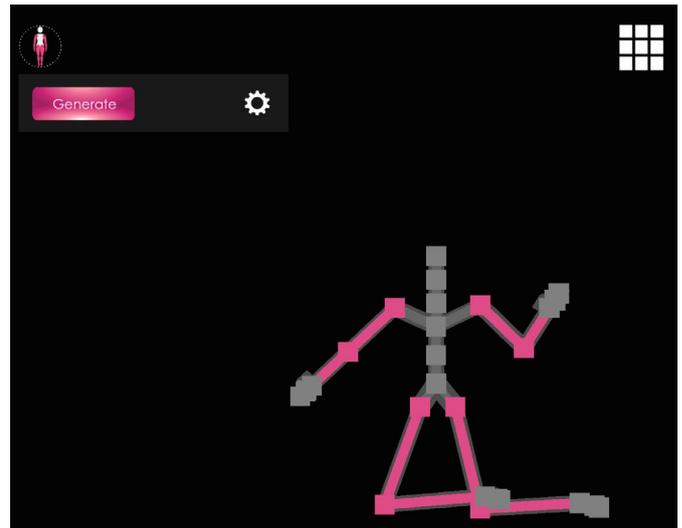


Figure 6: Pre-Defined Fitness Function Interface

choreographer to change the weighting of each parameter, creating more personalized generated options.

Cochoreo Interaction

To use the Cochoreo feature in idanceForms, the choreographer goes into the keyframe editing screen and selects a new keyframe. The Cochoreo screen is simple, providing a button for 'Generate' and a gear icon for access to the settings (see Figure 7). Every time the Generate button is pressed a new keyframe is created. This keyframe can then be re-edited in the skeleton editing view. Limbs can be isolated by selecting them, changing the color from pink to white. Isolated limbs will stay in place during the next generation cycle and can be un-selected by touching them again. If generation including a spinal configuration or spatial orientation is desired the user can manipulate these features first and then generate new keyframes in which the edits will be retained.

The default fitness function generates keyframes based on the pre-defined rules, developed through an iterative design process to specifically restrict habits from dance technique and provoke novel movement options. This default fitness function weights body asymmetry, uneven reach space and unstable levels more strongly to encourage novel movement exploration.

The parameterized fitness function enables the choreographer to select options based in the Bartenieff Fundamental parameters to weight the probability of that feature more or less strongly (see Figure 8) (Studd and Cox 2013). Parameters include: Body Half (symmetry on one side of the body), Upper Lower (symmetry on top or bottom half of the body), Cross Lateral (symmetry across the body with one arm and one leg), Near Reach Space (arms contracted), Far Reach Space (arms extended), Knee Extension/ Flexion (creating more or less stable levels).

Body positions can also be interacted with in the gener-

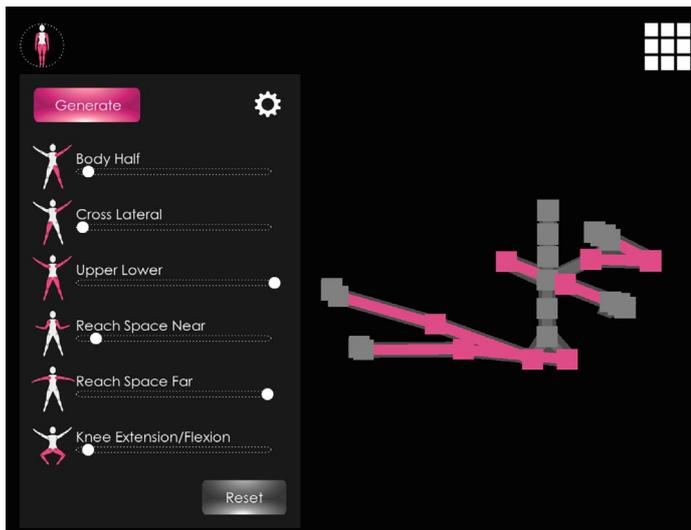


Figure 7: Parameterized Fitness Function Interface

ation by isolating limbs (see Figure 9). By isolating limbs they are removed from the algorithm while remaining limbs continue to be generated. The shape of the spine can also be manipulated by manually editing the vertebral joints using the skeleton editing features (selecting individual joints and moving them using the 3 axis) and then generating new limb positions.

Choreographic Study Exploring Creative Experience

We evaluated the system in a pilot study with 14 novice choreographers who were second year university dance major students. Choreographers met for two workshops over a week and had a composition assignment in-between

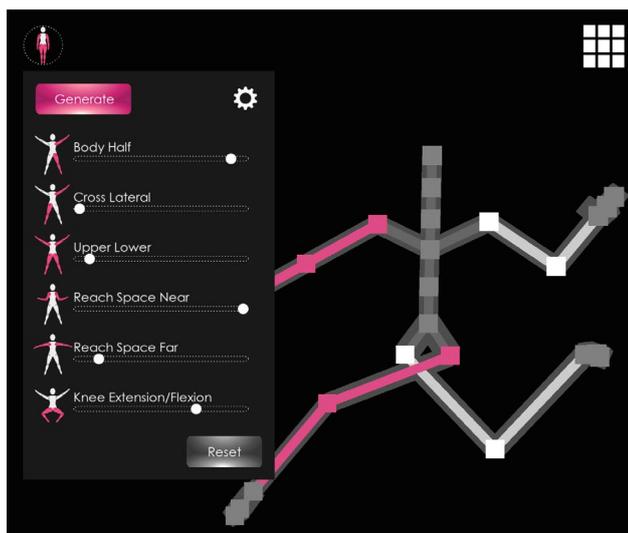


Figure 8: Isolating Limbs in Generation

workshops. The technology was introduced as a tool to support their existing choreographic process, and they were guided through short exercises to use it while constructing a movement phrase. Observational data was collected by the researcher through notes, photo and video documentation. Semi-structured focus groups were used to gather information about the creative experience. Data was analyzed using thematic analysis to highlight salient topics identified about choreographer's creative experience.

The goal of this research was to explore how choreographers can interactively develop creative movement with a system, where both user and system generate creative ideas and iteratively develop a movement phrase. Understanding how choreographers would work with the system required observing and understanding the embodied process of exploring and 'trying on' the movement on their particular bodies. When provided with the idanceForms app, choreographers explored the shape of the movement on their body. They then made mapping decisions about how to translate the data from the avatar to themselves, exploring it on the body and finding new connections where they could interactively augment the movement design themselves. In iteration with the system choreographers would insert new movements, and augment existing movements by either recapturing the movement into the device (and manipulating it there) or by viewing the movement 'cues from a different perspective.

Novelty of Generated Keyframes

Cochoreo's generated movement catalysts were viewed to be interesting, suggesting movement options that the choreographer would not have developed themselves. Paired with the manipulation tools of idanceForms, choreographers had a variety of options for controlling the generation of movement material. The camera keyframing feature enabled choreographers to capture positions with the iPad's camera, which was matching images to an existing database (and not always precise to the movement performed by the choreographer). However, when the data was less precise and more embodied it supported the choreographer's exploration of movement.

D: I liked working with the program because it pushed me to do movement I would never think of ... that was really interesting to me to try to put myself in this uncomfortable place and now a week later be comfortable in moving in that sort of way.

Resolution of Data/ Mapping Strategy

In the original design of Scuddle, generated positions were static and minimalistic stick figures. This design prompted choreographers to focus more on the interpretation of figures (and invention of new positions) than attempting to map the exact 2D position onto their moving 3D bodies. The low resolution of data catalyzed novel exploration, yet it could not guide positions into movement. Cochoreo generates keyframes, yet they are animated to create movements and use a 3D stick figure. This design uses a higher resolution of data, which prompts more attention to the physical mapping

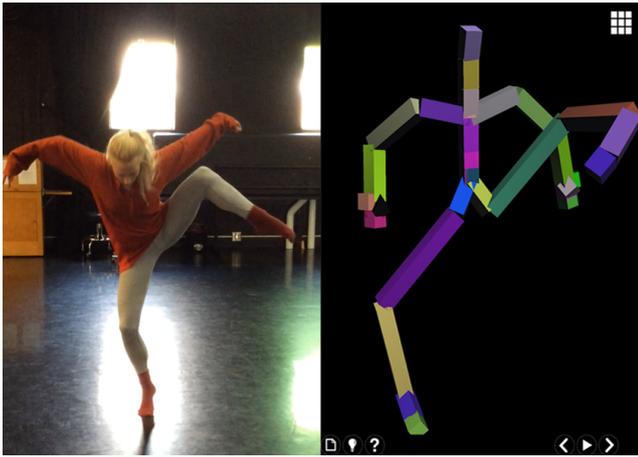


Figure 9: Pilot Study Session 2

from screen to body. When choreographers were learning material from the static keyframes, the higher resolution of data prompted them to focus less on their creative interpretation of the positions themselves, but brought attention to the transitions between positions as they had to maneuver dynamic changes in the data. Though when choreographers attended to the animated phrases instead of the static positions, they attended to the translation of dynamic parameters such as time and momentum more than the body position.

B: I felt that it felt better on my body if I used the app for inspiration for my movement and didn't necessarily try to replicate it exactly. So that really helped. And then when we sped up our movement or added repetition that helped it flow more easily through my body.

B: I felt that after going back and forth, like when I was first working with it it was very planal, but once we took it from there and took the movement home I could explore the other aspects of it. So then today in the space, even though we went back and added that bit in, I tried to keep the idea of my home movement but from a generated source.

Co-Imagining/ Interaction with System

Choreographers were asked to generate multiple keyframes in Cochoreo, then learn them on their own body to create a movement phrase. The goal was to eventually move smoothly back and forth between designing movement in the system and exploring the movement on the body. While the translation of data from device to body and back was a new challenge for many choreographers, they discovered unique perceptions to movement design through their interaction with the system. These included an attention to momentum as related to time in the system, the focus on angular limb positions and how they moved through time and attention to spatial orientation and engagement in relation to a focus on a mobile device. Choreographers also became aware of their movement habits (many which developed through movement training) and preferences when exploring move-

ment with specific sensory feedback (if a movement 'feels right' or 'looks right' in the mirror).

A2: It was interesting working with this movement away from an image (the software) because I feel like when you have this image in front of you, the mirror neurons you want to mimic this movement, and that's the way we learn movement, rather than learning from feelings, so not having that 'mirror' (of an image) and taking what this movement was in our memories and playing with the feeling of what it was, it was interesting and elicited new... it helped me evolve the movement. Having the exposure to it and then taking it away.

J: I usually focus more on momentum so it's interesting to approach with more emphasis on the angles, because it's like 'whoa I have limbs! I just realized I have limbs and it's in my face! Also realizing peripheral vision because there is a lot of stuff with angles happening back here which I don't usually think about.'

Conclusion

The evolution of Cochoreo as a sub-module within the *idanceForms* framework enabled us to explore how a creativity support tool could also provoke creative choreographic choices. We observed how choreographers devise movement using embodied methods, and augmented that process by inserting Cochoreo phrases within the embodied methods.

We view *Cochoreo* as a preliminary exploration of generative authoring tools for movement to evaluate how the affordances of such a system can support the creative values of a choreographer. While the goal of this project is to create an interactive toolkit for choreography design where a workflow can move smoothly between the choreographer and the technology, this is a complicated process that does not have obvious solutions in the near future. We are interested in how to take small steps to work towards this goal. In this study we observed the playful discovery process that each choreographer experienced and began weaving into crafted movement sequences. We see potential for systems that utilize generative movement augmentation to create embodied and personalized qualities of work, as opposed to designing for known creative processes using traditional interaction methods.

Future work includes connecting the Camera Keyframing feature Cochoreo to use embodied methods in the genetic algorithm. The choreographer would then be able to contribute to the initial gene population with their own movement data and could create target fitness functions. We are also investigating options for implementing novelty search to generate new positions that are maximally different from what the choreographer designs in Cochoreo.

Additional Media

Links to view videos of movement phrases:

Demonstration Videos of Cochoreo Generative Feature in *idanceForms*:

Predefined Fitness Function: <https://goo.gl/JjqjUc>

Parameterized Fitness Function: <https://goo.gl/AIkWNG>
Demonstration Videos of Select Choreographers in Final
Choreo Study:

Participant m: <https://goo.gl/gfLcBV>

Participant b: <https://goo.gl/G3lsHb>

References

- Alemi, O.; Li, W.; and Pasquier, P. 2015. Affect-expressive movement generation with factored conditional restricted boltzmann machines. In *Affective Computing and Intelligent Interaction (ACII)*, 442–448.
- Blom, L. A. 1982. *The Intimate Act of Choreography*. Pittsburgh, Pa: University of Pittsburgh Press.
- Brand, M., and Hertzmann, A. 2000. Style machines. In *The 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, 183–192. NY, USA: ACM Press/Addison-Wesley Publishing Co.
- Calvert, T. W.; Welman, C.; Gaudet, S.; Schiphorst, T.; and Lee, C. 1991. Composition of multiple figure sequences for dance and animation. *The Visual Computer* 7(2):114–121.
- Calvert, T. W.; Bruderlin, A.; Mah, S.; Schiphorst, T.; and Welman, C. 1993. The evolution of an interface for choreographers. In *Proceedings of the INTERCHI '93 conference on Human factors in computing systems*, 115–122. Amsterdam: IOS Press.
- Carlson, K.; Schiphorst, T.; Cochrane, K.; Phillips, J.; Tsang, H. H.; and Calvert, T. 2015a. Moment by Moment: Creating Movement Sketches with Camera Stillframes. In *ACM Conference on Creativity and Cognition, C&C '15*, 131–140. Glasgow: ACM.
- Carlson, K.; Tsang, H. H.; Phillips, J.; Schiphorst, T.; and Calvert, T. 2015b. Sketching Movement: Designing Creativity Tools for In-situ, Whole-body Authorship. In *The 2nd International Workshop on Movement and Computing, MOCO '15*, 68–75. Vancouver: ACM.
- Carlson, K.; Schiphorst, T.; and Pasquier, P. 2011. Scuddle: Generating Movement Catalysts for Computer-Aided Choreography. Mexico City, Mexico: ACM Press.
- Carnegie Mellon University - CMU Graphics Lab - motion capture library <http://mocap.cs.cmu.edu/>.
- Coughlan, T., and Johnson, P. 2009. Understanding Productive, Structural and Longitudinal Interactions in the Design of Tools for Creative Activities. In *The Seventh ACM Conference on Creativity and Cognition, C&C '09*, 155–164. NY, USA: ACM.
- Cunteanu, L. 2016. The Power of Co- in Contemporary Dance. *Revista-ARTA*.
- Fdili Alaoui, S.; Carlson, K.; and Schiphorst, T. 2014. Choreography As Mediated Through Compositional Tools for Movement: Constructing A Historical Perspective. In *The 2014 International Workshop on Movement and Computing, MOCO '14*, 1:1–1:6. Paris: ACM.
- Floreano, D. 2008. *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. Intelligent robotics and autonomous agents. Cambridge, Mass: MIT Press.
- Jacob, M.; Zook, A.; and Magerko, B. 2013. Viewpoints AI: Procedurally Representing and Reasoning about Gestures. In *Digital Games Research Association DIGRA 2013*.
- Lapointe, F.-J., and poque, M. 2005. The dancing genome project: generation of a human-computer choreography using a genetic algorithm. In *The ACM international conference on Multimedia*, 555–558. Singapore: ACM.
- Lapointe, F.-J. 2005. Choreogenetics: the generation of choreographic variants through genetic mutations and selection. In *The 2005 workshops on Genetic and evolutionary computation*, 366–369. Washington, D.C.: ACM.
- Nakazawa, M., and Paezold-Ruehl, A. 2009. DANCING, Dance ANd Choreography: an Intelligent Nondeterministic Generator. In *The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations*, 30–34. Portland, Oregon: ACM.
- Russell, S. J., and Norvig, P. 2010. *Artificial Intelligence: A Modern Approach*. N.J: Prentice Hall, 3rd ed edition.
- Schiphorst, T.; Calvert, T.; Lee, C.; Welman, C.; and Gaudet, S. 1990. Tools for interaction with the creative process of composition. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '90*, 167–174. Seattle: ACM.
- Soga, A.; Umino, B.; Yasuda, T.; and Yokoi, S. 2006. Web3d dance composer: automatic composition of ballet sequences. In *ACM SIGGRAPH*, 5 pages. Boston: ACM.
- Studd, K., and Cox, L. L. 2013. *Everybody is a body*. Indianapolis, IN: Dog Ear Publishing.
- Yu, T., and Johnson, P. 2003. Tour Jet, Pirouette: Dance Choreographing by Computers. In *Genetic and Evolutionary Computation GECCO 2003*. 201–209.

ROBODANZA: Live Performances of a Creative Dancing Humanoid

I. Infantino, A. Augello, A. Manfré, G. Pilato, F. Vella

Institute of High Performance Computing and Networking (ICAR)

National Research Council (CNR)

Palermo, Italy

surname@pa.icar.cnr.it

Abstract

The paper describes the artistic performances obtained with a creative system based on a cognitive architecture. The performances are executed by a humanoid robot whose creative behaviour is strongly influenced both by the interaction with human dancers and by internal and external evaluation mechanisms. The complexity of such a task requires the development of robust and fast algorithms in order to effectively perceive and process musical inputs, and the generation of coherent movements in order to realize an amusing and original choreography. A basic sketch of the choreography has been conceived and set-up in cooperation with professional dancers. The sketch takes into account both robot capabilities and limitations. Three live performances are discussed in detail, reporting their impact on the audience, the environmental conditions, and the adopted solutions to satisfy safety requirements, and achieve aesthetic pleasantness.

Introduction

Experience teaches that showing experimental prototypes outside the controlled environment of a laboratory is always a challenge, but often efforts are rewarded by the enthusiasm of the spectators and their appreciation. Science mixed with entertainment can transmit a direct and effective idea to a wide audience about the potentiality of a new technology. Recent models and systems proposed in the field of computational creativity (Boden 2009; Colton, Pease, and Charnley 2011), can find in this kind of presentation a perfect testbed and can benefit from the final judgement of an audience (Romero et al. 2012).

Recent scientific literature shows many examples of artificial systems operating in various artistic domains such as drama (Katevas, Healey, and Harris 2014) (Ogawa, Taura, and Ishiguro 2012) (Knight 2011), music (Wiggins et al. 2009) painting (Colton 2012), and dance (Nahrstedt et al. 2007) (Kar, Konar, and Chakraborty 2015) (Manfrè et al. 2016).

Robots having the human appearance (i.e. a humanoid) are an intriguing mean to convey the results of computational creativity models, especially for what concerns the dance domain because of their physicality, their sophisticated perception of the real world, their autonomous behaviour and their social interaction skills.

Dance presents major challenges for computational creativity and cognitive robotics, mainly because of the many factors to be considered: the perception of music (Seo et al. 2013), the execution of body movements, the execution of a sequence of movements in space, and the interaction with other dancers.

However various robotic dancing performances simply reproduce preprogrammed choreographies; robots are hence used just as a technological tool to support art and the real creative process is delegated mostly to the programmer or to the designer of the system, limiting both the decision-making autonomy and the capability of exploration of new solutions of the system.

Some works deal with more complex problems, with the aim to automatize as much as possible the robot behaviour. A basilar problem is the synchronization of music and movements in order to allow the robots to autonomously dance with real-time music introducing real-time beat extraction systems as in (Seo et al. 2013) and also the extraction of music emotions as in (Xia et al. 2012). In particular in (Xia et al. 2012) a dance is planned according to the beat end the emotions resulting from a preprocessing phase, then, a real-time synchronizing algorithm is used to minimize the error between the plan and the execution of the movements.

Other researchers focus the attention on the learning process of human movements styles (LaViers, Teague, and Egerstedt 2014); the authors investigate also the impression of an audience with regard to the consistency of the movements of the robot. In (Aucoeurier, Ogai, and Ikegami 2007) a robot provided with a biologically-inspired model, simulates the dynamic alternations between synchronisation and autonomy typically observed in human behaviour. In (Michalowski, Sabanovic, and Kozima 2007) the dance is proposed as a form of social interaction; according to the authors, the synchronization of the robot's movements with the music determined a greater involvement of children with a robot.

Despite such interest in this applicative field, at the best of our knowledge, there are no works introducing in the robot also creativity mechanisms. In our opinion, a realistic dancing behaviour does not involve only the perception of beat and emotions, and the choice of the most suitable dancing style. A realistic dancing behaviour includes also the capability of being creative, i.e. to create or improvise

new dancing movements. But such a creative behaviour involves different cognitive processes. It requires a motivation in creating something of new, the ability to get inspiration from the perceptions, properly represent them and comparing them with previous experiences, to assess the outcome of the creative process considering also external judgements. For this reason, we propose to model computational creativity and co-creation tools within proper cognitive frameworks. Cognitive architectures (CA) are inspired by functional mechanisms of the human brain and the various models proposed in literature (Goertzel et al. 2010) try to define the necessary modules to emulate the complex interactions among perception, memory, learning, planning, and action execution. These modules influence the external behaviors of agents and their interactions with humans.

In recent works, we have introduced a cognitive architecture supporting creativity (Augello et al. 2015; 2016), involving motivations and emotions (Augello et al. 2013). In particular, we explored the features of the Psi model (Bartl and Dörner 1998)(Bach, Dörner, and Vuine 2006) and its architecture, since it explicitly involves the concepts of emotion and motivation in cognitive processes, which are two important factors in creativity processes. We successfully employed a robot equipped with our creative system in two different artistic domains: *digital paintings*, and *dance creation* (Manfrè et al. 2016; Augello et al. 2016).

In this work we describe the architecture of the dancing robot, discussing its use in three live performances and the consequent impact on different kind of audiences. The behaviour of the robot is influenced by the interaction with human dancers, and by internal and external evaluation mechanisms. The live performances are discussed in detail, reporting their impact on the audience, the environmental conditions, and the adopted solutions to satisfy safety requirements, and achieve aesthetic pleasantness.

The paper has the following structure: the next section describes the architecture of the artificial system, and choreography fundamentals adapted to a dancing robot; then we describe in detail three live performances held in 2015 and the obtained results. Finally, we discuss in the last section what we have learned from these experiences, and the developments they may have in future.

The Artificial Creative System

The performances described in this paper are the result of the exploitation of a cognitive architecture developed in the past years and experimented by using an Aldebaran NAO humanoid platform (Gaglio et al. 2011; Augello et al. 2013; 2015).

The cognitive framework of the robotic dancer is depicted in Figure 1. The cognitive architecture is based on Psi model (Bartl and Dörner 1998), driven by *motivation* (Augello et al. 2014) (a numerical parameter), which is derived from *urges* (i.e. relevant demands of the artificial system): *Competence* and *Certainty* are directly influenced by evaluations (both internal and external) determining learning (Augello et al. 2015), affective state (Infantino 2012), behavior, and acting; *Affiliation* determines the social attitude of the robot; *physiological needs* are basic demands

(Infantino et al. 2013) such as *energy*, *correct functionality of body parts*, *motor temperatures*, and so on.

Working plans are stored in a *Long Term Memory* and they can be activated by motivation parameter and social interaction stimuli (see for example the interaction of the robotic painter with a user in (Augello et al. 2016)). In the dance domain, the plan is constituted by a set M of movements and rules to be associated with the perceived music stored as a transition matrix TM (Manfrè et al. 2016) of an HMM (Hidden Markov Model) subsystem.

The cognitive architecture supports artificial creativity through a simple interactive genetic algorithm used in the learning phase under the supervision of a teacher. The learning phase produces a set S of possible behaviours ranked by a score determined by an external evaluation expressed by the final audience.

The perception is based on simple audio features (beat intervals and loudness) extracted from musical input. The music is modeled by using k classes over a temporal sequence of N beats. A set of m elementary movements has been decided by professional dancers (Kirsch, Dawson, and Cross 2015): some of them are directly acquired by an RGBD (Red, Green, Blue, Depth) camera and translated into robot joints movements while other ones are created by the animation tool of NAO software. We have chosen a Hidden Markov Model approach, defined by two matrices: a transition matrix (TM) m by m allows the robot to choose the next movement after executing a given movement; an emission matrix (EM) m by k allows the robot to associate the perceived music to a given movement.

While the TM is designed by human dancers or a choreographer, the EM arises from an interactive genetic algorithm based on the human evaluation of the dance created. The n best EM s are therefore selected for each evolution step by using a fitness function based on the cosine distance from a given *master sequence* of movements (i.e. an example gave by the human dancer). During the learning phase, the dance obtained from the best EM is executed and it is evaluated by a human. If the dance triggers a positive reaction, the successful EM is saved and in the next evolution steps, it will be always considered as a parent of new individuals generated by crossover processes (preserving the unitary sum of probabilities). The evolution process ends when a given number of EM s that are selected and saved.

During the execution of the dance, the robot chooses an EM matrix from its *repertoire* S , and while perceiving the music it autonomously selects the best movement to execute. The set S could be viewed as a simplified version of a collection of styles (Ghedini, Pachet, and Roy 2016) activated by the HMM model. The robot switches between the two possible movements by counting their occurrences (for example a maximum 4 repetitions). Moreover, in order to introduce a variability of movements when the same musical sequence is repeated for a long time, we have decided to associate possible substitutes to each movement of the robot.

The training phase links given music and movements under expert supervision: the teacher indicates a reference sequence s^* of movements, and evaluates some selected sequences during genetic evolution. HMM model allows the

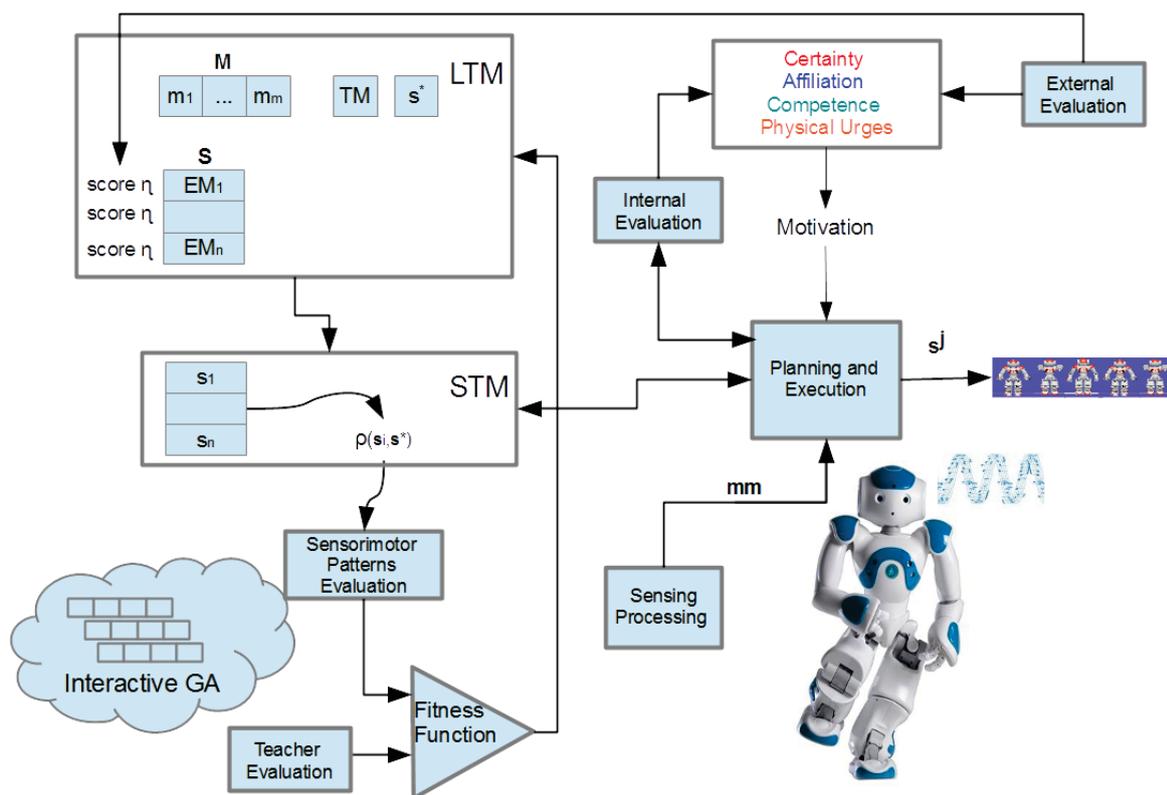


Figure 1: The Cognitive Framework of the Robotic Dancer.

system to react to any kind of music, exploiting a relation between detected musical features and an item in the movement set. The detected musical features are used to create the **mm** vector that represents the robot's model of music. The robot executes a new creative sequence of movements when listens to a new music taking into account its evaluated previous experience. Evaluation of robot behavior on new musical pieces is part of robot cognitive reaction by means of the Certainty and Confidence parameters.

Planning and Choreography Design

Thanks to the help of professional dancers, we conceived and set-up a performance under the following constraints: to physically execute admissible robot movements; to take advantage of robot peculiarities; to assure human safety; to perform the play in a real environment; to offer a performance which is aesthetically acceptable for the audience.

Conventional choreography design is based on the analysis and the planning of different aspects: composition of shape, space, timing, and dynamics. The shape is related to body posture and the dancer's figure. Real professional dancers train their bodies to be flexible and perform natural transitional movements. The shape could vary for body levels and parts involved, symmetry or asymmetry in positions

and sequences, the scale of execution.

Using an RGBD (Red, Green, Blue, Depth) camera we have recorded many dance movements of human dancers, converted them in robot movements by reproducing postures (Koenemann and Bennewitz 2012) and positions of body parts (hands, legs, head). Among the coded postures and positions we have selected those that have been judged as being aesthetically acceptable by professional dancers.

Another element of composition in choreography is the design in space, i.e. the paths and patterns that the dancer traces in the performance area. Complex dances consider geometrical paths, and as for the body shape, spatial patterns could be either symmetric or not, and they can be executed at different scales. We have chosen the simplest option that allows us to have some positive advantages: the robot is placed on a table and movement of legs are limited within a circumference of 60 centimeters of diameter. In this way we resolve the problem to have robot (57,3 cm tall) and human dancers at the same level; the robot is well visible also if the audience is a crowd; the surface of the table could be used as source of rhythmic sound; the robot movements are safer either for dancers and audience. The designed choreography is centered on this table and the robot standing on it, representing the spatial reference of the dancers.

The third element of dance composition is timing: taking into account tempo, metric, rhythm, and dynamics (Xia et al. 2012). The robot is capable of estimating the position of the source in the environment. By means of a software algorithm, beats can be detected and their features recorded (time interval, and loudness) in order to recognize patterns. During the dance performance, tempo and rhythm are given by dancers by using the table or their body.



Figure 2: Learning phase in the laboratory performing trials of a possible choreography with human dancers.

Live performances

The aim of the live performances has been to test our cognitive robotic technologies in a real environment under the evaluation of a real audience. The choreography conceived with dancers (see Figure 2) include both artistic and technical aspects: we stressed the environment sensing capabilities of the humanoid, robot's "naturalness" to execute sequences of movements, its artificial creative mechanisms to obtain an emotional impact. The whole performance had to be a logical artistic structure, including a prologue, a main part, and an epilogue. The choreography has tried to capture the spectator attention adding step by step a growing complexity both of the scene structure and of the interactions between humans and robot. Figure 3 shows the sequence diagram of the choreography. During the whole execution of the performance, the robot is completely autonomous and its behaviour is not controlled by anyone. Only the dancers in the scene can establish a coordination with the robot to synchronize the transitions among subsequent blocks of the choreography. We have chosen to use physical touches, exploiting

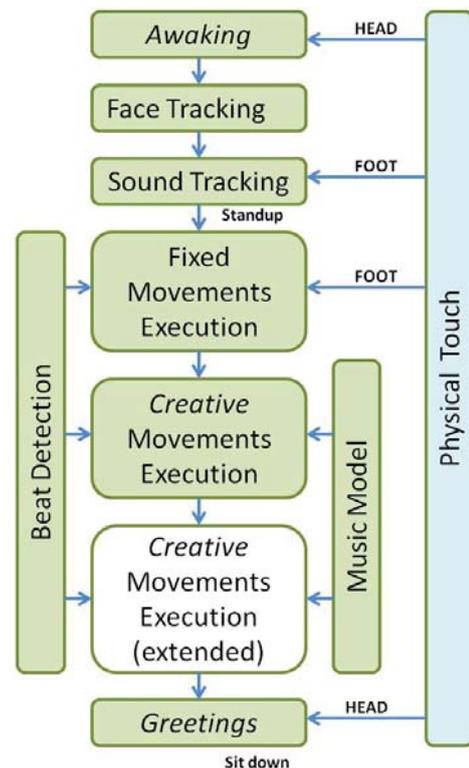


Figure 3: The choreography sequence diagram. The transitions between blocks are activated by dancers touching the robot. Music and beat detection modules drive movement executions of the humanoid. The white block represents an improvement of the performance introduced in third live event considering two more dancers and a musical piece.

some sensible areas of the robot (i.e. head, and bumpers on feet), assuring both robustness in the real complex environment and a strong emotional relationship between the human performer and the robot. The epilog shows three basilar capabilities of the robot: motion, face and sound tracking. The initial position of the robot is seated and with a crouched posture. The dancers start to go around the robot and observe with curiosity the strange "inanimate" object on the table. One of the dancers touches the robot's head in order to awake it, which starts to searching and tracking human faces. In this phase, the dancers play with the robot searching for gaining its attention, swapping their positions, and overlapping. A touch of a foot bumper causes the robot to localize rhythmic sounds. The dancers start to beat hands and to tap on the table. The robot turns its head in the direction of the perceived rhythms and opens and closes its hand following the perceived tempo. After another touch of a robot's foot, the performance reaches its main part, where dance and creativity capabilities of the robot are exhibited to the audience. The robot stands up, processes the audio input, tracking the rhythm generated by dancers and executing a sequence of three fixed movements. During this phase, the

robot is not creative, but it simply reproduces movements learned by the dancers. Automatically, after this phase, the robot exhibits its creative capabilities by trying to compose itself a sequence of movements. This task is accomplished by following the perceived model of the music and selecting a dance style from its repertoire (i.e. one individual of the privileged population selected during the learning phase based on the GA). The synchronization between movement and music is obtained by beat detection algorithm. The creative execution ends when a dancer touches one of the feet of the robot.

In the final part of the performance, the robot and the human dancers thank and greet the audience with a bow. After the two first live performances, following useful suggestions obtained by the audience, we have introduced a new creative phase that followed the first one: robot dances on a music and two more dancers have been introduced in the scene. The new very young performers have improvised their own dance both imitating robot postures, and professional dancers movements, and creating their own original movements.

The *cognitive architecture* has a great importance mainly because realizes a closed loop among learning, perception, execution, and final evaluation. In the loop humans strongly influence the robot behaviour by natural interactions, and avoiding a hard-coded programming of the robot. Artificial creativity allows the robot to explore new sequences of movements and to evaluate the effectiveness exploiting its internal evaluation (calculated by suitable distance metrics) and by expert judgments (by interactive GA). During the execution, the cognitive architecture drives the human-robot interaction by searching human faces (Affiliation needs), and by reacting to physical contacts between human and its body parts (head, feet), in order to establish a *feeling* with the human dancing partners. Also, physical urges could cause different behaviours among various performances: for example in the reported performances, the first two were done inhibiting the use of robot's left foot since the final joint was not working. Finally, external evaluations of the performances have determined the updating of the scores of S (depending by computation of Certainty and Competence): live performances have reinforced the element of its executed repertoire, but during the various rehearsals, dancers evaluations have caused the change of the rank of the S items.

All the videos of live performances and others testing sessions are accessible at the following link <http://www.pa.icar.cnr.it/scarlab/robodance/>.

Evaluation

After each performance, the spectators were asked to fill a questionnaire to evaluate the performance, expressing the following numerical ratings on a scale from 1 to 5:

- originality of the choreography
- naturalness of the robot-dancers interaction
- timing and movements of the robot
- evaluation of overall performance

Spectators were also asked if the performance had caused them some emotional impacts (possibly specifying which one), and if the robotic dance was perceived or not as being "mechanical". Table 1 reports the obtained results, which could be considered positive taking into account also that many of the spectators were experts of disciplines related to robotics. The spectators wrote on a free text space of the questionnaire, what they would have liked also the execution of the dance with a music accompaniment.

The first performance (see Figure 4) was held during a demo session of the AISC cognitive science conference (AISC midterm conference)¹. We obtained 30 evaluation forms by experts in various fields of Cognitive Sciences, and the results are reported in table 1. In all categories of judgment, the obtained values are above average, and the various comments provided on the questionnaires show that the audience was very attentive to the technical aspects, and considered artistic and creative aspects as secondary.



Figure 4: Live at AISC midterm event

The second live performance took place at Tavola Tonda Event on 28th June 2015, in the presence of a heterogeneous audience, among them many professional dancers, and musicians were presents(see Figures 5). We collected 30 evaluation forms, and the results are reported in table 2. In this case, unlike the first performance, the evaluations expressed a greater emphasis on the emotional and artistic aspects. The opinions were very positive, and the *Robodanza* performance achieved a great interest and curiosity. An important consideration with respect to the first indoor performance is the execution of the performance in an outdoor environment

¹<http://www.aisc-net.org/home/2015/03/05/aiscmidterm2015/>

characterized by high loudness and with a wider audience, but inserted in a real context of music and dance event (see Figure 5).



Figure 5: Live at Tavola Tonda Event. The second live has been performed in an external location, and in the presence of many people.

We also asked the dancers to express some qualitative evaluations of the played performances. Synthetically their judgment has been the following:

- the potentiality of artistic expression by choreographies involving robot are very high
- the interaction with the robot has been natural and funny and stimulates improvisation
- robot's movements have been considered not enough satisfying to perform more complex dance postures
- the positive reaction of the public has proved better than expected

The third live performance (see Figure 6) was held at the Conference on Biologically Inspired Cognitive Architectures (BICA) on 6th November 2015, in Lyon (France)². Following the suggestions and evaluations of audiences of previous live performances, we decide to add a new element in the choreography: the robot played a music (obtained in the learning phase when it recorded rhythmic beats and vocalizations) similar to previous creative phase (see Figure 3), and two children danced together with professional dancers

²<http://bicasociety.org/meetings/2015/>



Figure 6: Live at BICA 2015

and the robot. Also in this phase, the robot used its creative mechanisms while playing the musical piece. The children were free to execute their movements: they performed just two rehearsals the day before the live performance, and they autonomously conceived their dance behavior. In this case, we observed an interesting side effect: children imitated robot's movements and created a sequence similar to the robot one, synchronizing with it. The evaluation results are reported in Table 3.

Discussion

The three live performances have been completed asking the people attending the event to give a score on multiple aspects of the robodance and, to provide their ideas and opinions with open answers. It is straightforward to represent in a table the collected scores, while the opinions and open answers written in the form are not presented here. In any case, the ideas and the proposal have been taken into account and the evaluation of the public has been used to address learn-

Results	Means	Medians	%
Originality of choreography	3.07	3	61%
Naturalness of interaction	2.63	3	53%
Timing and Movements	2.87	3	57%
Overall judgement	2.73	3	55%
Perceived as dance (or mechanical)	.	.	50%
Emotions (yes/no)	.	.	60%

Table 1: Judgements expressed by audience of the first live performance at AISC 2015 midterm event.

Results	Means	Medians	%
Originality of choreography	4.07	4	81%
Naturalness of interaction	4.23	4	84%
Timing and Movements	3.87	4	77%
Overall judgement	4.20	4	84%
Perceived as dance (or mechanical)	.	.	80%
Emotions (yes/no)	.	.	86%

Table 2: Judgements expressed by audience of the second live performance at TavolaTonda event

Results	Means	Medians	%
Originality of choreography	3.58	4	72%
Naturalness of interaction	3.64	4	73%
Timing and Movements	3.55	4	71%
Overall judgement	4.03	4	81%
Perceived as dance (or mechanical)	.	.	69%
Emotions (yes/no)	.	.	63%

Table 3: Judgements expressed by audience of the third live performance at BICA 2015

ing and performing aspect of the robot dance through the update of learning capabilities and The audience of the three performances was bound the events where the demo took place. And since the events were oriented to people with different cultural background the form were filled by people from a variety of interest and all the three evaluation were good but being able to improve the cognitive system according the received feedback can be detected a positive trend in the evaluation of the creativity aspects.

The evaluations and feedbacks obtained by the audience in the three live performances led us to the considerations that are briefly summed up below.

The *overall judgment* considers the global evaluation that

tend to consider the engagement of the people to the exhibition. The fruition of an artefact should be straightforward and should not pass through the analysis of the piece of art that should be a second step. The value of *overall judgment* started from a value of 55% to reach a value of 81% in the last performance, showing an impressive improvement thanks to added components of the choreography, and reachings values comparables with the ones obtained with the generic audience of the second event.

The *Originality of the choreography* is bound to the richness of the dance and to the fact that the movements were repetitive or not. The score reaches its maximum during the Tavola Tonda Event where the audience was not bound to the robotic and technical world but it was more interested in the aspects of music and dance. The same trend has characterized the evaluation of the *Naturalness of Interaction* and the *Perception as dance* that were positively evaluated with the highest ratio by people interested in artistic aspects. The *emotions* raised by the performance were principally felt by the audience of the Tavola Tonda Event and less in the context bound to technical conference (the first and the third ones); nevertheless, the third performance had a higher ratio of positive results in the last event.

Conclusions and Future Work

We have tried to explain the complexity to realize a live dance performance involving humans and robot. Thanks to a cognitive architecture supporting artificial creativity, such aim appears to be feasible, and the obtained positive feedbacks have encouraged us to design further *live experiments*. At present, we are working on improving music perception capability of the robot. We are trying to stress the architecture in order to respond quickly to changes or interruptions of musical inputs, by synchronizing music and movements at different rhythms (fractions of the main tempo of the musical piece). Moreover, we are planning to introduce also a verbal interaction with the robot in order to stimulate a deeper empathy with the audience, and use artificial emotions to drive part of creative process. We are working on the improvement of the interactive genetic algorithm considering more suitable fitness functions. Besides, we are studying how to generate and evaluate new movements involving several robot body parts (e.g. symmetry, shape, and so on).

Acknowledgments

We thank Antonio Chella, and Salvatore Gaglio for their precious scientific collaboration; Barbara Crescimanno, Elisa D'Alessandro, and Veronica Racito (TavolaTonda) for the performance design and first two performances in Palermo; Rosanna Bova, Giampiero Rizzo, and Amélie Cordier for their technical support in BICA event; Chiara Castello, Giulia Demma, Daphnée Cornu-De Carvalho, and Sara Jouvin for their artistic contributions for BICA live performance.

References

Aucouturier, J.-J.; Ogai, Y.; and Ikegami, T. 2007. Making a robot dance to music using chaotic itinerancy in a network

- of fitzhugh-nagumo neurons. In *Neural information processing*, 647–656. Springer.
- Augello, A.; Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2013. Binding representational spaces of colors and emotions for creativity. *Biologically Inspired Cognitive Architectures* 5:64–71.
- Augello, A.; Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2014. Robotic creativity driven by motivation and semantic analysis. In *Semantic Computing (ICSC), 2014 IEEE International Conference on*, 285–289.
- Augello, A.; Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2015. Creativity evaluation in a cognitive architecture. *Biologically Inspired Cognitive Architectures* 11:29–37.
- Augello, A.; Infantino, I.; Lieto, A.; Pilato, G.; Rizzo, R.; and Vella, F. 2016. Artwork creation by a cognitive architecture integrating computational creativity and dual process approaches. *Biologically Inspired Cognitive Architectures* 15:74 – 86.
- Bach, J.; Dörner, D.; and Vuine, R. 2006. Psi and micropsi: a novel approach to modeling emotion and cognition in a cognitive architecture. In *Proceedings of the 7th international conference on cognitive modeling, Trieste*, 20–25.
- Bartl, C., and Dörner, D. 1998. Psi: A theory of the integration of cognition, emotion and motivation. In *Proceedings of the 2nd European Conference on Cognitive Modelling*, 66–73. DTIC Document.
- Boden, M. 2009. Computer Models of Creativity. *AI Magazine* 23–34.
- Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The face and idea descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*.
- Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and creativity*. Springer. 3–38.
- Gaglio, S.; Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2011. Vision and emotional flow in a cognitive architecture for human-machine interaction. In *BICA*, 112–117.
- Ghedini, F.; Pachet, F.; and Roy, P. 2016. Creating music and texts with flow machines. In *Multidisciplinary Contributions to the Science of Creative Thinking*. Springer. 325–343.
- Goertzel, B.; Lian, R.; Arel, I.; de Garis, H.; and Chen, S. 2010. A world survey of artificial brain projects, part ii: Biologically inspired cognitive architectures. *Neurocomputing* 74(1-3):30–49.
- Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2013. Humanoid introspection: A practical approach. *International Journal of Advanced Robotic Systems* 10.
- Infantino, I. 2012. *Affective human-humanoid interaction through cognitive architecture*. INTECH Open Access Publisher.
- Kar, R.; Konar, A.; and Chakraborty, A. 2015. Dance composition using microsoft kinect. In *Transactions on Computational Science XXV*. Springer. 20–34.
- Katevas, K.; Healey, P. G.; and Harris, M. T. 2014. Robot stand-up: engineering a comic performance. In *Proceedings of the Workshop on Humanoid Robots and Creativity at the IEEE-RAS International Conference on Humanoid Robots Humanoids (Madrid)*.
- Kirsch, L. P.; Dawson, K.; and Cross, E. S. 2015. Dance experience sculpts aesthetic perception and related brain circuits. *Annals of the New York Academy of Sciences* 1337(1):130–139.
- Knight, H. 2011. Eight lessons learned about non-verbal interactions through robot theater. In *Social robotics*. Springer. 42–51.
- Koenemann, J., and Bennewitz, M. 2012. Whole-body imitation of human motions with a nao humanoid. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, 425–425. IEEE.
- LaViers, A.; Teague, L.; and Egerstedt, M. 2014. Style-based robotic motion in contemporary dance performance. In *Controls and Art*. Springer. 205–229.
- Manfrè, A.; Infantino, I.; Vella, F.; and Gaglio, S. 2016. An automatic system for humanoid dance creation. *Biologically Inspired Cognitive Architectures* 15:1 – 9.
- Michalowski, M. P.; Sabanovic, S.; and Kozima, H. 2007. A dancing robot for rhythmic social interaction. In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, 89–96. IEEE.
- Nahrstedt, K.; Bajcsy, R.; Wymore, L.; Sheppard, R.; and Mezur, K. 2007. Computational model of human creativity in dance choreography. *Urbana* 51:61801.
- Ogawa, K.; Taura, K.; and Ishiguro, H. 2012. Possibilities of androids as poetry-reciting agent. In *RO-MAN, 2012 IEEE*, 565–570.
- Romero, J.; Machado, P.; Carballal, A.; and Correia, J. 2012. Computing aesthetics with image judgement systems. In McCormack, J., and dInverno, M., eds., *Computers and Creativity*. Springer Berlin Heidelberg. 295–322.
- Seo, J.-H.; Yang, J.-Y.; Kim, J.; and Kwon, D.-S. 2013. Autonomous humanoid robot dance generation system based on real-time music input. In *RO-MAN, 2013 IEEE*, 204–209. IEEE.
- Wiggins, G. A.; Pearce, M. T.; Müllensiefen, D.; et al. 2009. *Computational modeling of music cognition and musical creativity*. na.
- Xia, G.; Tay, J.; Dannenberg, R.; and Veloso, M. 2012. Autonomous robot dancing driven by beats and emotions of music. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 205–212. International Foundation for Autonomous Agents and Multiagent Systems.

Interactive Augmented Reality for Dance

Taylor Brockhoeft¹, Jennifer Petuch², James Bach¹, Emil Djerekarov¹, Margareta Ackerman¹, Gary Tyson¹

Computer Science Department¹ and School of Dance²

Florida State University

Tallahassee, FL 32306 USA

tjb12@my.fsu.edu, jap14@my.fsu.edu, bach@cs.fsu.edu, ed13h@my.fsu.edu, mackerman@fsu.edu, tyson@cs.fsu.edu

“Like the overlap in a Venn diagram, shared kinesthetic and intellectual constructs from the field of dance and the field of technology will reinforce and enhance one another, resulting in an ultimately deepened experience for both viewer and performer.” -Alyssa Schoeneman

Abstract

With the rise of the digital age, dancers and choreographers started looking for new ways to connect with younger audiences who were left disengaged from traditional dance productions. This led to the growing popularity of multimedia performances where digitally projected spaces appear to be influenced by dancers' movements. Unfortunately current approaches, such as reliance on pre-rendered videos, merely create the illusion of interaction with dancers, when in fact the dancers are actually closely synchronized with the multimedia display to create the illusion. This calls for unprecedented accuracy of movement and timing on the part of the dancers, which increases cost and rehearsal time, as well as greatly limits the dancers' creative expression.

We propose the first truly interactive solution for integrating digital spaces into dance performance: ViFlow. Our approach is simple, cost effective, and fully interactive in real-time, allowing the dancers to retain full freedom of movement and creative expression. In addition, our system eliminates reliance on a technical expert. A movement-based language enables choreographers to directly interact with ViFlow, empowering them to independently create fully interactive, live augmented reality productions.

Introduction

Digital technology continues to impact a variety of seemingly disparate fields from the sciences to the humanities and arts. This is true of dance performance as well, as interactive technology incorporated into choreographic works is a prime point of access for younger audiences.

Due in no small part to the overwhelming impact of technology on younger generations, the artistic preferences of today's youth differ radically from those raised without the prevalence of technology. This results in the decline of youth attending live dance performances (Tepper 2008). Randy Cohen, vice president for research and policy at Americans for the Arts, commented that: *“People are not*



Figure 1: An illustration of interactive augmented reality in a live dance performance using ViFlow. Captured during a recent performance, this image shows a dynamically generated visual effect of sand streams falling on the dancers. These streams of sand move in real-time to follow the location of the performers, allowing the dancers to maintain freedom of movement. The system offers many other dynamic effects through its gear-free motion capture system.

walking away from the arts so much, but walking away from the traditional delivery mechanisms. A lot of what we're seeing is people engaging in the arts differently.” (Cohen 2013). Given that younger viewers are less intrigued by traditional dance productions, dancers and choreographers are looking for ways to engage younger viewers without alienating their core audiences.

Through digital technology, dance thrives. Adding a multimedia component to a dance performance alleviates the need for supplementary explanations of the choreography. The inclusion of digital effects creates a more easily relatable experience for general audiences. Recently there has been an effort to integrate augmented reality into dance performance. The goal is to use projections that respond to the performers' movement. For example, a performer raising her arms may trigger a projected explosion on the screen behind her. Or, the dancers may be followed by downwards streams of sand as they move across the stage (see Figure 1). However, current approaches to augmented reality in professional dance merely create the illusion of interaction. Furthermore, only a few choreographers today have the technological collaboration necessary to incorporate projection effects in the theater space.

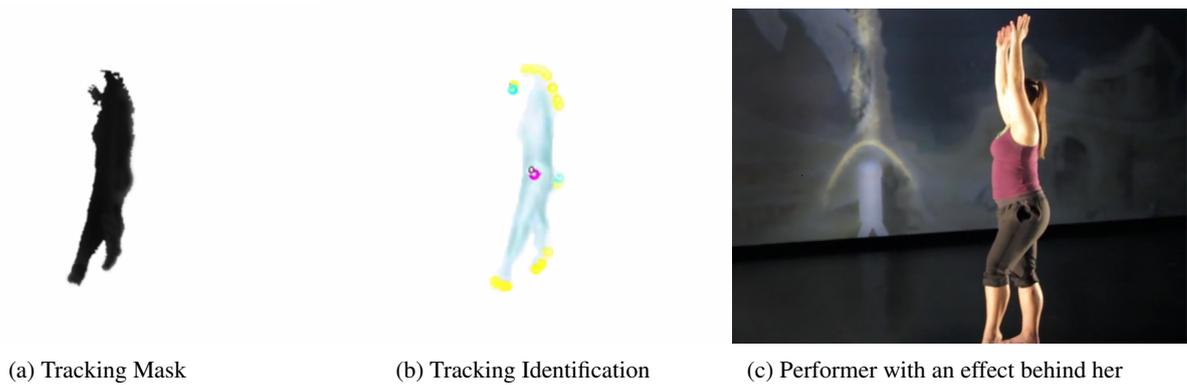


Figure 2: The ViFlow system in action. Figure (a) shows the raw silhouette generated from tracking the IR reflection of the performer, (b) displays the calculated points within the silhouette identified as the dancer core, hands, and feet, and (c) depicts the use of these points when applied to effects for interactive performance in the dynamically generated backdrop image.

Florida State University is fortunate to have an established collaboration between a top-ranked School of Dance and Department of Computer Science in an environment supportive of interdisciplinary creative activities. Where these collaborative efforts have occurred, we have seen a new artistic form flourish. However, the vast majority of dance programs and companies lack access to the financial resources and technical expertise necessary to explore this new creative space. We believe that this access problem can be solved through the development of a new generation of low-cost, interactive video analysis and projection tools capable of providing choreographers direct access to the video layering that they desire to augment their dance compositions.

Augmented dance performances that utilize pre-rendered video projected behind performers on stage to create the illusion of interactivity have several notable drawbacks. The dancers must rehearse extensively to stay in sync with the video. This results in an increase in production time and cost, and makes it impractical to alter choreographic choices. Further, this approach restricts the range of motion available to dancers as they must align with a precise location and timing. This not only sets limits on improvisation, but restricts the development of creative expression and movement invention of the dancer and choreographer. If a dancer even slightly misses a cue, the illusion is ineffective and distracting for the viewer.

A small number of dance companies (Wechsler, Weiß, and Dowling 2004) (Bardainne and Mondot 2015) have started to integrate dynamic visual effects through solutions such as touch-screen technology (see the following section for details.) However, moving away from static video into dynamically generated visualizations gives rise to a new set of challenges. Dynamic digital effects require a specialized skillset to setup and operate. The complex technical requirements of such systems often dictate that the visual content has to be produced by a separate team of technical developers in conjunction with performing artists. This requirement can lead to miscommunication as the language incorporated into the lexicon of dancers differs significantly from that em-

ployed by computer programmers and graphical designers. This disconnect can impair the overall quality of the performance as artists may ask for too much or too little from technical experts because they are unfamiliar with the inner workings of the technology and its capabilities.

In this paper we introduce ViFlow (short for Visual Flow¹), a new system that remedies these problems. Dancers, choreographers, and artists can use our system to create interactive augmented reality for live performances. In contrast with previous methods that provide the illusion of interactivity, ViFlow is truly interactive. With minimal low-cost hardware, just an infrared light emitter and an infrared sensitive webcam, we can track multiple users' motions on stage. The projected visual effects are then changed in real time in response to the dancers' movements (see Figure 2 for an illustration). Further, by requiring no physical gear, our approach places no restriction on movements, interaction among dancers, or costume choices. In addition, our system is highly configurable enabling it to be used in virtually any performance space.

With traditional systems, an artist's vision must be translated to the system through a technical consultant. To eliminate the need for a technical expert, we have created a gesture-based language that allows performers to specify visualization behavior through movement. Visual content is edited on the fly in a fashion similar to that of a dance rehearsal using our internal gesture based menu system and a simple movement-driven language. Using this movement-based language, an entire show's visual choreography can be composed solely by an artist on stage without the need of an outside technical consultant. This solution expands the artist's creative space by allowing the artist's vision to be directly interpreted by the system without a technical expert.

ViFlow was first presented live at Florida State University's Nancy Smith Fitcher Theatre on February 19, 2016 as part of *Days of Dance* performance series audi-

¹Flow is one of the main components of the dynamics of movement. In our system, it also refers to the smooth interaction between the dancer's movements and the visual effects.

tions. This collaborative piece with ViFlow was chosen to be shown in full production. Footage of the use of ViFlow by the performers of this piece can be found at <https://www.youtube.com/watch?v=9zH-JwlrRMO>.

Related Works

The dance industry has a rich history of utilizing multimedia to enhance performance. As new technology is developed, dancers have explored how to utilize it to enhance their artistic expression and movement invention. We will present a brief history of multimedia in dance performances, including previous systems for interactive performance, and discuss the application of interactive sets in related art forms. We will also present the most relevant prior work on the technology created for motion capture and discuss limitations of their application to live dance performance.

History of Interactive Sets in Dance

Many artists in the dance industry have experimented with the juxtaposition of dance and multimedia. As early as the 1950s, the American choreographer, Alwin Nikolais, was well known for his dance pieces that incorporated hand-painted slides projected onto the dancers bodies on stage. Over the past decade, more multimedia choreographers in the dance industry have been experimenting with projections, particularly interactive projection. Choreographers Middendorp, Magliano, and Hanabusa used video projection and very well trained dancers to provide an interplay between dancer and projection. Lack of true interaction is still detectable to the audience as precision of movement is difficult to sustain throughout complex pieces. This has the potential of turning the audience into judges focusing on the timing of a piece while missing some of the emotional impact developed through the choreography.

In the early 2000s, as technology was becoming more accessible, dance companies started collaborating with technical experts to produce interactive shows with computer generated imagery (CGI). Adrien M/Claire B used a physics particle simulation environment they developed called eMotion² that resulted in effects that looked more fluid. This was achieved by employing offstage puppeteers with tablet-like input devices that they used to trace the movements of performers on stage and thus determine the location of the projected visual effects (Bardainne and Mondot 2015). Synchronization is still required, though the burden is eased, because dancers are no longer required to maintain synchronized movement. This duty now falls to the puppeteer.

Eyecon (Wechsler, Weiß, and Dowling 2004) is an infrared tracking-based system utilized in Obarzanek's *Mortal Engine*. The projected effects create a convincing illusion of dancers appearing as bio-fiction creatures in an organic-like environment. However, Eyecon's solution does not provide the ability to differentiate and individually track each performer. As a result, all performers must share the same effect. The system does not provide the ability for separate dancers to have separate on-screen interactions. Moreover, Eyecon can only be applied in very limited performance

²eMotion System: <http://www.am-cb.net/emotion/>

spaces. The software forces dancers to be very close to the stage walls or floor. This is because the tracking mechanism determines a dancer's location by shining infrared light against a highly reflective surface, and then looking for dark spots or "shadows" created by the presence of the dancer. By contrast, we identify the reflections of infrared light directly from the dancers' bodies, which allows us to reliably detect each dancer anywhere on the stage without imposing a limit on location, stage size, or number of dancers.

Studies have also been conducted to examine the interactions of people with virtual forms or robots. One such study by (Jacob and Magerko 2015), presents the VAI (Viewpoint Artificial intelligence) installation which aims to explore how well a performer can build a collaborative relationship with a virtual partner. VAI allows performers to watch a virtual dance partner react to their own movements. VAI's virtual dancers move independently, however, VAI's movements are reactive to the movement of the human performer. This enhances the relationship between the dancer and the performer because VAI appears to act intelligently.

Another study by (Corness, Seo, and Carlson 2015), utilized the Sphero robot as a dance partner. In this study, the Sphero robot was remotely controlled by a person in another room. Although the performer was aware of this, they had no interaction with the controller apart from dancing with the Sphero. In this case, the performer does not only drive, but must also react to the independent choices made by the Sphero operator. Users reported feeling connected to the device, and often compared it to playing with a small child.

Interactivity in performance can even extend past the artist's control and be given to the audience. For LAIT (Laboratory for Audience Interactive Technologies) audience members are able to download an application to their phones that allows them to directly impact and interact with the show (Toenjes and Reimer 2015). Audience members can then collectively engage in the performance, changing certain visualizations or triggering cues. It can be used to allow an audience member to click on a button to signal recognition of a specific dance gesture or to use aggregate accelerometer data of the entire audience to drive a particle system projected on a screen behind the performers.

Interactive Sets in Other Art Forms

Multimedia effects and visualizations are also being used with increasing frequency in the music industry. A number of large international music festivals, such as A State of Trance and Global Gathering, have emerged over the last fifteen years that rely heavily on musically driven visual and interactive content to augment the overall experience for the audience. A recent multimedia stage production for musician Armin Van Buuren makes use of motion sensors attached on the arm of the artist to detect movements, which in turn trigger a variety of visual effects.³

The use of technology with dance performance is not limited to live productions. Often, artists will produce dance films to show their piece. As an example, the piece *Un-*

³Project by Stage Design firm 250K, Haute Technique, and Thalmic Labs Inc. <https://www.myo.com/arminvanbuuren>

named *Sound-Sculpture*, by Daniel Franke, used multiple Microsoft Kinect devices to perform a 3D scan of a dancer's movements (Franke 2012). Subsequently, the collected data was used to create a computer generated version of the performer that could be manipulated by the amplitude of the accompanying music.

Motion Capture Approaches (Tracking)

Many traditional motion capture systems use multiple cameras with markers on the tracked objects. Such systems are often used by Hollywood film studios and professional game studios. These systems are very expensive and require a high level of technical expertise to operate. Cameras are arranged in multiple places around a subject to capture movement in 3D space. Each camera must be set up and configured for each new performance space and requires markers on the body, which restrict movement and interaction among dancers. (Sharma et al. 2013)

Microsoft's Kinect is a popular tool that does not require markers and is used for interactive artwork displays, gesture control, and motion capture. The Kinect is a 3D depth sensing camera. User skeletal data and positioning is easily grabbed in real time. However Kinect only has a working area of about 8x10 feet, resulting in a limited performance space, thus rendering it impractical for professional productions on a traditional Proscenium stage, which is generally about 30x50 feet in size. (Shingade and Ghotkar 2014).

Organic motion capture⁴ is another marker-less system that provides 3D motion capture. It uses multiple cameras to capture motion, but requires that the background environment from all angles be easily distinguishable from the performer, so that the system can accurately isolate the moving shapes and build a skeleton. Additionally, the dancers are confined to a small, encapsulated performance space.

Several researchers (Lee and Nevatia 2009), (Peursum, Venkatesh, and West 2010), (Caillette, Galata, and Howard 2008) have built systems using commercial cameras that rely heavily on statistical methods and machine learning models to predict the location of a person's limbs during body movement. Due to the delay caused by such computations, these systems are too slow to react and cannot perform in real time (Shingade and Ghotkar 2014).

One of the most accurate forms of movement tracking is based on Inertial Measurement Units (IMUs) that measure orientation and acceleration of a given point in 3D space using electromagnetic sensors. Xsens⁵ and Synertial⁶ have pioneered the use of many IMUs for motion capture suits which are worn by performers and contain sensors along all major joints. The collected data from all sensors is used to construct an accurate digital three dimensional version of the performer's body. Due to their complexity, cost, and high number of bodily attached sensors, IMU systems are not considered a viable technology for live performance.

⁴Organic Motion - <http://www.organicmotion.com/>

⁵Xsens IMU system - www.xsens.com

⁶Synertial - <http://synertial.com/>

Setup and System Design

ViFlow has been designed specifically for live performance with minimal constraints on the performers. The system is also easy to configure for different spaces. The camera can receive information from a variety of different camera setups and is therefore conducive to placement in a wide spectrum of dance venues. By using Infrared(IR) light in the primary tracking system, it also enables conventional lighting setups ranging from very low light settings to fully illuminated outdoor venues.

Hardware and Physical Setup

ViFlow requires three hardware components: A camera modified to detect light in the infrared spectrum, infrared light emitters, and a computer running the ViFlow software. We utilize infrared light because it is invisible to the audience and results in a high contrast video feed that alleviates the process of isolating the performers from the rest of the environment, when compared to a regular RGB video feed. By flooding the performance space with infrared light, we can identify the location of each performer within the frame of the camera. At the same time, ViFlow does not process any of the light in the visible spectrum and thus is not influenced by stage lighting, digital effect projections, or colorful costumes.

Most video cameras have a filter over the image sensor that blocks infrared light and prevents overexposure of the sensor in traditional applications. For ViFlow, this filter is replaced with the magnetic disk material found in old floppy diskettes. This effectively blocks all visible light while allowing infrared light to pass through.

In order to provide sufficient infrared light coverage for an entire stage, professional light projectors are used in conjunction with a series of filters. The exact setup consists of Roscolux⁷ gel filters - Yellow R15, Magenta R46, and Cyan R68 layered to make a natural light filter, in conjunction with an assortment of 750-1000 watt LED stage projectors. See Figure 3 for an illustration.

The projector lights are placed around the perimeter of the stage inside the wings (see Figure 4). At least two lights should be positioned in front of the stage to provide illumination to the center stage area. This prevents forms from being lost while tracking in the event that one dancer is blocking light coming from the wings of the stage.

The camera placement is arbitrary and can be placed anywhere to suit the needs of the performance. However, care must be taken to handle possible body occlusions (i.e. two dancers behind each other in the camera's line of sight) when multiple performers are on stage. To alleviate this problem, the camera can be placed high over the front of the stage angled downwards. (see Figure 4)

ViFlow Software

The software developed for this project is split into two components: the Tracking Software and the Rendering/Effect creation software. The tracking software includes data collection, analysis, and transmission of positional data to the

⁷Roscolux is a brand of professional lighting gels.

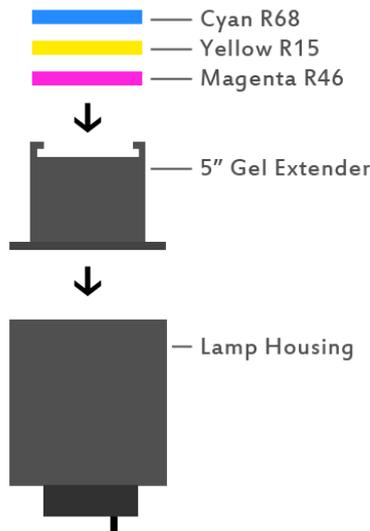


Figure 3: Gels may be placed in any order on the gel extender. We used LED lighting, which runs much cooler than traditional incandescent lighting.

front end program, where it displays the effects for a performance. ViFlow makes use of *OpenCV*, a popular open source computer vision framework. ViFlow must be calibrated to the lighting for each stage setup. This profile can be saved and reused later. Once calibrated, ViFlow can get data on each performer's silhouette and movement.

At present, there are certain limitations in the tracking capabilities of ViFlow. Since a traditional 2D camera is used, there is only a limited amount of depth data that can be derived. Because of the angled setup of the camera, we do obtain some depth data through interpolation on the y axis, but it lacks the fine granularity for detecting depth in small movements. Fortunately, performances do not rely on very fine gesture precision, and dancers naturally seem to employ exaggerated, far-reached gestures designed to be clearly visible and distinguishable to larger audiences. In working with numerous dancers, we have found that this more theatrical movement seems to be instilled in them both on and off stage.

Visual Effects

The front end uses Unity3D by Unity Technologies⁸ for displaying the visual medium. Unity3D is a cross-platform game engine that connects the graphical aspects of developing a game to JavaScript or C# programming. Unity has customization tools to generate content and is extensible enough to support the tracker. The front end consists of five elements: a camera, a character model, an environment, visual effects, and an interactive menu using gesture control which is discussed in more detail in following sections.

The camera object correlates to what the end-user will see in the environment and the contents of the camera viewport

⁸Unity3D can be downloaded from <https://unity3d.com>

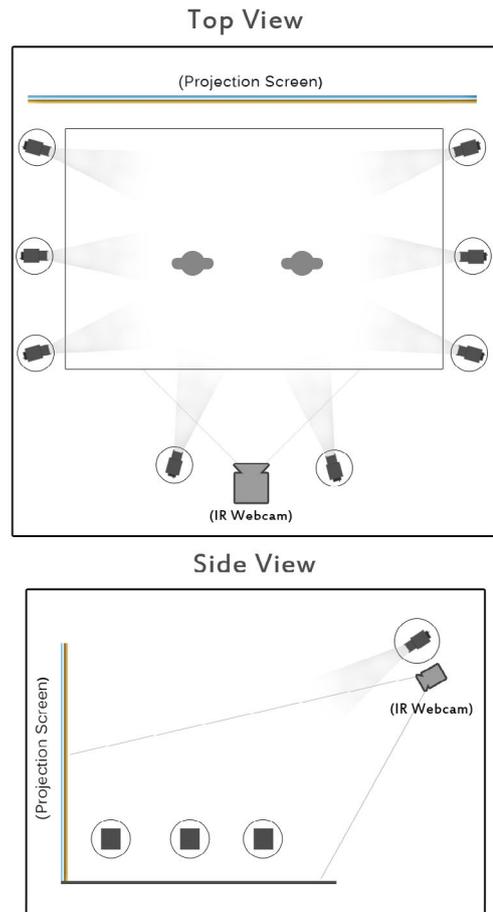


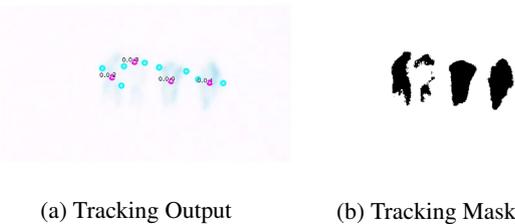
Figure 4: Positioning of the camera and lights in our installation at the Nancy Smith Fichter Dance Theatre at Florida State University's School of Dance. Lights are arranged to provide frontal, side, and back illumination. Depending on the size of the space, additional lights may be needed for full coverage. (Lights are circled in diagram.)

are projected onto the stage. The visual perspective is both 2D and 3D to support different styles of effects.

The character model belongs to a collection of objects representing each performer. Each object is a collection of two attached sphere colliders for hand representations and a body capsule collider as seen in Figure 6. The colliders are part of the Unity engine and are the point of interaction and triggers menus, environmental props, and interactive effects.

Environments consist of multiple objects including, walls, floors, and ceilings of various shapes and colors. Aesthetic considerations for these objects are applied per performance or scene such as Figure 7. Most of our environmental textures consist of creative usage of colors, abstract art, and free art textures.

The effects are delivered in a variety of methods such as interactive objects, particle systems, and timed effects. Some objects are a combination of other effects designed to



(a) Tracking Output

(b) Tracking Mask

Figure 5: Four figures being tracked with our tracking software. Each individual is bathed in infrared light, thus allowing us to easily segment their form from the background. This shot is from the camera angle depicted in Figure 4.

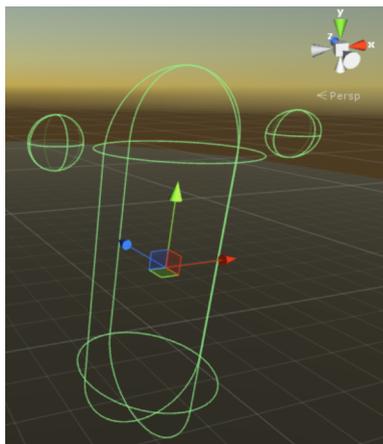


Figure 6: Character Model Object. The small orbs are the colliders for hand positions and the larger capsule is the body collider.

deliver a specific effect such as an interactive object that will trigger a particle system explosion upon interaction with a performer.

The particle system delivers ambience and interactive effects like rain, fog, waterfalls, fire, shiny rainbow flares, or explosions. ViFlow's effects provide a set of adjustable features such as color, intensity, or direction. The particle systems have been preconfigured as interactive effects such as a sand waterfall that splashes off the performers as seen in Figure 1 or a wildfire trail that follows the performers in Figure 8.

Some effects involve environmental objects that the dancer can interact with. One effect is a symmetric wall of orbs that cover the lower portion of the 2D viewport. When touched by the performer's Unity collider, these dots have preconfigured effects such as shrinking, floating up, or just spiraling away. The customizations supported for the performers allow them to place the effects in specific locations, change their colors, and adjust to predefined effects.

Lastly, there are global effects that can be both environmentally aesthetic, such as sand storms and snow falls, or interactive such as a large face that watches the dancer and responds based on their position. The face might smile when they are running and frown when they are not moving, or

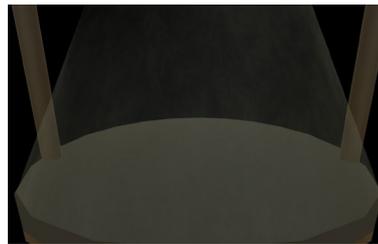


Figure 7: This static environment is the lower part of an hourglass, used in a performance whose theme centers on time manipulation. The dancers in this piece interact with a sand waterfall flowing out of the hourglass.



Figure 8: Two Unity particle systems, one used as an interactive fire effect and the other is a triggered explosion.

turn left and right as the dancers are moving stage left or right.

Communication Gap Between Dancers and Technologists

Multimedia productions in the realm of performing arts are traditionally complex due to the high degree of collaboration and synchronization that is required between artists on stage and the dedicated technical team behind the scenes. Working in conjunction with a technical group necessitates a significant time investment for synchronization of multimedia content and dance choreography. Moreover, there are a number of problems that arise due to the vastly different backgrounds of artists and technicians in relation to linguistic expression. In order to address these communication difficulties, we developed a system which allows artists to directly control and configure digital effects without the need for additional technical personnel by utilizing a series of dance movements which collectively form a gesture based movement language within ViFlow.

One of the main goals of our system is to enhance the expressive power of performing artists by blending two traditionally disjoint disciplines - dance choreography and computer vision. An important take away from this collaboration is the stark contrast and vast difference in the language, phrasing, and style of expression used by dancers and those with computing oriented backgrounds. The linguistic gap

between these two groups creates a variety of development challenges such as system requirements misinterpretations and difficulties in creating agreed upon visual content.

To better understand the disparity between different people's interpretations of various visual effects provided by our system, we asked several dancers and system developers to describe visual content in multimedia performances. The phrasing used to describe the effects and dancer interactions of the system were highly inconsistent, as well as a potential source of ambiguity and conflict during implementation.

Dancers and developers were separately shown a batch of video clips of dance performances that utilized pre-rendered visual effects. Each person was asked to describe the effect that was shown in the video. The goal was to see how the two different groups would describe the same artistic visual content, and moreover, to gain some insight into how well people with a non-artistic, technical background could interpret a visual effect description coming from an artist.

The collected responses exposed two major issues. First, the descriptions were inconsistent from person to person, and second, that there was a significant linguistic gap between artists and people with a computing background. As an example, consider this description of a visual effect written by a dancer: *"I see metallic needles, projected onto a dark surface behind a solo dancer. They begin subtly, as if only a reference, and as they intensify and grow in number we realize that they are the echoes of a moving body. They appear as breathing, rippling, paint strokes, reflecting motion"*. A different dancer describes the same effect as *"sunlight through palm fronds, becomes porcupine quills being ruffled by movement of dancer"*. A system developer on the other hand, described the same visual effect as *"a series of small line segments resembling a vector field, synchronized to dance movements"*. It is evident that the descriptions are drastically different.

This presents a major challenge as typically, a technician would have to translate artists descriptions into visual effects. Yet, the descriptions provided by dancers leave a lot of room for personal interpretation, and lead to difficulties for artists and technicians when they need to reach agreement on how a visualization should look like on screen. In order to address this critical linguistic problem, our system incorporates a dance derived, gesture-based, motion system that allows performers to parameterize effects directly by themselves while dancing, without having to go through a technician who would face interpretation difficulties. This allows dancers a new level of artistic freedom and independence, empowering them to fully incorporate interactive projections into their creative repertoire.

Front End User Interface and Gesture Control

Our interactive system strives to eliminate the need for a technician to serve as an interpreter, or middleman, between an artists original vision and the effects displayed during a performance. As discussed above, a number of linguistic problems make this traditional approach inefficient. We address this problem by implementing a direct dance-based gesture control, which is used for user interactions with the system as well as customizing effects for a performance.

The system has two primary modes of operation: a *show-time mode* which is used to run and display the computerized visual component of the choreographed performance during rehearsals or production, and an *edit mode* which is used to customize effects and build the sequence of events for a performance. In other words, edit mode is used to build and prepare the final show-time product.

Edit mode implements our novel gesture-based approach for direct artist control of computer visualizations. It utilizes a dancer's body language (using the camera input as previously described in the System Setup and Design Section) to control the appearance of digital content in ViFlow.

Effects are controlled and parameterized by the body language and movements of the dancer. A number of parameters are controlled through different gestures. For example, when configuring a wildfire trail effect, shown in Figure 8, the flame trail is controlled by the movement speed of a dancer on stage, while the size of the flame is controlled via hand gestures showing expansion as the arms of a dancer move away from each other. In a different scenario, in which a column of sand is shown as a waterfall behind a dancer, arm movements from left to right and up and down are used to control the speed of the sand waterfall, as well as the direction of the flow. Depending on the selected effect, different dance movements control different parameters. Since all effects are designed for specific dance routines, this effectively creates a dance derived movement-gesture language, which can be naturally and intuitively used by a dancer to create the exact visual effects desired.

When a dancer is satisfied with the visualization that has been created, it is saved and added to a queue of effects to be used later during the production. Each effect in the queue is supplied with a time at which it should be loaded. When a dancer is ready, this set of effects and timings are saved and can be used during the final performance in show-time mode.

Discussion: Creativity Across Domains

This interdisciplinary research project brought together two fields with different perspectives on what it means to be creative. In our joint work we learned to appreciate both the differences in how we approach the creative process and our goals for the final product.

From the perspective of dance and choreography, this project charts new territories. There is no precedent for allowing the choreographer this degree of freedom with interactive effects on a full scale stage, and very little in the way of similar work. This leaves the creative visionary with a world of possibilities with respect to choreographic choices, visual effects, and creative interpretation, all of which must be pieced together into a visually stunning performance. The challenge lies in part in searching the vast creative space as well as the desire to incorporate creative self-expression, which plays a central role in the arts.

In sharp contrast, our computer science team was given the well-defined goal of creating interactive technology that would work well in the theater space. This greatly limited our search space and provided a clear method for evaluating our work: If the technology works, then we're on the right

track. Our end goal can be defined as an "invention", where the focus is on the *usefulness* of our product - though in order to be a research project it also had to be novel. Unlike the goals of choreography in our project, self-expression played no notable part for the computer science team.

Another intriguing difference is how we view the importance of the process versus the final product. Innovation in the realm of computing tends to be an iterative process, where an idea may start out as a research effort, with intermediate steps demonstrated with a proof-of-concept implementation. Emphasis is placed on the methodology behind the new device or software product.

On the other hand, most dance choreographers focus primarily on the end result without necessarily emphasizing the methodology behind it. At all phases of the creative process, choreographers evaluate new ideas with a strong emphasis on how the finished product will be perceived by the audience. In the technological realm, the concern for general audience acceptance is only factored in later in the process.

During the early stages of ViFlow development, one of the critiques coming from dance instructors after seeing a trial performance was that "the audience will never realize all that went into the preliminary development process," and that the technique for rendering projections (i.e. pre-recorded vs. real-time with dancer movement tracking) is irrelevant to the final performance from an audience's point of view. In a sense, a finished dance performance does not make it a point to market its technological components, as this is merely an aspect of backstage production. Technology related products on the other hand are in large part differentiated not only based on the end goal and functionality, but also on the methodology behind the solution.

Conclusions

ViFlow has been created to provide a platform for the production of digitally enhanced dance performance that is approachable to choreographers with limited technical background. This is achieved by moving the creation of visual projection effects from the computer keyboard to the performance stage in a manner more closely matching the dance choreographic construction.

ViFlow integrates low-cost vision recognition hardware and video projection hardware with software developed at Florida State University. The prototype system has been successfully integrated into public performance pieces in the College of Dance and continues to be improved as new technology becomes available, and as we gain more experience with the ways in which choreographers choose to utilize the system.

The use of ViFlow empowers dancers to explore visualization techniques dynamically, at the same time and in the same manner as they explore dance technique and movement invention in the construction of a new performance. In doing so, ViFlow can significantly reduce production time and cost, while greatly enhancing the creative pallet for the choreographer. We anticipate that this relationship will continue into the future and hope that ViFlow will be adopted by other university dance programs and professional dance companies. While we have targeted production companies

as the primary target for ViFlow development, we believe that the algorithms can be used in a system targeting individual dancers who would like to explore interactive visualizations at home.

References

- [Bardainne and Mondot 2015] Bardainne, C., and Mondot, A. 2015. Searching for a digital performing art. In *Imagine Math 3*. Springer. 313–320.
- [Caillette, Galata, and Howard 2008] Caillette, F.; Galata, A.; and Howard, T. 2008. Real-time 3-d human body tracking using learnt models of behaviour. *Computer Vision and Image Understanding* 109(2):112–125.
- [Cohen 2013] Cohen, P. 2013. A new survey finds a drop in arts attendance. *New York Times*, September 26.
- [Corness, Seo, and Carlson 2015] Corness, G.; Seo, J. H.; and Carlson, K. 2015. Perceiving physical media agents: Exploring intention in a robot dance partner.
- [Franke 2012] Franke, D. 2012. Unnamed sound-sculpture. <http://onformative.com/work/unnamed-soundsculpture>. Accessed: 2016-02-29.
- [Jacob and Magerko 2015] Jacob, M., and Magerko, B. 2015. Interaction-based authoring for scalable co-creative agents. In *Proceedings of International Conference on Computational Creativity*.
- [Lee and Nevatia 2009] Lee, M. W., and Nevatia, R. 2009. Human pose tracking in monocular sequence using multi-level structured models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31(1):27–38.
- [Peursum, Venkatesh, and West 2010] Peursum, P.; Venkatesh, S.; and West, G. 2010. A study on smoothing for particle-filtered 3d human body tracking. *International Journal of Computer Vision* 87(1-2):53–74.
- [Sharma et al. 2013] Sharma, A.; Agarwal, M.; Sharma, A.; and Dhuria, P. 2013. Motion capture process, techniques and applications. *Int. J. Recent Innov. Trends Comput. Commun* 1:251–257.
- [Shingade and Ghotkar 2014] Shingade, A., and Ghotkar, A. 2014. Animation of 3d human model using markerless motion capture applied to sports. *arXiv preprint arXiv:1402.2363*.
- [Tepper 2008] Tepper, S. J. 2008. *Engaging art: the next great transformation of America's cultural life*. Routledge.
- [Toenjes and Reimer 2015] Toenjes, J. M., and Reimer, A. 2015. Lait the laboratory for audience interactive technologies: Dont turn it off turn it on!. In *The 21st International Symposium on Electronic Art*.
- [Wechsler, Weiß, and Dowling 2004] Wechsler, R.; Weiß, F.; and Dowling, P. 2004. Eyecon: A motion sensing tool for creating interactive dance, music, and video projections. In *Proceedings of the AISB 2004 COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems*, 74–79. Citeseer.