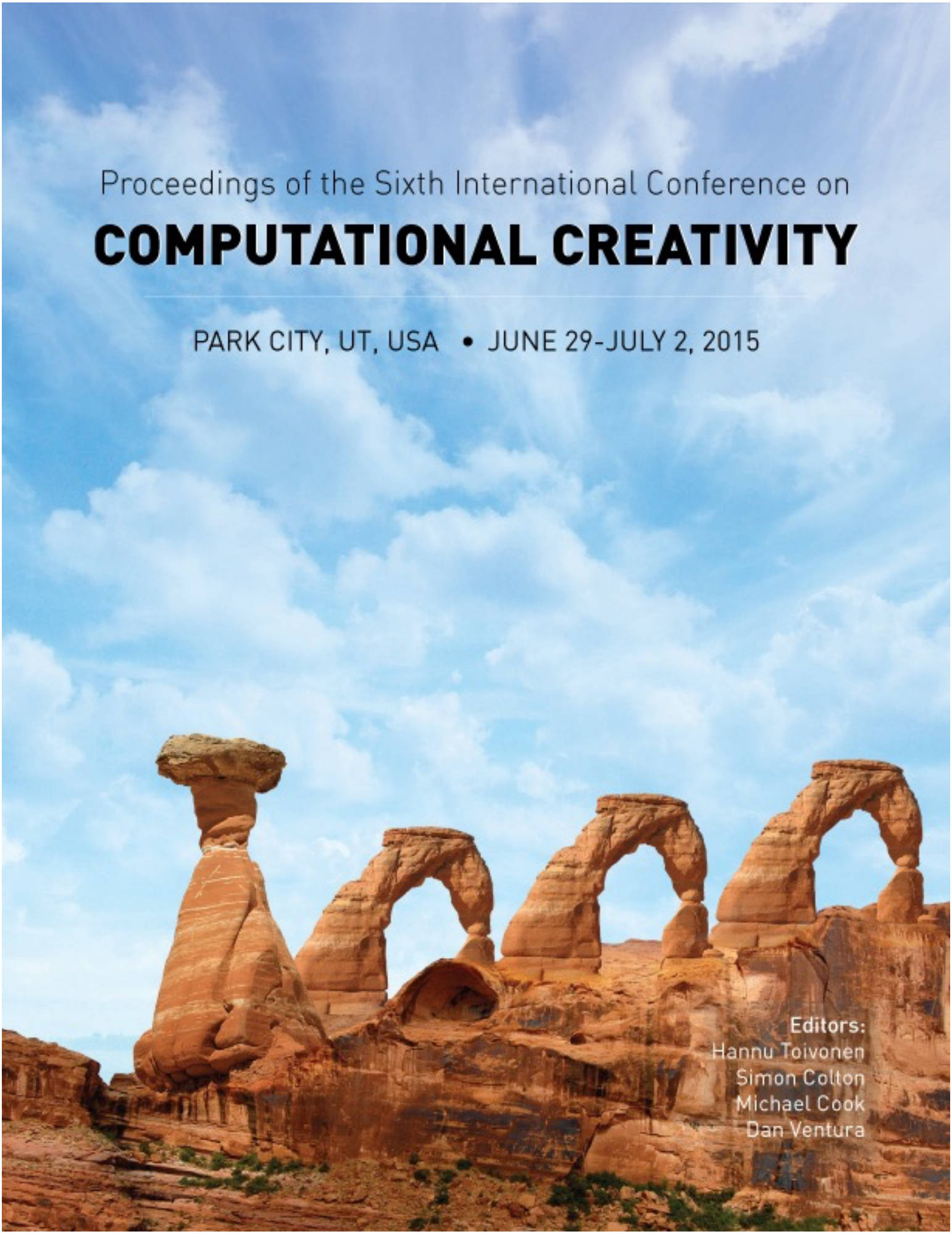
Proceedings of the Sixth International Conference on

# COMPUTATIONAL CREATIVITY

PARK CITY, UT, USA • JUNE 29-JULY 2, 2015



**Editors:**
Hannu Toivonen
Simon Colton
Michael Cook
Dan Ventura

Brigham Young University
Provo, Utah

http://computationalcreativity.net/iccc2015/

First published 2015

TITLE: PROCEEDINGS OF THE SIXTH INTERNATIONAL CONFERENCE ON COMPUTATIONAL CREATIVITY

EDITORS: Hannu Toivonen, Simon Colton, Michael Cook, Dan Ventura

ISBN: 978-0-8425-2970-9

About the cover: The cover design incorporates the famous red rock formations of southern Utah to represent the conference acronym, ICCC. The 'I' is formed using a common rock structure known as a hoodoo, and the three 'C's using three copies of iconic Delicate Arch, which is located in Arches National Park near Moab. Conceptual design by Dan Ventura. Technical Design by Simon Cho.

Technical editor: Martin Mumford

## Preface

Welcome to the Sixth International Conference on Computational Creativity, ICCC 2015!

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. The ICCC conference series has been organized annually by the Association for Computational Creativity since 2010, and was preceded by workshops since 2004. It is the only scientific conference that focuses on computational creativity and also covers all aspects of it.

Papers were submitted to ICCC 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.

The conference received 55 submissions from all over the world. The submissions were evaluated for their merits according to their category. Every submission was reviewed by three to four 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 200 reviews were carried out in the process.

Based on the reviews and discussions, 28 submissions were accepted for presentation. In an attempt to include more papers with interesting but possibly immature content, this year another 13 papers were accepted conditionally, subject to small revisions formulated and checked by the Senior Members. Eventually, all these papers were accepted in their revised forms.

The papers accepted to ICCC 2015 cover a large variety of topics. 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 three-and-a-half-day conference program of ICCC 2015 consists of exciting sessions consisting of presentation of the selected papers, a keynote on interactive narrative and participatory drama by Emily Short, some additional short talks, and a rich social program. This year, we are proud to host our very first workshop – Computational Creativity & Games – taking place immediately before the main conference. ICCC 2015 takes place June 29 – July 2 in Park City, Utah, USA, on historic Main Street at the Treasure Mountain Inn.

We would like to thank all those who invested their substantial efforts into making this conference what is: authors of the submissions for the excellent and interesting content of the conference, the 36 Program Committee members and the additional 10 Senior Program Committee members for their insight and efforts in the paper selection and shepherding process, Emily Short for a thought-provoking keynote, the workshop authors and organizers for interesting content to complement the main conference program, and the local organization in Utah for taking care of the practical arrangements.

We hope you are inspired by the papers and presentations!

| Hannu Toivonen | Dan Ventura | Michael Cook | Simon Colton |
|---|---|---|---|
| *Program Chair* | *Local Chair* | *Publicity Chair* | *General Chair* |

June 2015

**Conference Chairs**

General Chair: Simon Colton, Goldsmiths College, University of London
Program Chair: Hannu Toivonen, University of Helsinki, Finland
Publicity Chair: Michael Cook, Goldsmiths College, University of London
Local Chair: Dan Ventura, Brigham Young University

**Local Organizing Committee**

Thea Leonard, Gordon Billings, Jen Bonnet, Jenny Thornton, Martin Mumford, Simon Cho, Kimberly Jenkins, Brian Pugmire, Lynn Patten and Ye Liang

**Senior Program Committee**

Oliver Bown, University of Sydney
F. Amílcar Cardoso, University of Coimbra
Pablo Gervás, Universidad Complutense de Madrid
Rafael Pérez y Pérez, Universidad Autónoma Metropolitana at Cuajimalpa
Nick Montfort, Massachusetts Institute of Technology
Tony Veale, University College Dublin
Graeme Ritchie, University of Aberdeen
Geraint Wiggins, Queen Mary University of London
Rob Saunders, University of Sydney

**Program Committee**

Josep Blat, Universitat Pompeu Fabra
Alberto Diaz, Universidad Complutense de Madrid
Diarmuid O'Donoghue, National University of Ireland, Maynooth
Liane Gabora, Neological / UBC
Ashok Goel, Georgia Institute of Technology
Hugo Gonçalo Oliveira, CISUC, University of Coimbra
Jeremy Gow, Goldsmiths, University of London
Kazjon Grace, University of North Carolina at Charlotte
Andrés Gómez de Silva Garza, Instituto Tecnológico Autónomo de México
Raquel Hervás, Universidad Complutense de Madrid
Bipin Indurkhya, Akademia Górniczno-Hutnicza im. S. Staszica w Krakowie
Anna Jordanous, University of Kent
Amy K. Hoover, University of Central Florida
Robert Keller, Harvey Mudd College
Nada Lavrač, Jozef Stefan Institute
Carlos León, Universidad Complutense de Madrid
Antonios Liapis, IT University of Copenhagen
Ramon Lopez De Mantaras, IIIA – CSIC
Brian Magerko, Georgia Institute of Technology
Neil Maiden, City University London, Centre for HCI Design
Ruli Manurung, University of Indonesia

David Meredith, Aalborg University
Alexandre Miguel Pinto, University Of Coimbra
Santiago Negrete-Yankelevich, Universidad Autónoma Metropolitana
Francois Pachet, CSL Sony Paris
Philippe Pasquier, Simon Fraser University
Matthew Purver, Queen Mary University of London
Mark Riedl, Georgia Institute of Technology
Adam M. Smith, University of Washington
Oliviero Stock, FBK-IRST
Julian Togelius, New York University
Tatsuo Unemi, Soka University
Lav Varshney, University of Illinois at Urbana-Champaign
Georgios Yannakakis, University of Malta
Frank van der Velde, University of Twente

# Contents

**Keynote Talk 2015**

*Machine Improvisation on a Human-Authored Script: Beyond Versu*

Emily Short

**Biography:** Emily Short is a narrative design consultant with a special interest in interactive dialogue. She is the primary author of over two dozen works of interactive fiction, including Galatea and Alabaster, which focus on conversation as the main form of interaction; Mystery House Possessed, which creates a mystery with a randomly chosen, AI-driven murderer on each playthrough; and First Draft of the Revolution, an interactive epistolary novella. Her more puzzle-focused work includes the critically acclaimed wordplay game Counterfeit Monkey.

She has provided a variety of content creation, world-building, and narrative systems design services for large and small clients including ngmoco:), ArenaNet, and Failbetter Games, as well as finished white-label games for advertising and publishing clients. She is also part of the team behind Inform 7, a natural-language programming language for creating interactive fiction, where she assisted in feature design and wrote over 300 code examples included in the standard manual.

**Abstract:** Versu is a system for interactive narrative built around AI-driven agents who can choose among various pre-authored options for how to act; the largest story built in Versu is Blood & Laurels, a piece with over 150K words of content which could be combined in response to character and player actions into individual playthroughs running to about 15,000 words each. This project presented a number of challenges around authoring recombinable segments, setting the limits on permitted computer improvisation, and providing sufficiently flexible text to reflect the complexity of the underlying world model.

This talk reviews some of the challenges and successes of that project as well as the author's subsequent text generation work, all of which aims to produce participatory drama with playful, apt, and surprising juxtapositions of human-made text.

# The man behind the curtain: Overcoming skepticism about creative computing

**Martin Mumford and Dan Ventura**
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
martindm@byu.edu, ventura@cs.byu.edu

**Abstract**

The common misconception among non-specialists is that a computer program can only perform tasks which the programmer knows how to perform (albeit much faster). This leads to a belief that if an artificial system exhibits creative behavior, it only does so because it is leveraging the programmer's creativity. We review past efforts to evaluate creative systems and identify the biases against them. As evidenced in our case studies, a common bias indicates that creativity requires both intelligence and autonomy. We suggest that in order to overcome this skepticism, separation of programmer and program is crucial and that the program must be the responsible party for convincing the observer of this separation.

## Introduction

Demonstrations of computational creativity are often viewed with intense skepticism – much like a Victorian-era magician's trick full of smoke and mirrors. After all, creativity is regarded in many circles as a uniquely human characteristic, and so the claim of a *creative computer* invites immediate and often passionate skepticism. Even when an artificial system exhibits convincing creative behavior, the credit usually rests on the programmer as the true creative individual behind the act.

What can be done to convince the audience that there are no strings attached – that a program is being creative independently from its programmer? How far should they be allowed to probe, to test, and to know about the system's workings to be convinced?

It is important to motivate the separation of programmer and program in computational creativity applications. Consider a piece of software designed to monitor the landing gear on an aircraft. This software likely utilizes planning or decision-making algorithms, based on relevant conditions. If the software malfunctions in-flight, the aircraft may be damaged. Complex though the software may be, it cannot take the blame for following the instructions of its programming. Now consider a creative joke generator which tweets a new joke each day. One day, a generated joke happens to be highly offensive, and sparks criticism. This criticism cannot be targeted at the program, but at the programmer instead, for it is perceived to be following complex coded instructions. This is especially important as technology becomes more complex, and the general public becomes less aware of the specific details of its implementation. After all, as the science fiction author Arthur C. Clarke puts it:

> "Any sufficiently advanced technology is indistinguishable from magic."

This leaves us with a powerful motivator to understand how people perceive the division of creativity between creator and creation. Because computers are currently perceived as incapable of autonomy and thought, as programmers, we will be credited for *and* be held accountable for what our programs do.

In this paper we focus on the issues of perception and skepticism regarding artificial creativity. This discussion is hardly new but rather a modern revival motivated by recent progress in the field. As creative systems become more advanced, exhibiting more compelling creative behaviors, and applications begin to appear in the wild, the discussion becomes relevant again.

We outline a high-level review of suggested properties of creative systems, as well as previously proposed tests for evaluating the creativity of a system. We then report on a brief case study illustrating the impact of interactivity on perception. This is supplemented by a survey taken by software engineers, computer scientists, as well as non-specialists, which exposes some of the primary obstacles in the public perception of artificial creativity. We also offer an example from popular culture which highlights the issue of perceived autonomy as it relates to creativity. Finally, we discuss the impact of these perceptions on the potential direction and progress of the field.

## A History of Skepticism

The Lady Lovelace, upon hearing about the possible creativity of Charles Babbage's Analytical Engine, put forth the same argument that is still used today – as quoted in (Dartnall 1994), that "[Machines] have no pretensions whatever to *originate* anything," having no autonomous thought, and thus cannot be considered creative.

Nearly two hundred years later, despite significant advances in machine learning and computational creativity, this remains the dominant perception, with some degree of truth. In an attempt to address the question of whether or not

computers have the capacity for creative acts, several characteristics of creativity have been put forth by behavioral and computer scientists.

### Necessary and Sufficient Conditions

The search for qualities of creative systems is rooted in the question "What is creativity?" While an ill-formed and hotly contested question, it has nevertheless motivated scholars to seek out some of the necessary conditions to determine whether a system should be considered creative. The properties put forth so far are still subject to debate, and far from sufficient or exhaustive, but offer a guiding set of characteristics by which to begin judging the creativity of a system. An artificial system possessing many of these characteristics could be persuasively argued to be creative, because it shares those attributes with creative humans.

**Properties of the Artefact** The most straightforward way to judge a system is by the artefacts it produces. This requires no knowledge of system process, and success is often measured by comparing human-generated and computer-generated artefacts side-by-side or in a blind preference test.

Creative qualities artefacts should exhibit have included quality (Wiggins 2006; Colton 2008b), novelty or imagination (Ritchie 2007; Wiggins 2006; Colton 2008b), robustness or variability, and typicality (Ritchie 2007; Colton 2008b).

**Properties of the System** In addition to the artefacts, the process of creation itself has been suggested as a major factor in judging creative acts. Some of the aspects of the process include: appreciation or aesthetics (Colton 2008b; Colton, Pease, and Charnley 2011), individual style, intentionality, the ability to explain or justify decisions (Colton, Pease, and Charnley 2011), social context in a larger community of creators (Saunders and Gero 2001; Jennings 2010), and taking the audience into account (Maher, Brady, and Fisher 2013). Recent work has even been done on meta-evaluation – the evaluation of creative evaluation frameworks (Jordanous 2014).

Furthermore, we understand that the ability to learn is intertwined with the ability to create. A system that can learn its own fitness function for an aesthetic measure, for example, is arguably more creative than one that must have it explicitly specified by the developer, and some work has been done on automatically learning aesthetics (Colton 2008a).

### Tests of Computational Creativity

A few general psychological creativity tests exist but are often in a format inaccessible to computers. For example, the Torrance Tests of Creative Thinking (TTCT) involve many verbal and drawing tasks which are beyond the abilities of modern computer vision and natural language processing. And so, in addition to a set of essential qualities for creativity, academics have sought to define a "Turing Test" for creativity more suited to computers.

Even if a convincing, well-defined test existed, the concept itself has been criticized (Pease and Colton 2011) as limiting the potential style and variety of creativity in computers, much as the original psychological counterparts (Kim 2006) have been criticized.

Turing Tests have been subject to scrutiny by the Chinese Room argument (Searle 1980), which appears to coincide with the most common criticism of creative systems – that no matter how creative they may seem, their internal workings could still comprise some form of Searle's rule-book. The Lovelace Test (Bringsjord, Bello, and Ferrucci 2003) tries to address this issue by dealing with the separation of programmer and program, rather than focusing on the system exclusively. Specifically, one of the requirements of the Lovelace Test is that the programmer cannot explain how an artefact was generated by the system, even when given ample time to do so.

Notably, Bringsjord implies that the Lovelace Test can only essentially be passed when a system is perceived of as 'thinking for itself', and 'having a mind'. The perception of creativity is thoroughly entangled with the perception of intelligence and autonomy. While programmer surprise and inability to explain can help to establish the system as a separate entity, such surprise can be faked. Overcoming residual skepticism may require methods that establish the autonomy of a system without the need to rely on programmer reactions.

## Modern Skepticism

In its current state, the field of Computational Creativity continues to face heavy skepticism from non-specialists. This is actually quite healthy for our field, as such skepticism provides a motivation to build systems that are not only theoretically sound, but convincingly demonstrable and socially acceptable. We explored the primary complaints and biases against the notion of creative computers, with the intent to discover the core issues that need to be addressed. This exploration revolved around the question, "What would it take to subjectively convince someone of a system's creativity?"

### Man behind the curtain: A case study

In order to explore what it would take to alter people's perception, we created a simple analogy-making program, the output of which might be considered creative. This program was presented in three stages to 35 participants who were told that it was powered by a creative artificial intelligence.

- Stage one: No interactivity. The user presses a button and the computer produces a random analogy.
- Stage two: Selective interactivity. The user selects two nouns from a short list, and the computer produces an analogy between them.
- Stage three: Full interactivity. The user inputs any two concepts, and the computer produces an analogy.

The first two stages only *appear* to be creative – but in reality the computer is selecting from a pool of pre-generated analogies. Although the analogies could have been retrieved nearly instantly, a loading screen was presented to give the appearance of processing happening 'behind the curtain.'

The pre-generated analogies were created by hand using two seemingly unrelated concepts, and connected in a clever

and humorous way using similar properties between the two. For example, 'cats are like lawnmowers: temperamental and destructive.' For stage two, items could be selected from two lists of five, making 25 possible analogies in the pool.

Since we did not actually construct a creative analogy generator, the only way to provide full interactivity was to utilize a human operator using a networked device to 'respond' to analogy requests. In our case we actually placed a man behind a curtain – the operator was sitting behind a partition nearby as users participated. In order to ensure consistent quality and style of analogies between different stages, the writer of the analogies for the first two stages also served as the operator for the third.

Users were asked at random to either participate in one of the three tiers, or to move through all three consecutively. After observing the analogies that were 'generated' by the computer, they were asked to evaluate the creativity of the task in general, as well as to determine where they felt the attribution of creativity belonged on a 5-point Likert scale from programmer to program.

First, we observed that as the degree of allowed interactivity increased, the users were more inclined to test the system for patterns or trickery. When asked to split the attribution of creativity between programmer and program, a 1.0 on the scale represented 'all programmer' and 5.0 represented 'all program', where 3.0 represented an equal responsibility between the two. The average placement was 2.25 for stage 1, 2.46 for stage 2, and 3.1 for stage 3, showing an improved willingness to attribute creativity to the computer.

Second, we observed that those who tried successively more interactive levels attributed dramatically more creativity to the system (more so than those participating in individual tests). This is likely because they had to revise their own assessments multiple times.

Finally, among the highly skeptical, we found that a clear, repeated input-output pattern caused any and all creativity of the system to be discounted. Because the first two tests simulated a creative system by drawing from a pool of pregenerated analogies, and that pool was not particularly deep, astute users would probe the system until it eventually produced a duplicate. Each user who discovered a duplicate would invariably rate the system as having low creativity.

## Conflicts

There also exist a few 'double-edged swords' in a creative system that can subjectively decrease or increase the perception of creativity.

**Knowledge of System** Keeping the system as a black-box (no knowledge) forces the user to evaluate the system based on the artefacts alone. Unfortunately this can mask the true creativity or lack of creativity in a system. For some individuals, keeping the system internals unknown is crucial, based on the notion that creative people produce artefacts *ex nihilo*, or that the creative process is fundamentally mysterious and cannot be explained. To expose the process might disrupt the appearance of creativity for these individuals.

For example, it is trivial to implement a genetic algorithm to *evolve* a painting of the Mona Lisa, simply by setting the fitness function to be a pixel-by-pixel comparison between the phenotype and a picture of the Mona Lisa. Yet watching the painting evolve and take form in real time, it is easy for an outside observer to attribute to the program some level of intelligence and creativity. Of course, had the curtain been pulled back and the process exposed to the observer, they would have been disappointed at the naive way in which the system randomly combines and mutates.

Exposing the high-level workings of the system allows the observer to make judgments about the process itself. However, exposing all of the system's process could remove the mystery of the process, leading to the perception that the program is 'merely following instructions,' no matter how complex they may be.

In our analogy-making experiment, several technically-minded users attempted to discover the internal workings, inventing progressively harder requests meant to probe for templates and patterns. These individuals were impressed if they could not determine a consistent pattern, and remained unconvinced if they could imagine a clear process by which the artefacts were generated.

**Humanized Process** People tend to project human emotions and behaviors onto non-human objects. A process that seems more 'human' (pausing as if in thought, backtracking, slight errors, etc.) can improve the perception of creativity. As Colton observes (2008b),

> "...it is apparent that being able to watch The Painting Fool create its paintings means that people project more value onto them than they would if the paintings were rapidly generated through, say, an image filtering process. This seems to be because they can project critical thought processes onto the software, and empathise with it more."

On the other hand a process with elements that appear highly computer-like (superhuman speed, enormous scale, lack of mistakes, logical explanations, etc.) can sometimes lend strength to the perception that a computer is doing all the work. Ultimately, the most persuasive portrayal might incorporate aspects of both philosophies.

## The Creative Threshold

We conducted three surveys among different audiences asking about computers and creativity. Each participant was asked to rate whether computers were currently capable of creativity, and whether they will someday be capable of creativity, on a Likert scale from 0 to 10. They were then asked to define what they thought were essential requirements or characteristics of creativity. Finally, they were asked to describe what behavior or characteristics a system should have to convince them that it was creative. The exact questions and selected responses can be found in Appendix A.

We first sought to understand the opinion of those that were technologically literate, but unfamiliar with programming and code. This survey was conducted on Reddit (a social bulletin board website) and had 75 respondents. We did not collect demographic information, but general statistics of Reddit users are can be found elsewhere (Duggan and Smith 2013) for those interested.

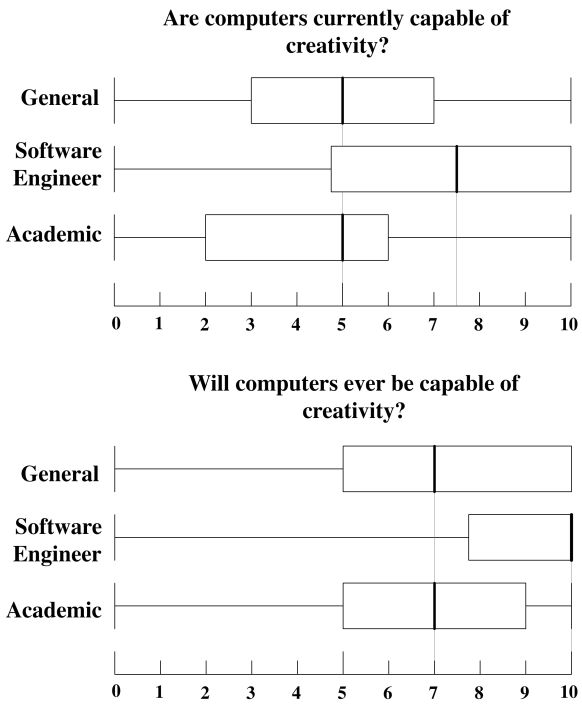



Figure 1: Quantitative analysis of responses by group: each boxplot shows the first quartile (left), median (bold) and third quartile (right).

For comparison, the same survey was given to a group of 26 software engineers working in the industry, and again to a group of 37 computer science professors and graduate students at Brigham Young University.

We originally anticipated that people familiar with programming or AI would have a deeper understanding of its potential, and thus show less skepticism at the concept of computational creativity. Academics, being the most familiar with current research and progress were expected to show the strongest optimism. However, the academics surveyed displayed somewhat more skepticism than any other group. More surprising still, the programmers demonstrated a disproportionately high level of confidence.

Among the open-ended responses in all three groups about the requirements for creativity, eight broad classes emerged:

- Lateral Thinking: Often described as 'outside the box', including methods of thinking that 'do not rely on logic,' going beyond formal inductive and deductive reasoning.
- Flexibility: The ability to work within arbitrary constraints and handle many kinds of tasks.
- Aesthetics: Taste, or the ability to judge quality and discern good artefacts from bad ones.
- Novelty: Producing artefacts which are original, unique, or different from what has been seen before.
- Analogy: The ability to make interesting analogies between seemingly unrelated concepts, or to combine or otherwise transform old concepts into something new.
- Self-Improvement: The ability to learn from experience over time.
- Autonomy: Often described as 'independent thought', 'unique intelligence', or emphasizing a lack of predefined rules.
- Human Emotions: bravery and curiosity were the most common human emotions listed.

Particularly among the most skeptical participants (those who rated it unlikely that computers are or ever will be creative), autonomy was the top priority for creativity. Responses such as, 'agency', 'choose for itself', 'independent intellectual ability', and 'independent thought' suggested that the system must be autonomous to convince them. Consider the following responses specifically about code: 'not based on algorithms', 'not a result of programming', 'create its own programs', 'no explicit code detailing what to do', and 'write the program on its own'.

Of course, computer programs can already exceed their original programming, through machine learning for example. Decades ago, classical AI algorithms were already capable of learning things that their creators did not know, and acquiring skills that their creators did not possess. The observed unwillingness to acknowledge a program as an independent entity appears to stem from a philosophical standpoint, even among other computer scientists, that code merely follows instructions (albeit extremely complex ones). This is a valid point of debate, though a particularly fuzzy one, since even creative humans could be argued to be following a complex set of chemical and psychological instructions.

This need for an intelligent autonomous entity separate from the programmer sparks interesting questions. Is it possible for a computer system to possess all of the creative attributes typically outlined in our field (appreciation, skill, novelty, typicality, intentionality, learning, individual style, curiosity, accountability), and yet *still* not be creative? Alternatively, can a machine be creative without being intelligent? More broadly, is *general* or *strong* artificial intelligence necessary before people become comfortable with ascribing creativity to a machine?

We are not prepared to claim that general intelligence is required for creative behavior, but instead observe that people are generally unwilling to *attribute* creativity to a system until it appears to be a separate, intelligent entity.

## In popular culture

We turn to a portrayal of creative computing in popular culture to demonstrate the perception that in order to be creative, a computer must have autonomous thought and exceed its programming.

In an episode of the television series Star Trek: Voyager, a trial is conducted to determine whether a computer program (the holographic doctor) should retain the rights to the creative work (holonovel) which he created. Part of the trial

appeals to the argument that attributes of the artefact are enough to deem a computer creative:

> BROHT:A replicator created this cup of coffee. Should that replicator be able to determine whether or not I can drink it?
> TUVOK: But I have never encountered a replicator that could compose music, or paint landscapes, or perform microsurgery. Have you? Would you say that you have a reputation for publishing respected, original works of literature?
> BROHT: I'd like to think so.
> TUVOK: Has there ever been another work written about a hologram's struggle for equality?
> BROHT: Not that I know of.
> TUVOK: Then in that respect, it is original.
> BROHT: I suppose so.
> TUVOK: Your honour, Section seven ... defines an artist as a person who creates an original artistic work. Mister Broht admits that the Doctor created this programme and that it is original. I therefore submit that the Doctor should be entitled to all rights and privileges accorded an artist under the law.

However, the appeal to originality was ultimately not enough evidence to convince the judges. The winning argument rested on the doctor's autonomy and independent thought:

> KIM: He decided it wasn't enough to be just a doctor, so he added command subroutines to his matrix and now, in an emergency, he's as capable as any bridge officer.
> ARBITRATOR: That only proves the Doctor's programme can be modified.
> KIM: Your honour, I think it shows he has a desire to become more than he is, just like any other person.
> JANEWAY: Starfleet had programmed him to follow orders. The fact that he was capable of doing otherwise proves that he can think for himself.

In this fictional case, as with the personal biases discovered in the survey, the deciding factor is intelligent, autonomous thought. This gives rise to several open questions for discussion:

- In what way are different aspects of intelligence interrelated with different aspects of creativity?
- Is intelligence necessary for creativity?
- If so, is artificial general intelligence necessary for general creativity?
- Is the threshold of evaluating creativity arbitrarily lower for humans or living beings such as crows (which have been shown to solve problems creatively) than for inanimate systems like programs?
- Though increasing the *intelligence* of our creative programs could boost creative perception, would it necessarily have a positive impact on the true creativity of the system, or the quality of artefacts it produces?
- How best can we convincingly demonstrate the autonomy of a creative system?

## Future Skepticism

There are many current approaches we can utilize to overcome some of the perceptual barriers, one of which is the capacity for a program to code parts of itself. Work is already being conducted in *creative code generation* (Cook 2013), which could boost the perception of autonomy by non-specialists. Metaprogramming (writing code that writes code) does not necessarily translate to more creative programs, but it certainly lends credence to the idea that the program is separate from the programmer. This in turn provides an entity other than the programmer to which creativity can be attributed. Additionally, using machine learning methods to improve a system's aesthetic sense, cognitive ability, or skill level strengthens the claim that it is able to 'exceed its original programming'.

More broadly, we need to consider the impact of these perceptual issues on the goals of our field as a whole. To what extent should public opinion factor into our goals? Several of the requirements for creativity are already shared by both public opinion and computational creativity researchers. A heavier emphasis on boosting perception may only serve as a motivation for trickery and selective methods of presentation, which would not necessarily increase the creativity of our systems or the quality of artefacts they produce.

Consider the difference between the aircraft landing gear software and the joke generator in the introduction. We understand there is a creative difference between aircraft software and a joke generator. Aircraft software was designed to be predictable and react to very particular situations in very particular ways – a clear mapping from inputs to outputs. Thus a software failure is likely to be the fault of the programmer. However, a joke generator is ideally unpredictable – *that's the point*. Its creator may be surprised at the jokes it generates, but the audience cannot necessarily ascribe this to the generator program being an autonomous entity. It could then be argued that the programmer is indeed responsible for the offensive joke, but unknowingly so, because the programmer was unaware of the range of possible jokes that the program could generate.

A parent is socially responsible for the behavior of their child, but they cannot take credit for the child's creative acts or creative capacity, and nor can a mentor or teacher. However this relationship changes dramatically in software, where the programmer is not merely training an existing system, but making architectural decisions about the way it should think. If we could manipulate or condition the human brain to be more creative, or to deliberately specify how the thought process works, would the credit for the individual's creative acts rest partly on us?

A primary goal of our field is to shift the burden of creativity from ourselves to our programs. However, our level of direct involvement in the minds of our machines makes this transference difficult, despite our best efforts to facilitate it. The philosophical question to ask is whether this difficulty is entirely a matter of perception, in which case it is a problem of persuasion, or whether more of ourselves resides in the machine than we would like to admit. This entanglement between creator and creation may be unavoid-

able, until our creative systems can be considered separate, intelligent entities with *independent thought*, at which point we open an entirely different can of worms.

## References


Bringsjord, S.; Bello, P.; and Ferrucci, D. 2003. Creativity, the Turing test, and the (better) Lovelace test. In *The Turing Test*. Springer. 215–239.

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. 2008a. Automatic invention of fitness functions with application to scene generation. In *Applications of Evolutionary Computing*. Springer. 381–391.

Colton, S. 2008b. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.

Cook, M. 2013. Creativity in code: generating rules for video games. *ACM Crossroads* 19(4):40–43.

Dartnall, T. 1994. *Artificial Intelligence and Creativity: An Interdisciplinary Approach*, volume 17. Springer Science & Business Media.

Duggan, M., and Smith, A. 2013. 6% of online adults are Reddit users. *Pew Internet & American Life Project* 3.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Jordanous, A. 2014. Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proceedings of the 5th International Conference on Computational Creativity*.

Kim, K. H. 2006. Can we trust creativity tests? a review of the Torrance tests of creative thinking (TTCT). *Creativity Research Journal* 18(1):3–14.

Maher, M. L.; Brady, K.; and Fisher, D. H. 2013. Computational models of surprise in evaluating creative design. In *Proceedings of the 4th International Conference on Computational Creativity*, 147–151.

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 Symposium on AI and Philosophy*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. In *Computational and Cognitive Models of Creative Design V*. Key Centre of Design Computing and Cognition, University of Sydney. 113–139.

Searle, J. R. 1980. Minds, brains, and programs. *Behavioral and Brain Sciences* 3(3):417–424.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Appendix A: Survey Responses

- Question 1: Do you think that computers are currently capable of being creative?
- Question 2: Do you think computers will ever be capable of creativity?
- Question 3: Name a couple of capabilities or traits required for someone to be considered 'creative'
- Question 4: Briefly, what would a computer program have to do to convince you that it (not the programmer) was being creative?

Selected responses:

| Q1 | Q2 | Q4 | Q5 |
|---|---|---|---|
| 0 | 1 | Predictive capacity, Agency, Contextual analysis | Prove to me it has agency to choose for itself |
| 6 | 8 | Iterative thinking and creation, ability to change direction mid-production, show work | Show steps, come to different conclusions when fed similar data/asked similar questions |
| 3 | 5 | must be a sentient being | since the AI would likely learn through formulas/programs created by the programmer, if it could create its own programs that are beyond human comprehension then that would be creative |
| 9 | 10 | New Ideas, Take an old idea and adapt it to a new situation | Maybe create a recognizable graphic from lines or circles or something Or respond to questions asked in ways that were unexpected and unpredictable |
| 7 | 9 | Come up with a new and unseen "thing" or take something old and use in a new or different way | Do not know |
| 6 | 7 | problem solving | solve a problem using non-data inputs or observations |
| 0 | 3 | Original thought, inspiration | Come up with an idea that hadn't been thought of before |
| 7 | 10 | Not merely following rules, affect and logic combined | It would have to modify its own programs |
| 3 | 4 | capable of thinking "outside the box", coming up with innovative solutions to various problems | manifest fully independent intellectual ability |
| 10 | 10 | free thought | adapt to change |
| 10 | 10 | something able to come up with new ideas | respond to complex questions and problem solve |
| 5 | 6 | Think of ideas and new things on your own | Write the program on its own to show its creativity |
| 4 | 9 | innovation, unorthodox solutions | create a new idea |
| 3 | 6 | Inventive, open minded, designer | Synthesize to make something unique and relative to a need, feeling, etc. May have an aesthetic component |
| 4 | 6 | free choice | make something creative w/o human input |

# Generating Code For Expressing Simple Preferences: Moving On From Hardcoding And Randomness

**Michael Cook and Simon Colton**
Computational Creativity Group
Goldsmiths, University of London
gamesbyangelina.org — ccg.doc.gold.ac.uk

**Abstract**

Software expressing intent and justifying creative decisions are important considerations when building systems in the context of Computational Creativity. However, getting software to express subjective opinions like simple preferences is difficult without mimicking existing people's opinions or using random choice. In this paper, we propose an alternative way of enabling software to make meaningful decisions in small-scale subjective scenarios, such as choosing a favourite colour. Our system uses a combination of metrics as a fitness function for evolving short pieces of code that choose between artefacts. These 'preference functions' can make choices between simple items that are neither random nor based on an already existing opinion, and additionally have a sense of consistency. We describe the system, offer some example results from the work and suggest how this might lead to further developments in generative subjectivity in the future.

## Introduction

Computationally creative software usually makes many decisions in the process of producing an artefact. These decisions are often in the context of problems for which 'notions of optimality are not defined' (Eigenfeldt, Burnett, and Pasquier 2012) and so there is no definitive equation or objective measure that can guide them to the 'best' answer. As a compromise, the developers of such software provide ways to guide the software in making these decisions: sometimes by providing predetermined heuristics; sometimes by allowing the software to create models trained on decisions made by people; sometimes using random chance.

In many of these creative decisions there are no right or wrong answers. For example, in (Veale 2013) a system writes poetry by first generating several potential metaphors to work from. These metaphors are all considered good candidates that could produce poems – the system selects one at random, because it has no meaningful reason to choose between them. This is a small decision within a much larger system, and in many ways it is insignificant compared to the larger creative act the software performs. In this paper, however, we argue that there are two important consequences to relying on random choice or predetermined heuristics for decisions such as this. Firstly, we prevent our software from intelligently discussing these choices in framing information, and as a result miss out on opportunities to add value to the artefacts created or raise the perception of our software as creative (Colton, Charnley, and Pease 2011). Secondly, when observers discover or are informed that these choices are made due to external factors or randomness, then we contend that their perception of the software as creative is lowered significantly, even if the decision in question seems trivial.

Software is not human – it does not have emotional attachments, it does not have childhood memories, it does not have biochemical reactions. This does not mean, however, that we must shy away from providing software with the ability to make and justify subjective decisions. If the claims that it makes about its preferences are *consistent*, *defensible* and *reasonable*, we believe that this will add to the perception of the software being creative without deceiving the observer about the software's lack of humanity.

We describe here a system that can generate simple snippets of code, which we call *preference functions*, that take as input two objects of some type and express an ordering on them – in other words, they express what amounts to a *preference* between the two objects. This system works by evolving code segments, using a particular combination of metrics, which we also introduce here, as a fitness function. These metrics have been carefully designed to be domain agnostic, and to limit our influence as designers on the output the system ultimately produces in terms of the subjectivity it expresses. While this process is not perfect, we believe this work represents an encouraging first step towards software making meaningful subjective decisions. To illustrate this, we provide several examples of generated functions in different domains, including colour selection and videogame design, that highlight how this technique might be used in software. We then discuss what further work is needed to integrate this technique into the framing and context of computationally creative software.

## Background

### Framing and Subjectivity

Framing is the name given to the process by which software produces text or perhaps other content to provide context to a generated artefact. Thus far in Computational Creativ-

ity this generally takes the form of a 'wall text'-like commentary that appears alongside the artefact in order to help explain the creative process, as in (Colton, Goodwin, and Veale 2012). According to (Colton, Charnley, and Pease 2011), the authors claim that the act of framing can increase the 'value' of generative acts undertaken by software in several ways, one of which is 'by providing calculations about the concepts/expressions [in an artefact] with respect to the aesthetic measures'. In (Colton, Goodwin, and Veale 2012), for example, the software generates commentaries which explains why it chose particular poetic styles or focused on particular words.

In (Charnley, Pease, and Colton 2012), the authors consider three particular aspects of a creative work that framing can tackle: motivation, intention and process. The authors summarise these as 'Why did you do that?', 'What did you mean when you did that?' and 'How did you do that?' respectively. Of motivation, they say:

> *[it is] distinctly human in nature and it currently makes limited sense to speak of the life or attitudes of software in any real sense.*

However, the authors also point out later that 'framing need not be factually accurate', and that 'the motivation of a software creator may come from a bespoke process which has no basis in how humans are motivated'. We claim that it is reasonable for software to possess arbitrary or subjective preference about elements of its creative process, for the purposes of framing and justifying its motivation and output. The technique we outline in this paper has no basis in how people are motivated, as in the quote above, but it does aim to offer a form of motivation for software's actions that is satisfying to the observer and may withstand limited interrogation through framing or even dialogue.

### Randomness and Believability

In (Colton and Wiggins 2012) the authors define Computational Creativity as the creation of systems which 'exhibit behaviours that unbiased observers would deem to be creative' (paraphrased). The mention of unbiased observers is crucial to the definition, since Computational Creativity is highly reliant on the *perception* of creativity. A common criticism of creative software is that the designer of the software is a major contributor to the software's creativity. (Colton 2009) proposes a process of 'climbing the meta-mountain' to overcome this, whereby creative software is iteratively improved to remove the influence of the original designer on the software, instead adding in new subsystems which take the place of the designer's involvement and allow the software to make the same decisions for itself.

The danger of removing designer influence for removal's sake is that the system that replaces the designer's involvement may not actually increase the perception of creativity. There is anecdotal evidence to suggest that people distrust the actions of software, even in cases where the software is proactively explaining that its decisions were intelligently motivated. The work described in (Cook and Colton 2014), for example, provoked an angry response from one member of the public who wrote 'AI, or just basic random number generation?' in response to the software framing its choice of a piece of music.

There are many explanations for why people might be biased against software in some instances, one being that they have good cause to be suspicious: random choice *is* used very often in the design of intelligent systems, including those in Computational Creativity. Moreover, as we have already stated, researchers are not afraid to have their software tell stories that are 'not factually accurate' in order to explain their decisions. This is not a dying practice, either: examining system description papers from the 2014 International Conference on Computational Creativity alone, we identified seven systems which explicitly mention random decision-making in their description (omitting cases where random selection might be part of a search-specific process, such as evolution) such as (Rashel and Manurung 2014), a poetry generator which randomly selects an output from any poems which meet a minimum quality, or (Tomašič, Žnidaršič, and Papa 2014) which breaks ties in slogan selection using random choice. Other systems described relying on hand-crafted metrics for making subjective decisions which inherit their decision-making capacity directly from the system's designer.

We believe that the underlying cause for this bias against software making decisions independently is not that people believe that software *cannot* make such decisions, but rather that random choice is not satisfying as a context for these decisions. Random choice cannot be interrogated or understood, does not form a long-term pattern of decision-making, and is also not something that people often do – even when people may in fact be making pseudorandom decisions, we often justify them post-hoc, particularly in the case of creative activity – see (Charnley, Pease, and Colton 2012) for examples. Most importantly, random choice cannot be easily framed through commentary on a creative artefact, because it has no context to reveal. This limits the software's ability to explain itself after the fact.

## A System For Generating Preferences

If we acknowledge that inheriting decisions from a person damages the perception of software as being creative, but also accept that random decision-making is unsatisfying and can be equally damaging to perceptions, it leaves us in an awkward position whenever our software must tackle decisions which are subjective or where the factors involved are hard to quantify. Ideally, we would like our software to be able to provide meaningful reasons for small, subjective decisions. By meaningful, we mean that the decision is *defensible* in some way: there is a reasoning behind it, even if that reasoning is ultimately arguable (as subjective opinions often are, by their nature). In this section we will describe an evolutionary system that generates code to provide the basis for such decisions, with the primary aim being that these decisions are defensible, despite being subjective.

The system we describe here generates what we call *preference functions* – small snippets of code which express a preference of some kind between two objects of the same type. They are based on the concept of Comparators in Java

which are used to express orderings over lists. A Comparator takes two objects and returns either -1, 0 or 1 if the first object is less than, equal to, or greater than the second object respectively according to some ordering. Our functions act similarly, where a preference can be thought of as an ordering over the set of objects of a particular type. More formally, we define a preference function $p$ as a function which takes two arguments $t_1, t_2$ of type $T$, and returns one of three integer values $r \in \{-1, 0, 1\}$. The return value indicates the following three situations:

$$p(t_1, t_2) = \begin{cases} 1 & t_1 >_p t_2 \\ 0 & t_1 =_p t_2 \\ -1 & t_1 <_p t_2 \end{cases}$$

Where $*_p$ is an ordering according to preference, i.e. $t_1 <_p t_2$ states that $t_1$ is preferred over $t_2$ in some way. The comparator can therefore be used to order a list $L_T$ of objects of type $T$.

Before we describe the operation of the evolutionary system, we will go into some detail about the fitness function that evaluates a particular preference function. Earlier, we claimed that our intention was to limit the influence of a person's opinion over the system's eventual decisions. Below we will propose metrics which direct the search for preference functions – in some sense we are defining the kinds of preferences the system looks for. We will try to justify our decisions and show that these metrics are flexible, domain-agnostic, and aim for defensibility without specifying anything about what kinds of preference should be expressed.

## Fitness Metrics

In this section we describe several metrics that can be used to assess certain qualities of a preference function. This does not make a judgement about the 'goodness' of the preference expressed; a preference function which scores higher on these criteria is not objectively better than one with lower scores. Rather, we aim to identify meta-level properties of preference functions in order to search for spaces of interesting, valid or defensible preferences. As a result, we've tried to avoid the use of emotive or judgemental vocabulary when describing the metrics.

**Specificity** The *specificity* of a preference function $p$ for a set of objects $O$ is defined as:

$$1 - \frac{|N_p|}{|P|}$$

with

$$\begin{aligned} P = &\quad \{(a,b) \mid (a,b) \in O \times O \wedge a \neq b\} \\ N_p = &\quad \{(a,b) \mid (a,b) \in P \wedge p(a,b) = 0\} \end{aligned}$$

In other words: the specificity of a preference function for a particular list of objects is the proportion of the list for which it returns a nonzero result, i.e. a definite preference. Note that this excludes identity preferences (you can't prefer something to the same thing), but it does not assume transitivity on $p$ and it includes reflexive preference, i.e. $p(a,b)$ and $p(b,a)$.

**Transitive Consistency** The *transitive consistency* of a preference function $p$ for a set of objects $O$ is defined as:

$$\frac{|T_p|}{|Q|}$$

with

$$\begin{aligned} Q = &\quad \left\{ (a,b,c) \;\middle|\; \begin{array}{l} (a,b,c) \in (O \times O \times O) \\ \wedge\, a \neq b \wedge b \neq c \wedge a \neq c \end{array} \right\} \\ T_p = &\quad \{(a,b,c) \mid (a,b,c) \in Q \wedge tight_p(a,b,c)\} \end{aligned}$$

Where $tight_p$ holds for a triple $(a,b,c)$ if the triple is transitively non-contradictory under the preference function $p$. That is:

$$a \geq_p b \wedge b \geq_p c \implies a \geq_p c$$

In other words, transitive consistency is a measure of how much the decisions made by the preference function confirm one another when compared alongside each other. A high transitive consistency means that the preference is well-ordered. As with other metrics, this is not inherently good or bad. Low transitive consistency can imply that the preference selects based on unconnected or competing features in the artefacts, which is not uncommon in everyday preferences.

**Reflexivity** The *reflexivity* of a preference function $p$ for a set of objects $O$ is defined as:

$$\frac{|P_r|}{|P|}$$

with

$$P_r = \quad \{(a,b) \mid (a,b) \in P \wedge p(a,b) = -p(b,a)\}$$

In other words, reflexivity is a measure of how dependent $p$ is on the ordering of its arguments. A high reflexivity suggests that the preference being expressed is not dependent on the arguments being supplied to it. This metric is useful specifically because of how we generate code, since it is possible to generate functions which make decisions based on the ordering of their parameters (always preferring the first parameter, for example). High reflexivity means that no parameter is preferred over another simply because of the order they are passed in.

**Agreement** Agreement is a special metric for comparing two preference functions. This isn't used to generate preferences, but can be used to compare them, or generate functions in opposition to one another. Two preference functions $p_1$ and $p_2$ are said to be in {k,n}-agreement iff:

$$k \leq \frac{|P_{p_1,p_2}|}{|P|}$$

with

$$P_{p_1,p_2} = \quad \left\{ (a,b) \;\middle|\; \begin{array}{l} (a,b) \in P \wedge \\ \left( \begin{array}{l} p_1(a,b) = p_2(a,b) \vee \\ p_1(a,b) = 0 \vee p_2(a,b) = 0 \end{array} \right) \end{array} \right\}$$

With $P$ and $O$ as before, and where $|O| = n$. In other words, agreement measures how closely the definite decisions of

two preference functions are the same – the proportion of pairs $(a, b)$ for which $p_1$ and $p_2$ either evaluate the same value or one of them is zero, is greater than $k$ for a list of objects of size $n$. This is a good measure of how close two preference functions are on the same sample of inputs.

To summarise, the first three metrics judge preference functions along several dimensions: how often they make a definite (nonzero) judgement on two objects, how consistent their judgements are across a list of objects, and how dependent the decisions are on the random ordering of inputs. The final metric, agreement, can be used to model how similar or dissimilar two preference functions are on the same set of inputs. We will now describe the preference function generation system, which uses a combination of the first three metrics in its evaluation.

## Representing Preference Functions

In the following subsections we describe an evolutionary system implemented in C# which uses the CodeDOM API to generate code segments which act as the body of a preference function. The language and library are arbitrary choices for convenience, and should be transferable to any platform for which code generation is possible.

CodeDOM is an API within Microsoft's .NET library that allows for the high-level (and extremely verbose) representation of code, which can later be exported to `.cs` files and then compiled into executable assemblies within C#. The code below is equivalent to `(p && q)` in C#, where `p` and `q` are local variables:

```
new CodeBinaryOperatorExpression(
  new CodeVariableDeclarationStatement("p"),
  CodeBinaryOperatorType.BooleanAnd,
  new CodeVariableDeclarationStatement("q"));
```

CodeDOM compilation units can be exported to code programatically, and these files can be compiled and executed at runtime using C# CodeProvider classes. CodeDOM represents almost every aspect of the C# language, but for our purposes we do not extend the code generation to the entire C# specification, as this yields diminishing returns and is too large a state space for this stage of experimentation. Generation of the preference functions described here is limited to:

### Expressions

- Primitive expressions containing `int`, `bool` or `String` types.
- References to the object parameters given to the function.
- Boolean binary operations including `&&` and `||`.
- Numeric binary operations.
- Type casts between certain compatible types.
- Array index references.
- Field accesses in objects.

### Statements

- Conditional control flow statements (`if` statements)
- Assignment
- Return of either $-1$, $0$ or $1$.

We define a *code segment* as a list of one or more statements. This can be put inside a CodeDOM representation of a method. We define a generic abstract template class which defines a method `public int compare(int a, int b)`[1]. Generated code segments are put inside a class which extends this abstract template, providing an implementation for the `compare` method.

## Evolving Preference Functions

A population of code segments is randomly generated, and evaluated using the following objective function:

$$\begin{aligned} fitness(p) &= 0.5 \times reflexivity(p) \\ &+ 0.25 \times specificity(p) \\ &+ 0.25 \times consistency(p) \end{aligned}$$

This objective function was developed through manual experimentation, but again we stress that this is not considered optimal in any way. Specificity may be more important in some domains, while totally unimportant in others, for instance – it depends on the nature of the preference functions that the programmer wishes to generate. In our case, reflexivity was found to be important in ensuring a perception of defensibility in the resulting preference functions. A high weighting for reflexivity might be preferable in many application domains, we will determine this in further development and use of these criteria, and we expect variation to be found in the other metrics as well according to the needs of the individual system.

Because of the nature of code generation, particularly our code generator's implementation, it is possible for a code segment to either fail compilation, or to throw exceptions during evaluation. We catch and ignore any errors in this process and assign a negative fitness to the code segment.

Crossover of two code segments uses one-point crossover on the list of code statements making up the segment. This is currently acceptable for the subspace of the C# specification we cover, although once local variables are introduced, this approach will need revision to avoid constantly introducing scope errors (where a local variable is referenced in the latter half of a function but its declaration was not carried over during crossover). Mutation is applied by randomly regenerating one of the code statements in the list of statements. As with crossover, once the system's focus moves to more complex method constructions, a finer-grained mutation process may be required that is capable of making small changes to individual statements in a method.

In order to speed up the evolutionary system, we compile an entire population of preference functions simultaneously, passing each comparator as a separate file along with the template comparators they inherit from. If errors are thrown during the compilation of a particular comparator, they do not affect the compilation of the other files passed. Testing

[1] The types of the parameters to the function are changed from `int` depending on what type the system is evolving comparators for.

of this method showed it was far more efficient than single-file compilation, even when done in parallel, because most of the overhead of compilation is in initialising and shutting down the compiler itself. This may change in the case of generating extremely large code blocks, but we do not expect it to be an issue in the near future.

## Results

In this section we give several example results from the system, for different domains. We begin with some simple examples for comparing integers, then show two more applied examples: preference functions which decide between colours expressed in RGB format, and preference functions which compare pieces of game content embedded as part of a simple videogame. In each case, we give the code of the preference function, and an English description of the code.

These are hand-selected preference functions, however the curation coefficient – that is, the proportion of the system's output which we would be happy to show to others, as in (Colton and Wiggins 2012) – is extremely high. So far we have not seen any high-fitness comparators (>0.95 fitness) that would not act as justifiable, if simple, preferences in some way. Curation is necessary only to avoid showing the same function twice, because the system frequently generates comparators with identical functionality but very different code, as we do not yet implement a novelty search (Lehman and Stanley 2010).

### Basic Preference Examples

The results shown in Figures 1 and 2 were generated with a population of 20 code segments, a test set of 100 random integers in the range $\{-500, 500\}$ to evaluate the preference functions, and 15 generations of evolution. We found this to be sufficient to evolve high-fitness (0.95 or higher) functions that compared integers.

Figure 1 shows a preference function which prefers negative numbers over positive ones. This is expressed in a rather awkward way: by adding the two arguments together and comparing them with one of the arguments on its own. The else case in this conditional statement returns the opposite ordering instruction (`-1`), meaning that this function has a high consistency while also being precise.

Figure 2 shows a more standard ordering on integers, from smallest to largest. Both this method and Figure 1 have large amounts of unreachable or redundant code. This is expected, given that the system is concerned with the function of code rather than its design. The unnecessary code is not impossible to filter out with the right interpretation of compiler messages, since the C# compiler recognises many of these issues and will present warnings to the system when attempting to compile. We touch on this topic in the discussion section.

In a further experiment, we expanded the expressivity of the code generation to include the `char` primitive type as well as the notion of casting to a type. Evolving high-fitness results for this target domain was more difficult and required a larger evolutionary run than with integer types. We ran populations of 30 code segments, a test set of 100 random `chars` whose ASCII codes fall in the range $\{0, 128\}$ to evaluate the preference functions, and 15 generations of evolution. Figure 3 shows a function which sorts `chars` in reverse lexicographic order. We increased the population size because usable preferences were proving difficult to evolve – as one can see, this is most likely because type casting was required to produce the simplest preference functions, which makes the code much longer and therefore harder to evolve.

```
public int compare(int i, int j) {
    if ((i < i)) {
        return 0;
        return 0;
    }
    if (((j + i) < j)) {
        i = i;
        return -1;
    }
    else {
        j = -491;
        return 1;
    }
    return 0;
}
```

Figure 1: If `i` is negative, it is preferred over `j`; the second conditional check is true if $i < 0$.

```
public int compare(int i, int j) {
    if ((i <= j)) {
        return -1;
        j = ((i * 335) % j);
    }
    else {
        return 1;
        j = j;
    }
    return 1;
    return -1;
}
```

Figure 2: Orders numbers from largest to smallest. The first conditional returns a reverse ordering (-1) if the first argument is smaller than the second. Note the copious amount of unreachable code. This constitutes a compile-time warning in C#, which is suppressed here.

### Object Preferences

Figure 5 shows a preference function evolved for comparing a more complex type – in this case, an object with four fields representing a Monster from a simple game. The class skeleton for the object is shown in Figure 4. These examples were also evolved with a population of size 40, run for 30 generations, with a test set size of 100. Evolving preference functions for objects gives the system a wider state space to explore with more interesting comparisons available to it, with the potential to generate preference functions which compare along two axes simultaneously.

We are building a prototype game, *I Like This Monster* that uses preference generation as part of a process of automated game design. Choosing one particular kind of game

```
public int compare(char i, char j) {
    if ((((int)(j)) <= ((int)(i)))) {
        return 1;
        return -1;
    }
    else {
        i = ((char)(((((int)(i)) -
        ((int)(i))) * (((int)(j)) -
        ((int)(j))))));
        return -1;
    }
    return 0;
    return 0;
}
```

Figure 3: Reverse lexicographic ordering on characters. Note that explicit casts to `int` types has caused a lot of excess bracketing.

element over another has a large component of subjectivity to it, particularly if the game content is already balanced for difficulty and fun. Rather than randomly choosing certain game elements, or choosing them according to a fixed designer preference, the game generates a preference for certain game elements like monsters. This preference is then used to select from a database of pre-generated game content to decide what is included in the game. This is analogous to generating multiple poems and choosing between them as in (Rashel and Manurung 2014) – but unlike random choice, the use of preference functions means that the decision can be *framed* and given a justification. We discuss how one might generate text from preference functions below.

For a more visual example of a preference function, Figure 8 shows another example. In this case, a preference function is generated for an object representing RGB colours, with three `int` fields representing each colour component. A preference function was generated which prefers colours with more red in them. Figure 8 shows the effect this has: the top row shows a randomly generated line of RGB colours, and the bottom row shows the same line ordered from least preferred on the left to most preferred on the right. The preference is very simplistic – it doesn't quite correlate to a visual language of 'redness', but the software can justify its decision on a code level even if it does not directly corresponding to visual processing in people.

In all of the results given in this section, we found that reflexivity is the metric which was maximised quickest. This is likely because it is the simplest to satisfy, as it primarily safeguards against particular bad patterns of code being generated (as long as the function does not return an answer based on the ordering of the arguments, it is always maximised). If the target is high specificity, this is often maximised next, as this requires the preference function simply return nonzero values. However, more complex specificity requirements may require branching and non-constant return statements. In this case, it is much harder to maximise than transitive consistency. These observations largely apply here because the domains we are considering are relatively simple and the preference functions we are generating are low

```
public class Monster{
    public string name;
    public int health;
    public int damage;
    public boolean poisonous;
}
```

Figure 4: A dummy class specification used for generating preference functions. `health` cannot have a negative value, but `damage` can (some monsters heal by attacking).

```
public int compare(Monster i, Monster j){
  i.name = j.name;
  if ((j.health > i.health)){
    i = j;
    return 1;
  }
  else{
    return -1;
    j.name = j.name;
  }
  i.damage = (i.health / i.health);
}
```

Figure 5: An ordering on Monster objects based on their `health` variable.

in complexity and length. We expect this to change in future – functions which compare multiple variables simultaneously, for example, are far more likely to be transitively inconsistent, while functions which return variable values or have high branching are more likely to have lower specificity. This raises the question of how to find these functions over evolving simpler preferences – it may be that additional 'interestingness' metrics are required, or it may simply be that asking for longer preferences or a novelty search powered by agreement will be enough to promote the evolution or more complex preferences.

## Related Work

No work we are aware of directly tackles the problem of generating meaningful, defensible preferences for creative agents across arbitrary domains. However, the idea of

```
public override int compare(RGB i, RGB j){
  if(((j.r * j.r) > (j.r + (i.r - j.r)))){
    j = i;
    return -1;
  }
  else {
     return 1;
  }
}
```

Figure 6: An ordering on RGB objects based on their `r` (red component) variable.

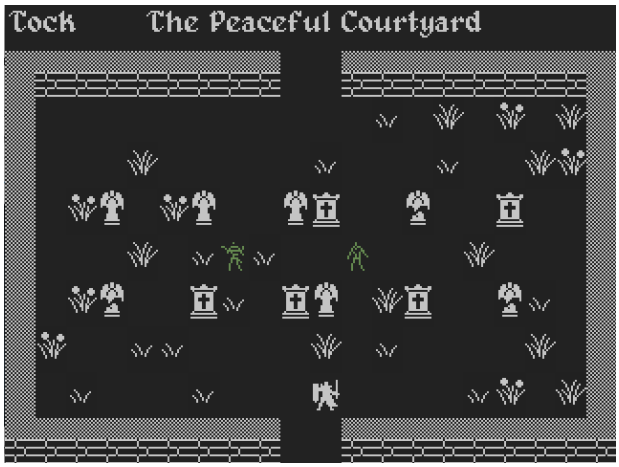



Figure 7: A screenshot from *I Like This Monster*, showing a level where a particular kind of enemy - poisonous creatures - has been selected because of a preference for monsters with the `poisonous` field set to true.

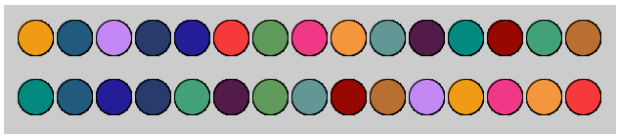


Figure 8: A random colour palette (top row) and an ordering of the same palette according to a preference about RGB colors which prefers colours with more red in them (bottom row, preferred colours towards the right).

computationally representing subjective decisions has some precedent. In (Saunders and Gero 2001) the authors describe a community of creative agents which are designed to have some concept of novelty and interestingness. Each agent possesses a neural network which learns by viewing artworks generated by agents in the community. This can be used to gauge novelty for a given artwork by assessing how much the artwork differentiates itself from the network's current state. Interestingness is based on a Wundt curve calculation in which the most interesting artefacts lie between the extremes of high and low novelty.

(Saunders and Gero 2001) can be seen as a form of preference modelling, in that the agents are armed with a way of making decisions about creative works, if we interpret interestingness to be a subjective quota. Our work is different in a few important ways: it generates a range of preferences based on different factors that vary according to the objects being considered, whereas the community of agents only work in the realm of novelty. The work is also more prescriptive, in our opinion, than the metrics we propose - although we should stress that the authors do not claim to be investigating the generation of varied preferences, the work has other objectives, but we have cited it here as an interesting piece of related work.

Similarly, (Maher, Fisher, and Brady 2013) presents a computational model of surprise, which could be considered to be a form of preference if applied to selection or evaluation (as the authors propose). Similar to (Saunders and Gero 2001), we differentiate ourselves from this work primarily because our aim is to produce a higher-level system which can generate a variety of preferences based on different factors, rather than primarily basing it on surprise or novelty.

The automatic creation of code by software is not a new concept. Code generation, or 'unrolling' of code, is a common concept in software engineering, used for purposes such as optimisation, or the automatic reconfiguration of code in response to dynamically changing execution environments. This is often highly template-based, and the code is generated for precise functional objectives that are normally known well in advance.

Code-generating systems also exist in artificial intelligence. Machine learning software, for example, can be viewed as producing programs as their primary output. Decision trees, neural networks or inductive logic programs can all be seen as forms of computer programs, sometimes (such as the case of ILP or evolutionary programming) quite explicitly. Machine learning techniques have been seen in Computational Creativity on many occasions. For instance, (Morris et al. 2012) uses machine learning as the basis for a computationally creative soup recipe inventor, trained on a corpus of existing soup recipes, and (Colton 2008) uses machine learning in a module within The Painting Fool, a computationally creative artist.

The generation of code is perhaps most explicitly present in Computational Creativity in (Cook et al. 2013), in which we presented Mechanic Miner, a system which explores, modifies and executes the codebase for a simple videogame, in order to discover new concepts for game mechanics and rules. The system was capable of generating single lines of code, modifying the existing game's code to include this new instruction, and then playing the game to evaluate the effect of the generated code on gameplay. In doing this, the system rediscovered several existing game design concepts, previously invented and used by game designers. It was also capable of surprising us as the creators of the system, by presenting solutions which were highly unexpected or took advantage of the system's detailed use of code to perform unexpected operations on the target videogame. This notion of generating directly executable, readable program code in an everyday programming language is one of the motivations for the work we have described here.

## Discussion

The preference functions presented in this paper represent a first step towards a system which can reliably generate interesting preferences for arbitrary targets. We believe it represents a promising new avenue for exploration, and one that could greatly enhance the quality of framing that Computational Creativity systems are able to provide.

Generating code which claims to represent 'preference' is potentially controversial. The reason for many decisions being randomised or guided by hand-designed heuristics in the first place is that software does not hold personal opinions and is not human. We would argue, however, that we are in

the business of perception – recall the definition of Computational Creativity from earlier as being dependent on 'unbiased observers'. Whether we like it or not, our software is judged on how it presents itself, and our first-hand experience of building systems and presenting them to the public has shown us that random decision-making and heuristics inherited from people are as damaging to expectation and perception as any amount of personification.

Furthermore, we would argue that having software express a preference is not necessarily in bad faith. Representing a random decision as having a basis in personhood is deceiving the observer, but with a preference function there is a chain of reasoning, a process that is itself accountable and can be framed, that shows where this preference has originated from. This preference can be interrogated, an observer can present new examples to it to try and better understand what it prefers and why. The software is not claiming to have an emotional basis for this – it is simply stating a preference that it used to guide its decision-making process. Of course this does not offer a perfect solution to all the problems of subjective decision-making in creative software, but we believe it offers a new way of exploring the issue.

It is worth noting that these preferences are, in some ways, equivalent to random choice. They are arbitrary, domain-agnostic, they do not care about their impact on the viewer (unless we used the agreement to be contrarian, perhaps). We are not claiming here that these preferences provide a benefit to the code over randomly choosing something, nor do we even claim that it makes the system more creative in terms of its functionality. We do believe, however, that they provide a benefit to the *perception* of the software as creative if its decisions can be justified, if we can claim that no random number generation is involved, and if its decision-making process can be inspected and interrogated by observers.

There are several important areas of future work to be undertaken in order for the system described in this paper to be able to work in large computationally creative systems. Some of these topics have already appeared earlier in this paper. Firstly, the system should be expanded with a larger state space to explore in terms of code generation, so that more complex functions can be generated. This may be possible with existing techniques simply by applying it at a larger scale, however the state space explosion is significant once more complex programming features – like method invocation – are taken into account. It may be that evolution is not the best approach for generating code at this scale, or that the process requires alteration in order to be more efficient for this kind of optimisation problem.

A second point of future work is automatic simplification of generated preference functions. This is an achievable goal, and many optimisation processes for program compilation already do this. We mention it here because it is particularly important for code generation in the context of Computational Creativity, as we explained in (Cook et al. 2013). Compressing a piece of code by removing unreachable or non-functional code makes it easier to understand, easier to compare, and also has the important side effect of making it easier to explain, which is a third future work task.

Being able to explain the function of a piece of code is crucial to this work – in the examples we gave in the Results section there was a lack of textual framing to the visual examples. In some senses it is possible to interpret the effect of the preferences simply by looking at the content produced, but in general it is desirable to be able to have the system itself express 'I prefer redder colors'. Producing English renderings of the function of code is complex – we are currently exploring possibilities which use some metadata tagging on the code prior to generating preferences, but there are many more and better approaches yet to be discovered.

Finally, representing preference functions in a higher-level mathematical language may be advantageous for this work. Many of the problems we have encountered are direct consequences of the code-based representation, such as the presence of unreachable code and functionally identical generation. We hope to look into more abstract representation formats for future versions of our system.

## Conclusions

In this paper we introduced a series of criteria for assessing functions that describe preferences, motivated by a desire to provide non-random justifications for small creative decisions that don't rely on other people. We showed how an evolutionary system can use these criteria as the basis for a fitness function that evolves code which act as preference functions. We gave examples of preference functions we evolved using these criteria for comparing various types, including videogame content and colours, and discussed the issues it raises for Computational Creativity, in terms of the code itself and the nature of generated preferences.

The perception of creativity in software is a defining problem for our field. We hope that the work we have described here offers a new avenue to explore for framing decisions made by the software we build. Even the smallest of decisions are affected by people's perceptions of software as arbitrarily random, or clones of their designers. We believe that the future of decision-making in software lies beyond random choice and modelling human opinion – we need to give our software independence and remove the influence of other people on it. We acknowledge that we have by no means managed to remove ourselves from the process of decision-making – we have designed the system which produces preference functions, defined its metrics and provided it a fitness function. But we hope that we have offered a way to take one step further into the background, leaving our software to stand alone at the fore.

## Acknowledgments

The authors would like to thank Adam Smith and Rob Saunders for discussions about code generation, and Alison Pease for input on an earlier version of this paper. Thanks to the reviewers for very thorough reviews which helped improve this paper. This paper is a difficult balance of position paper and systems description, which made it hard to meet all of your comments, but we have taken them on board for our future work and writing, and we appreciate them greatly. This work was funded by EPSRC grant EP/L00206X.

## References


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*, 77–81.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI 2012 - 20th European Conference on Artificial Intelligence.*, 21–26.

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.; 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. Experiments in constraint-based automated scene generation. In *Proceedings of the International Joint Workshop on Computational Creativity 2008*.

Colton, S. 2009. Seven catchy phrases for computational creativity research. In *Computational Creativity : An Interdisciplinary Approach*, Schloss Dagstuhl Seminar Series.

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*.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of 16th European Conference on the Applications of Evolutionary Computation*.

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.

Lehman, J., and Stanley, K. O. 2010. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* 19(2):189–223.

Maher, M. L.; Fisher, D.; and Brady, K. 2013. Computational models of surprise as a mechanism for evaluating creative design,. In *Proceedings of the Fourth International Conference on Computational Creativity*, 147–151.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an articial chef. In *Proceedings of the Third International Conference on Computational Creativity*.

Rashel, F., and Manurung, R. 2014. Pemuisi: a constraint satisfaction-based generator of topical indonesian poetry. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Saunders, R., and Gero, J. S. 2001. 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.

Tomašič, P.; Žnidaršič, M.; and Papa, G. 2014. Implementation of a slogan generator. In *Proceedings of the Fifth International Conference on Computational Creativity*, 340 – 343.

Veale, T. 2013. Once more, with feeling! using creative affective metaphors to express information needs. In *Proceedings of the Fourth International Conference on Computational Creativity*.

# Attributing Creative Agency: Are we doing it right?

**Oliver Bown**
Design Lab
University of Sydney
NSW, 2006, Australia
oliver.bown@sydney.edu.au

**Abstract**

When contemplating the creativity of computational systems, a host of factors have been taken into consideration, many of which people have attempted to measure or otherwise operationalise: novelty, value, P-creativity versus H-creativity, exploration versus transformation, the subjective evaluation and contextualisation of the artefact, and so on. Whilst of equal importance, the systematic and rigorous attribution of creative agency to different actors in the production of a specific output has been given less attention. It is common to make the simplifying assumption that the most direct contributor to an artefact is that artefact's sole author, but arguably this is never the case: all human creativity occurs in the context of networks of mutual influence, including a cumulative pool of knowledge.

This paper looks at how we might better formalise creative authorship such that for any artefact, a set of agents could be precisely attributed with their relative contributions to the existence of that entity. It asks only what the nature of this formalisation should be, and concludes that a more critical approach is needed to the creative agency of human actors, and thus the expected creative agency of machines.

I draw on two critical notions that can inform a methodology for the ascription of creative origins in computational creativity: becoming, and the agency of networks of interaction.

I look at a example from both historical human creativity and computation creativity, to consider how we can break down creative agency and ascribe it to different sources. Practical implications are dicsussed.

## Introduction

In contemplating creativity, we are comfortable with taking at face value statements such as "Ludwig van Beethoven composed Beethoven's Fifth Symphony" or "Leonardo da Vinci painted the Mona Lisa". At the same time, we are well aware that such attributions are rough at the edges when scrutinised. Creativity does not occur in a vacuum. All creators are subject to influence from their culture or environment, and other forces at play in the creative process include chance, the influencing of opinions such as value attribution, the emergence of outcomes through collective action, and the need to consider the potentially active role played by passive objects, as discussed most famously by Latour (1996) and Clark (2003), but with recently renewed interest by Malafouris (2007), Miller (2010) and Ingold (2007).

Longstanding theories of creativity have successfully managed these apparently conflicting perspectives, most notably the work of Simonton (2003) and Csikszentmihalyi (1999). In both cases, creativity is properly understood as a process that operates at a macro level (sometimes described as a network or systems level). For Csikszentmihalyi the macro perspective is critical because the process of creativity involves the interaction between heterogenous groups of participants, and for Simonton it is because creativity is best modelled as a stochastic process across a population, which cannot be properly understood when looking at single instances.

However, it has been difficult to translate such knowledge into practical methods for evaluation in computational creativity, which despite its strong acknowledgement of such theories does not successfully draw on this macro-level perspective in evaluating individual systems. In this paper I present this challenge in terms of recurring misconception that evaluation can be performed on isolated individuals, i.e., at a micro-level, which I refer to as the "islands of creativity" view. Drawing on literature from creativity research, philosophy and the social sciences, I consider how a macro-level view of creativity can work in the applied task evaluating computationally creative systems.

I suggest that a critical step is to recognise how the objects of evaluation are dynamic, in flux, and have boundaries that shift at different stages in their history, as they interact with other people and things.

I propose a "dynamic analysis" of any system, which details (i) the fluid and temporary boundaries between entities, and when these aggregations act as agents, (ii) when and where influence occurs, (iii) what constitutes an output. Such an analysis, it is proposed, could help us better attribute creative agency in the evaluation of computationally creative systems, by clarifying how novelty and value are determined (by whom) and what influences feed into the creative system at different times.

## Simonton's macro-creativity model

Simonton, for example (Simonton, 2003), showed through quantitative analysis of scientific achievements that the

arrival of creative breakthroughs was sufficiently unpredictable as to be effectively random. This does not mean to say that a member of the population chosen at random might make an advance in quantum physics. Naturally, strategies for creative success involve becoming expert in a field, focusing on problems, working hard, knowing what to look for, and so on. Indeed, Simonton showed that success was proportional to activity: the more active you were in a field the more likely you were to produce creative outcomes, but equally the more likely you were to produce uncreative ones. What remained stable was the rate of success, measured as the ratio between successful output and total output.

From this perspective, in computational creativity what we might describe as strictly micro level focus – privileging the creative agency of individual creators without considering how these agents interact with each other and with other elements in the world – is a detrimental but seductive simplification, which is often assumed to be reasonable where in fact it is problematic. Simonton's micro-level view of creativity tallies with his macro-level view: at the micro-level an individual iterates through many trial-and-error attempts at a solution, understood in creativity research through cognitive processes such as incubation. This trial and error is the best that can be done in an unknown search space; there aren't reliable analytical or inductive approaches available to the kinds of problems that we would define as creative, because the problem spaces are unknown – at least, in the case of Boden's 'transformational' creativity (Boden, 1990). Thus we may imagine a population of individuals searching for solutions to the same problem, working at the same rate. When one individual discovers the solution, in Simonton's view, we should not leap to the conclusion that there is anything fundamentally different about the creative process used by that individual. Simonton also draws on evidence from 'simultaneous scientific discoveries' to support this view, arguing that the common occurrence of such discoveries is due to the fact that it is the discovery context, and not the creative ability of the discoverer, that is key to the arrival of the discovery.

Such work is widely acknowledge in computational creativity research, but this macro-level thinking remains largely absent in the methods that we apply to the evaluation of computationally creative systems.

## The "islands of creativity" problem

Such approaches have been successfully applied in the context of studies of traditional human (i.e., not computational) creativity. But in computational creativity, although we frequently pay homage to these macro theories, we have yet to find a way to incorporate them into a working methodology in the complex area of evaluation. I suggest that a significant obstacle to computational creativity evaluation lies in the idea of "islands of creativity", the idea that creativity is situated in specific systems (mostly humans, now also computers), without any fluidity between these systems and the rest of the world:

**Definition**: *The "islands of creativity" problem in creativity is the misuse of the simplifying view that individual human actors (or individual computer actors) are sole originators of specific creative artefacts. It conflicts with the more holistic view that stochastic and network macro processes involving interactions between heterogeneous elements underlie the big picture of creative production.*

Is this view actually a misconception, and what have been the implications of holding it? Would our approach to evaluation benefit from avoiding it, and shifting towards thinking about creation occurring through the relationships between entities? I will argue that looking at creativity only by reference to the human cognitive capacity for creativity continues to be problematic for computational creativity, not least because the kinds of computational systems that will do creative things in the near future may not do them in particularly human-like ways. Rejecting the "islands of creativity" problem is a necessary part of stepping away from a human-centric frame.

Specifically, the embrace of an alternative, macro-level theoretical framework may enable two important contributions to computational creativity: (i) in the way we understand what we mean by human cultural activities such as art and music. There is a tendency to trivialise such questions in pursuit of simple computable targets, whereas these areas of activity are some of the most ethnographically rich that humans exhibit, so as to be far from easily reducible; and (ii) by providing practical methods to help us attribute creative agency properly when asking questions of the form "did system x do something creative?"

## Defining creative production in terms of interaction

In the words of Heraclitus, via Nietzsche, "the whole flows as a river", the river's evident dynamism, by which it is constantly in a state of re-creation in the movement of water, is an apt way to understand those less obviously fluid things in our environment: "being is an empty fiction" (Nietzsche, 1998). We tend to take the consistency of objects at face value, but for practical, not only philosophical, reasons it can be preferable to view things not as entities that have the property of *being*; instead their existence is in constant re-creation, captured through the notion of 'becoming'. Viewing things without this frame of dynamism, as neat bounded entities, may be a practical way of simplifying and understanding the world in the everyday, but risks missing the myriad ways in which entities transform, influence each other, have porous boundaries and fuse and fissure. Such thinking has been applied successfully in the social sciences, and may be helpful in thinking about evaluation in computational creativity, particularly in how we frame the notion of a creative agent.

### Creative agents

Theorists have embraced the idea of fluidity in the context of social systems, which are more evidently fluid, using a network interaction approach, most famously the actor network approach of Latour (1996) and Law (1992) and the extended mind theory of Clark (2003). More recently, Malafouris (2007) makes a terse argument for the abandonment of the human as a privileged category of agency. For Malafouris,

much as for Clark, if a blind man can be said to 'see' with his stick, then the physical matter of the stick is exactly to the blind man what the optic nerve is to the sighted. For as long as the blind man is using the stick, we can designate a transient entity of the form blind-man+stick which is in some sense capable of sight. Importantly, the stick is part of that unit, not apart form it. The man does not see with the stick; the man+stick sees.

Similarly, he argues, as a potter shapes clay on a wheel, one cannot successfully draw neat lines of causality that show the potter's hands influencing the clay, and not vice versa. The potter is responsive to the clay, and in her adaptivity, allows causality to flow back in the opposite direction from clay to action. The right way to understand the resulting creation of a pot, Malafouris posits, must not presume potter as agent and wheel and clay as other, but to conceive of a unity in interaction between them.

In his words:

> "If human agency *is* then material agency *is*, there is no way that human and material agency can be disentangled. Or else, *while agency and intentionality may not be properties of things, they are not properties of humans either: they are the properties of material engagement, that is, of the grey zone where brain, body and culture conflate*." (original emphasis). (Malafouris, 2007, p. 22).

The purpose of this thought experiment is to preempt and thus interrogate the implied objection: "surely we can see that the potter is the active, intelligent agent in this interaction, whilst the wheel and clay are passive non-agents, there to be operated or shaped". This presents a problem: although it seems mistaken to start to talk of the agency or intentionality of clay and mechanical wheels, how else can we handle the fact that the resulting pot owes its form to clay and wheels, and not merely to a single human actor?

I understand Malafouris as saying here, as with the blind-man and his stick, that it would be more correct to say that the temporary interaction of potter+wheel+clay is responsible for the creation of the pot, than to say that the potter created the pot using the wheel and the clay. Although apparently a trivial distinction, the question of agency has been shifted in a way that significantly transforms discussions of creative authorship in computational creativity, and equally resolves the "islands of creativity" problem. This is a more palatable option than talking about the agency of inanimate objects, and is particularly apt in the context of machines, for which the perception of agency might slide easily up and down a scale. It also takes care of collaborative action between individuals, whether in a clearly bounded working unit such as a band, or a fluid genre movement.

Turning to computational creativity, we see that attention to this detail concerning the existence of bounded agents is generally overlooked. In major mathematical and logical formulations such as those of Ritchie (2007) and Wiggins (2006), understandably, this would be a complex step. Here the focus is more on artefacts anyway. In other work where the focus is on the individual and the process of production, there is still little in terms of acknowledging the fluid boundaries between components of a creative system.

## Dividing individuals

Further to this, thinking from philosophy of mind, AI, evolutionary psychology, anthropology, and other disciplines, has in different ways converged on a notion that human agents, equally, should not be viewed as unitary in action, but consist of networks of interaction themselves. This thinking can be found in Minsky's society of mind (Minsky, 1988), Baars' global workspace theory (Baars, 2005), Barkow, Cosmides and Tooby's (Barkow, Cosmides, and Tooby, 1992) multi-domain model of the evolved mind, and many psychological accounts that reveal conflicting drives and processes and dedicated channels of activity. In anthropological theory we have the notion of the 'dividual' (Marriott, 1976). This concept was initially specific to an ethnographic analysis of how South Asians viewed personhood, but it may also describe Western conceptions if we admit them to have more variability:

> "Single actors are not thought in South Asia to be 'individual', that is, indivisible, bounded units, as they are in much of Western social and psychological theory, as well as in common sense. Instead, it appears that persons are generally thought by South Asians to be 'dividual' or divisible. To exist, dividual persons absorb heterogeneous material influences. They must also give out from themselves particles of their own coded substances, essences, residues, or other active influences that may then reproduce in others something of the nature of the persons in whom they have originated ... What goes on between actors are the same connected processes of mixing and separation that go on within actors."

(Marriott, 1976, p. 111)

Although framed in terms of a distinction between Indian and Western perspectives, it is fair to say that in all world views there is some freedom to flip between different conceptions of personhood and individuality. It is common to talk about feeling like you are 'defined' by your family or friends or the objects you possess. We are also familiar with the idea expressed at the end of the quote, that two people can 'think together', for example through brainstorming, and that this is in some way isomorphic to the same process happening within an individual.

In our computationally creative systems, this fluidity is more evident. A piece of software is itself an assemblage of subsystems and may communicate beyond its nominal boundaries to form supersystems, including with humans. We should expect that in some cases it is clear that agency is more strongly associated with a specific subsystem than with others, whereas in other cases, agency takes the form of interaction between subsystems or the system and its environment.

## An evolutionary framework

As others have discussed (Dawkins and Krebs, 1978; Boyd and Richerson, 1985; Aunger, 2000; Shennan, 2002), Darwinian evolutionary theory provides a good template, recog-

nising in natural evolution exactly that agency lies in 'processes of interaction' rather than in specific entities (Nietzsche was also heavily inspired by Darwin). It is interesting to contemplate the non-human creativity of evolution in contrast to what we typically think of when considering human creativity. Given a specific organism and asking, "what created that organism?" we see very clearly that such an act of creation can only be understood as a continuous process of interaction between organisms and their environment, and amongst individual organisms. We cannot pin our form on the creativity of our parents, nor even on our entire ancestral history. This view naturally takes into account the the many interesting cases of coevolution, runaway sexual selection, niche construction and, in humans, gene-culture coevolution which produce things through diverse forms of interaction.

When we talk of function in such systems we are actually referring to teleofunctions (Sperber, 2007), specifically, functions that serve their own existence. This is in contrast to the functions of things we build, which are imposed upon them and are external to the existence of the thing. But cultural traits and artefacts can and often do have teleofunctions too and can come about in ways that are more or less similar to evolutionary processes occurring at a cultural level. Sperber (Sperber, 2007) discusses the interesting case of the perception of suntanning. Furthermore, machines that learn or evolve can have teleofunctions by virtue of the fact that their goals can be adaptive, but mostly, today, are built with regular functions.

## Dynamic analysis of fluidity in creative systems

Our earliest efforts at building machines that create have resulted in superlatively weak creative agents when held up against human beings, as would be expected. But the contemporary language of creativity is geared towards the superlative creativity of humans. It does not do well at describing the simple forms of computational creativity we are developing today. For this reason, an "islands of creativity" view, that works for humans, needs to be replaced by a more fluid conception of creativity that will work equally well for computational systems. By comparison, a view of this process of production based on networks of interaction between elements (whether brain, body and culture, as Malafouris suggests (Malafouris, 2007), or some other active ingredients) makes less of a conceptual meal of that scenario.

Even if these various perspectives may be technically true, is it any use to try to use them to rethink evaluation in computational creativity? It would be counterproductive to take clearly delineated elements and blur them into a loosely defined muddle of interaction purely for the sake of being more accurate. A danger with adopting this perspective is that useable categories disappear to dust. Evoking a Beethoven-piano-stave-pen-church-king-orchestra-etc.-etc. network complex to explain the creation of the Fifth Symphony may not have any practical value and if so, should not be pursued. But as part of a wider investigation into how qualitative, situated human science methods can contribute to the understanding of evaluation in computational creativity (Bown, 2014, 2012; McCormack et al., 2014), I believe that it will be necessary to take on the "islands of creativity" problem by introducing such thinking to form a method of "dynamic analysis" of creative systems.

As a first step in a dynamic analysis approach, we would need to look at where we have pre-emptively identified creative agents. Mostly, these will be either individual people, or the computational systems we have built. For each presumed agent, we should investigate what assumptions we hold about their boundedness, their autonomy (any cases in which we say the system did something "on its own") and the origins or their actions. We can also investigate where different systems might be seen to unite in co-action or break down into interacting components, and we can look at how each system is influenced to change its state or structure over time. In each case, this will be a temporal process where different system boundaries are recognised over time. In the case of many computationally creative systems, the full analysis of such a process would include the role of the system developer, observing outcomes and iterating their design in order to improve it (what Colton, Pease, and Ritchie (2001) refer to as "fine tuning"). We may also find that the process is so widely distributed across elements that such descriptions take on a more statistical nature, as we have seen in both Simonton's theories (Simonton, 2003), and in Darwinian evolutionary thinking. In this case, it should be fine to attribute some degree of creativity to a macro-level stochastic process itself.

Through the examples below it is proposed that a simple but effective way to dynamically analyse creative events is through simple dot-point timelines that discuss sequences of events, the influence of systems on each other, and the potential coupling of systems. This is relatively crude, but may have the potential to feed ultimately into more formal frameworks such as that of Wiggins (2006).

## Application

Without adopting a strong cultural Darwinism – which is contrary to what I would argue for, and what Sperber's article (Sperber, 2007) emphatically argues against – it follows from all of the above that every creative act should be framed in terms of processes of interaction. The issue still remains of showing that this is practically useful. I consider the following instances and how such an approach serves to clarify the creative agency.

### The Violin

In a recent article (Nia et al., 2015) evidence was given to support the theory that the shape of sound holes in violins emerged through an essentially evolutionary process whereby apprentices copied their masters' designs with random variation, and those designs with louder sounds, due to the shape of the holes on the body of the violin, were over time more successful. The winning design, the familiar f-shape that we know today, maximises the ratio between the perimeter of the hole and its size, providing greater amplification of the sound, whilst providing a pleasing visual appearance. Who designed the violin as we know it today? If the above account is correct we could answer as we would

with the design of organisms in the biological world, that there is no one designer, and there are not really any designers in the sense of psychological creative discovery. The design came about through a macro-level process. Indeed, we could go so far as to say that the design of the optimised sound holes was not due in any way to a human creative capacity, although, difficulty arises when we ask whether a given luthier's new design was actually a conscious improvement, or a random variation that turned out to be successful. As with Simonton (2003), we may be mistaken in attributing creativity to the individual mind instead of to the broader cultural process.

The creative process, as described by Nia et al. (2015), might look something like the following if represented as a dynamic analysis timeline:

1. An existing design is copied and modified in ways that do not explicitly attempt to optimise sound amplification
2. Given time, the louder designs make more money, and these workshops grow and reproduce whilst the workshops responsible for the quieter designs diminish.

### Paul Hession / Arne Eigenfeldt Live at Cafe Oto

At a recent concert of live algorithms[1], drummer Paul Hession and flautist/saxophonist Finn Peters performed with a number of live algorithms. I consider the performance between Paul Hession and Arne Eigenfeldt's (Eigenfeldt, 2014) system[2] (a discussion of the factors underlying such concerts can be found in Bown et al. (2013)). Clearly, as an improvised duet, the interaction between the two participants is critical to understanding the creative output. Musical improvisation is possibly the most unambiguous case of a process of interaction underlying a creative result. But over a longer timescale we can consider Eigenfeldt's development of the system, and his interaction with Hession during rehearsal as part of the creative process. It has been proposed in various ways (e.g., McLean and Wiggins, 2010), that creative software development involves a cycle of interaction between developer and software, and we can see this as directly analogous to the case of the potter described by Malafouris, with the same arguments applying. Such notions have also been discussed in the case of Cohen's work with AARON (McCorduck, 1990).

A full picture of the development of the outcome might look something like the following. Through discussions with many live algorithm developers, this seems typical, and really it is just a specific case of what any musicians do in preparing for a collaborative performance:

1. Designer takes on project, listens to recordings of Musician in order to approach design of System;
2. Designer iteratively develops System;
3. Designer, System and Musician rehearse;
4. System and Musician perform.

[1]Cafe Oto, London, June 29th 2014, as part of the New Interfaces for Musical Expression 2014 Conference.

[2]https://www.youtube.com/watch?v=vL6Jty5hOFc

In this we can look at the moments where there is influence. Of interest, in Stage 1, the musician has influence on the System. In Stage 2, the system has influence on the designer, and in Stage 3 the System has influence on the musician, influencing how they might choose to perform. Under Malafouris' framework, these interactions, no matter how consciously or authoritatively the subject of the influence is receiving this input, imply that boundaries between these entities are fluid, or porous. We should be aware that that design of the system contains iterative, hence albeit minutely autopoietic, development, and the final form of both system and musician are the result of a longer-term interaction.

Still, does this matter? It is not burningly evident that it does. But it provides a more complete analysis than if we say that a system, all of a sudden, stands alone as an autonomous agent and 'produces' things. A rich qualitative description takes account of the actual pathways that lead to something being produced.

## Conclusion

In this paper I consider what is still, despite its long standing in social sciences, quite a radical approach to thinking about attributing creative agency. This view removes the privilege of the human actor, making place for the idea of humans and other actors forming temporary networks of interaction that produce things. It does not unfortunately offer us a powerful analytical framework that makes agency attribution easy or formulaic, but asks us to avoid making mistaken and simple agency attributions, whether to humans or to creative machines.

## References


Aunger, R. 2000. *Darwinizing Culture*. New York: Oxford University Press.

Baars, B. J. 2005. Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. *Progress in brain research* 150:45–53.

Barkow, J. H.; Cosmides, L.; and Tooby, J. 1992. *The Adapted Mind: Evolutionary Psychology and the Generation of Culture*. New York: OUP.

Boden, M. 1990. *The Creative Mind*. George Weidenfeld and Nicholson Ltd.

Bown, O.; Eigenfeldt, A.; Martin, A.; Carey, B.; and Pasquier, P. 2013. The musical metacreation weekend: challenges arising from the live presentation of musically metacreative systems. In *Proceedings of the musical metacreation workshop, AIIDE conference, Boston*.

Bown, O. 2012. Generative and adaptive creativity. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Berlin: Springer. 361–381.

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.

Clark, A. 2003. *Natural-Born Cyborgs: Minds, Technologies, and the Future of Human Intelligence*. Oxford University Press.

Colton, S.; Pease, A.; and Ritchie, G. 2001. The effect of input knowledge on creativity. *Technical Reports of the Navy Center for Applied Research in Artificial Intelligence*.

Csikszentmihalyi, M. 1999. Implications of a systems perspective for the study of creativity. In Sternberg, R. J., ed., *The Handbook of Creativity*. New York: Cambridge University Press. 313–335.

Dawkins, R., and Krebs, J. R. 1978. Animal signals: Information or manipulation? In Krebs, J. R., and Davies, N. B., eds., *Behavioural Ecology: An Evolutionary Approach*. Sinauer Associates. 282–309.

Eigenfeldt, A. 2014. Generating structure–towards large-scale formal generation. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Ingold, T. 2007. Materials against materiality. *Archaeological dialogues* 14(01):1–16.

Latour, B. 1996. On actor-network theory: a few clarifications. *Soziale welt* 369–381.

Law, J. 1992. Notes on the theory of the actor network: Ordering, strategy and heterogeneity. Published online at www.lancs.ac.uk/fass/sociology/papers/law-notes-on-ant.pdf.

Malafouris, L. 2007. At the potter's wheel: An argument for material agency.

Marriott, M. 1976. *Hindu transactions: Diversity without dualism*. University of Chicago, Committee on Southern Asian Studies.

McCorduck, P. 1990. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. W. H. Freeman and Co.

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.

McLean, A., and Wiggins, G. A. 2010. Bricolage programming in the creative arts. In *22nd Annual Psychology of Programming Interest Group*.

Miller, D. 2010. *Stuff*. Polity.

Minsky, M. 1988. *Society of mind*. Simon and Schuster.

Nia, H. T.; Jain, A. D.; Liu, Y.; Alam, M.-R.; Barnas, R.; and Makris, N. C. 2015. The evolution of air resonance power efficiency in the violin and its ancestors. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 471(2175).

Nietzsche, F. 1998. *Twilight of the Idols*. Oxford University Press.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Shennan, S. 2002. *Genes, Memes and Human History*. Thames and Hudson, London.

Simonton, D. K. 2003. Scientific creativity as constrained stochastic behavior: the integration of product, person, and process perspectives. *Psychological bulletin* 129(4):475.

Sperber, D. 2007. Seedless grapes: Nature and culture. In Margolis, E., and Laurence, S., eds., *Creations of the Mind: Theories of Artefacts and Their Representation*. Oxford University Press. chapter 7.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

# Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests

**David A. Joyner[1,2], Darren Bedwell[1], Chris Graham[1], Warren Lemmon[1], Oscar Martinez[1], Ashok K. Goel[1]**

Design & Intelligence Laboratory[1]
Georgia Institute of Technology
Atlanta, GA 30332 USA

Udacity[2]
2465 Latham Street
Mountain View, CA 94040 USA

{djoyner3, dbedwell3, cgraham36, wlemmon3, omartinez8}@gatech.edu; goel@cc.gatech.edu

**Abstract**

The Raven's Progressive Matrices (RPM) test is a commonly used test of intelligence. The literature suggests a variety of problem-solving methods for addressing RPM problems. For a graduate-level artificial intelligence class in Fall 2014, we asked students to develop intelligent agents that could address 123 RPM-inspired problems, essentially crowdsourcing RPM problem solving. The students in the class submitted 224 agents that used a wide variety of problem-solving methods. In this paper, we first report on the aggregate results of those 224 agents on the 123 problems, then focus specifically on four of the most creative, novel, and effective agents in the class. We find that the four agents, using four very different problem-solving methods, were all able to achieve significant success. This suggests the RPM test may be amenable to a wider range of problem-solving methods than previously reported. It also suggests that human computation might be an effective strategy for collecting a wide variety of methods for creative tasks.

## Introduction

The Raven's Progressive Matrices (RPM) tests are a group of intelligence tests based on visual analogy problems (Raven, Raven, & Court 1998). In these problems, a matrix of visual frames is presented with a blank space; six or eight options are presented for filling in this space. Performance on RPM has been shown to correlate well with other intelligence tests (Snow, Kyllonen, & Marshalek 1984). Thus, although wholly visual, the RPM tests measure *general* human intelligence, and are often used as the psychometric measure of choice in educational and clinical settings.

Hunt (1974) suggested that humans use multiple problem-solving methods to address RPM problems, including "analytical" and "Gestalt" methods. Bringsjord & Schimanski (2003) have proposed intelligence tests such as RPM as a method of measuring the effectiveness of AI techniques. AI research has developed a variety of methods for addressing RPM and similar visual analogy problems, including both "analytical" methods that typically use propositional representations (Evans 1968; Lovett, Forbus, & Usher 2009; O'Donoghue, Bohan & Keane 2006; Prade & Richard 2011; Ragni & Neubert 2014), and "Gestalt" methods that often use imagistic representations (Dastani, Induskhya & Scha 2003; Kunda, McGreggor, & Goel 2013; McGreggor & Goel 2014; Schewring et al. 2009). Another way of classifying the various methods is by control of processing. For example, some methods for addressing RPM problems, such as the affine method (Kunda, McGreggor & Goel 2013), first generate an answer based on the (partial) matrix, and test this answer by comparing it with each available choice; other methods, such as the fractal method (McGreggor, Kunda & Goel 2014), test each available answer by computing the degree of fit in the matrix. While it may appear that generation of answers is a necessary part of creativity, we posit that generating explanations for available answers is also creative.

## The Raven's Test and Creativity

One major component in the value of the RPM test is its connection not only to intelligence, but also to creativity. Hunt (1974) laid the foundation for the creative nature of problem-solving methods on this test in identifying the two broad categories of methods mentioned previously, "Gestalt" and "analytical". Kirby & Lawson (1983) argued further that it is the diversity of problem-solving methods that makes the RPM test a valuable tool for assessing intelligence in humans. If creativity is in part the ability to develop novel, useful, and effective methods to a problem, then the RPM test's admission of multiple methods adds to its value as a tool for studying creative problem solving.

Second, Keating & Bobbitt (1998) argue that addressing many RPM problems requires metacognitive abilities to select among the available problem-solving methods, to monitor the progress of the selected method, to suspend or abandon the current method and move to a different method, and to combine insights from the use of multiple meth-

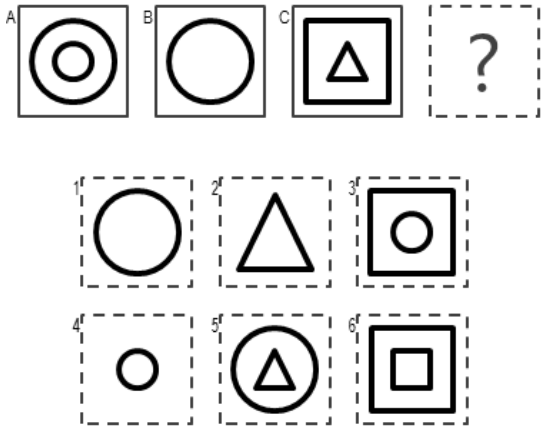


Figure 1: A 2x1 visual analogy problem. Although RPM tests do not have 2x1 problems, 20 2x1 problems are used as a soft introduction to solving visual analogies.

ods into one final answer choice. Third, the normatively correct choices for some RPM problems are often non-obvious, sometimes even unexpected, such as in the problem shown in Figure 1. Thus, from the perspective of both process (metacognitive processing) and product (unexpectedness of the answer), the RPM test measures not only intelligence, but also creativity.

One potential critique of the RPM test for studying creativity is that a set of answer choices are presented to the test-taker. However, this implies that the creative task necessarily entails generating a novel answer. The structure of the RPM problems turns this notion of creativity around: rather than generating an answer, the test-taker instead creatively generates an explanation for a particular answer choice. In Figure 1, for example, the most obvious answer would be a large square; however, none of the answer choices match this obvious answer. The presence of answer choices constrains the activity and forces the test-taker to creatively generate not an answer, but an explanation for why one of the presented choices is most compelling. This explanation is as much the output of the creativity process as the answer itself.

From the perspective of computational creativity, the above analysis makes the RPM test an excellent choice for designing, evaluating, and comparing new AI methods not only for intelligence, but also for creativity: the task admits a wide variety of AI methods characterized by different knowledge representations and different controls of processing. The question then becomes: how can we identify the novel techniques that may effectively address RPM problems?

We postulate that one strategy for acquiring new methods for addressing visual analogy problems on the RPM test is through crowdsourcing (Howe 2008), or, more accurately, human computation (Law & von Ahn 2011). Although crowdsourcing has typically been used for acquiring domain knowledge, human computation also admits acquisition of problem-solving methods. Yet, it is also important to acquire new methods for addressing visual analogy problems not from any crowd, but from intelligent, educated, high-achieving humans who themselves are likely to do well on the RPM test.

## The Experiment

In Fall 2014, we offered a new online Georgia Tech graduate-level CS 7637 course titled "CS 7637 Knowledge-Based AI: Cognitive Systems" as part of the new Georgia Tech Online MS in CS Program (Goel & Joyner 2014; Goel & Joyner 2015). We also offered an in-person class in parallel, with the two classes sharing the same syllabus and structure. The course describes its learning goals as, "to develop an understanding of (1) the basic architectures, representations and techniques for building knowledge-based AI agents, and (2) issues and methods of knowledge-based AI." Toward this end, students cover several knowledge representations (semantic networks, frames, scripts, formal logic), reasoning strategies (case-based reasoning, rule-based reasoning, model-base d reasoning), and target domains (computational creativity, design, metacognition). More comprehensive information on the structure and content of the class is available at the link above.

In previous offerings of the in-person class, we had used variants of problems on the RPM test to motivate the class projects (Goel, Kunda, Joyner, & Vattam 2013). Thus, we knew class projects based on the RPM test stimulated student engagement while providing an authentic opportunity to explore cutting-edge research. Therefore, in Fall of 2014, we again designed the class projects based on variants of problems on the RPM test. Students in both the online and in-person sections were asked to complete four projects that addressed 123 RPM-inspired problems in all, culminating in Project 4, wherein students designed agents that could answer all 123 problems using visual input. 224 students completed Project 4, addressing all the problems using the raw imagistic input. We collected all the data on these 224 Project 4 submissions, including the designs of the agents and their performance on the 123 problems.

In this paper, we will describe the results of this experiment. First, we will present at a high level the results of the 224 agents that were developed to address these RPM-inspired visual analogy problems. Second, we will examine in greater detail the design of four of the most creative and effective agents developed for the project. These agents operate according to four significantly different methods for reasoning about these problems. In describing these agents, we will clarify their relationship to elements of human creativity operationalized and instantiated in AI agents.

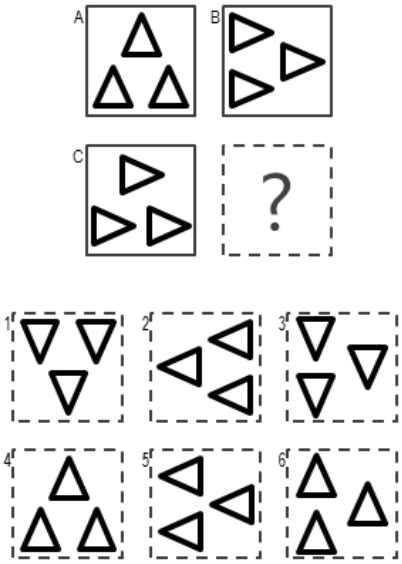



Figure 2: A 2x2 visual analogy problem, inspired by Raven's Progressive Matrices. In this paper, individual squares in a problem are called 'frames', while individual shapes within each frame are called 'objects'.

## RPM-Inspired Visual Analogy Problems

The standard set of Raven's Progressive Matrices test is made of 60 visual analogy problems: 24 of the problems are 2x2 matrices, and 36 of the problems are 3x3 matrices. For copyright reasons, we have not yet been able to use actual RPM in these class projects. Instead, we have developed a set of 123 RPM-inspired problems. These problems are broken into three categories: 27 2x1 matrices (as shown in Figure 1), 48 2x2 matrices (as shown in Figure 2), and 48 3x3 matrices (as shown in Figure 3). Although there are no 2x1 matrices in the actual RPM test, these are included in our set to provide a simpler initial set of problems for students to address before moving on to more difficult problems.

To develop these RPM-inspired problems, we examined individual problems on the actual RPM tests (both the standard and the advanced test) and wrote problems to have a close correspondence with the problems on the actual tests. Although the individual shapes and their properties differ, these RPM-inspired problems mimic the same transformations and problem types as the actual standard and advanced RPM tests. These correspondences, however, only exist at the level of individual problems; not every RPM has a corresponding RPM-inspired problem in our problem sets, and some types of problems are present more often in our problem sets than in the actual RPM tests. Therefore, no claim is made that our RPM-inspired problem sets are equivalent to the RPM tests as a whole; we only claim that the individual problems capture the same reasoning as problems on the original RPM tests. We are presently running two previously-designed agents (Kunda. McGreggor & Goel 2011; McGreggor, Kunda & Goel 2014) for solving the actual RPM tests against these new RPM-inspired problems in order to establish a conversion factor between the two sets.

## The Projects

In the Fall 2014 version of the KBAI class, students completed a series of four projects. In the first three projects, students designed agents that could address 2x1, 2x2, and 3x3 matrix problems. During these projects, the input into these agents was propositional representations of the 123 RPM-inspired visual analogy problems. The propositional representations were written by the instructors of the course to prevent students from building inferential advantages into the representations. During the design of their agents, students could see 83 of these problems: the remaining 40 were designated 'Test' problems and were hidden from students in order to test their agents for generality. Thus, students were encouraged to construct agents with general problem-solving ability rather than agents that would tightly fit a small set of previously-seen problems.

By the end of project 3, students had completed an agent that could solve 2x1, 2x2, and 3x3 visual analogy problems based on propositional input. In project 4, students designed an agent that could solve these same problems using visual input. Here, students' agents read in the images directly from .PNG files, with one file representing each frame from the problem. Students' agents were run against the same 123 problems. Students' grades were dependent on performance on 100 of these problems (the remaining 23 were provided as challenge problems with no credit granted for correct answers), and 40 of these 100 problems were withheld as 'Test' problems. This paper focuses only on the agents designed in project 4, which took visual input.

Table 1: Performance on the eight sets of RPM-inspired problems (123 problems in all). "*n*" gives the number of problems in that set. "Avg." gives the average number of correct answers in that set for the 224 agents. "1", "2", "3", and "4" give the performance of the four agents described in further detail under 'Four Agents', below.

| | *n* | Avg | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| 2x1 Basic | *20* | 8.8 | 18 | 14 | 17 | 12 |
| 2x1 Extra | *7* | 1.5 | 4 | 1 | 7 | 2 |
| 2x2 Basic | *20* | 8.8 | 18 | 16 | 20 | 14 |
| 2x2 Extra | *8* | 2.5 | 7 | 4 | 7 | 7 |
| 2x2 Test | *20* | 7.2 | 17 | 16 | 14 | 12 |
| 3x3 Basic | *20* | 11.0 | 19 | 17 | 20 | 15 |
| 3x3 Extra | *8* | 1.5 | 2 | 0 | 6 | 4 |
| 3x3 Test | *20* | 7.9 | 16 | 15 | 11 | 13 |

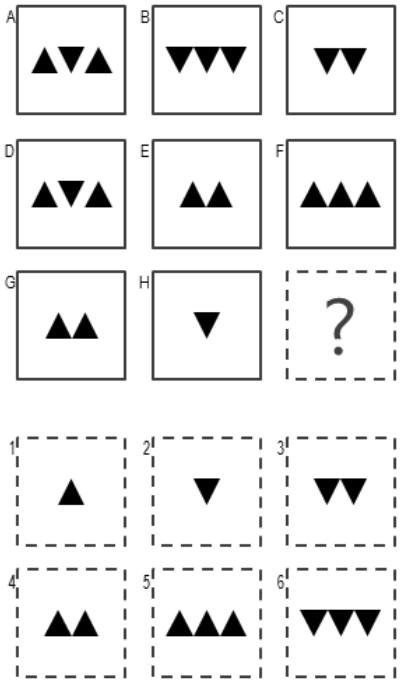


Figure 3: A 3x3 visual analogy problem, inspired by Raven's Progressive Matrices. Individual objects within frames in an RPM can be said to have 'properties'; for example, some of the triangles in this problem have a 180° rotation as a property.

## Aggregate Results

Students in the KBAI class submitted 224 agents, each of which ran against the 123 problems. The percentage of agents answering an individual problem correctly ranged from 87% (fo r the easiest 3x3 problem, which involved no transformations between frames) to 8% (for the hardest 3x3 problem, which demanded reasoning about the sum of the number of sides of multiple shapes). Among the problems completed for credit, one Test problem was correctly answered by only 10% of agents; this 2x2 problem involved two transformations – change-fill and remove-shape – that conflicted with one another.

Table 1 previously shows the performance of the agents as a whole, as well as the performance of the four agents highlighted below. The table is broken up by the eight distinct problem sets students addressed: 'Basic' sets *were* provided to students during the design of their agents and *were* evaluated for the project grade; 'Test' sets *were not* provided to students for the design of their agents and *were* evaluated for the project grade; 'Extra' sets *were* provided to students during the design of their agents but *were not* evaluated for the project grade. Agents' scores on the Basic and Test sets comprised 70% of students' project grades.

Perhaps surprisingly, students' agents performed better on 3x3 problems than on 2x2 problems. While 3x3 problems allow more complex problem structures, such as transformations in which two frames together determine the contents of a third, students noted that 3x3 problems gave their agents more information with which to work. With more information, their agents performed better, even on more complex problems.

## Four Agents

After evaluating the aggregate results, we examined the problem-solving methods of several of the best-performing agents and identified a number of particularly novel and successful methods for addressing these RPM-inspired problems. The majority of the 224 submitted agents operated by first writing a propositional representation based on shape recognition, and then solving the problem propositionally; we describe the most successful agent using this method below, which combines contour recognition with problem classification. However, we also identified several other methods to solving these problems. Here, we describe three additional creative methods to solving RPM-inspired problems based on imagistic representations.

## Agent 1: Contour Recognition & Reasoning

Agent 1 uses an intermediate propositional knowledge representation for working memory. In the agent's representation, each frame in an RPM consists of objects, and each object consists of the following attributes: shape, size, fill, rotation, and relative-position to other shapes. A library of shapes was available to the agent, storing 20 basic shapes and features such as symmetry and corner count. Agent 1's method has three phases: symbol extraction, top-down recognition, and bottom-up recognition.

Phase 1 uses image processing to extract a propositional representation for each problem. First, objects are found by isolating connected components, after which they are classified into shapes based on attributes of the object like corner count, edge lengths, and convexity. Other object attributes, including fill, rotation, size, and relative position are also computed in this phase.

Phase 2 uses top-down pattern finding. 19 pattern recognizers look for simple patterns that will be combined to form a pattern fingerprint. Recognizers include "constant rotation across objects in frame" (as seen between frames A and C in Figure 2) and "object count arithmetic sequence." For each problem matrix, patterns are found and combined for all in-row, -column, and -diagonal relationships. The agent then chooses the answer with the largest set of matchers. In the event of a tie, Phase 3 begins.

Phase 3 performs bottom-up reasoning by splitting each problem into 2x1 sub-problems: 2 for 2x2 matrices and 29 for 3x3 matrices (including diagonal sub-problems). The agent solves each sub-problem, producing multiple answer choices, then uses majority-rule to make a final answer selection.

To solve a 2x1 sub-problem, (1) all object pairs from frame A to frame B are created; (2) all object pairs from frame C to the answer choices are created; (3) all mappings between object pairings from step one and step two are created; and (4) each mapping is given a score. The scoring function includes the intuitiveness of the transformation in step two and the strength of analogy in step three. For example, a mapping would be scored highly for intuition for mapping a triangle from frame A to a triangle in frame B. However, if a triangle in frame A instead mapped to a square in frame B, the best analogy would map triangles from frame C to squares in frame D. The highest scoring mapping is the most intuitive analogy. In the worst case, phase 3's runtime is $O((n!)^3)$, where n is object count per frame. To offset this, time limits were imposed.

To take the problem shown in Figure 3 as an example: during Phase 1, 31 shapes and 14 frames would be represented in a fashion similar to the following: frames: [{id: 1, objects:[{id: 1; shape: triangle; fill: yes; angle: 0; left-of: [2, 3]; size: medium},{id: 2; shape: triangle; fill: yes; angle: 180; left-of: [3]; size: medium}…]}, …].

During Phase 2, each potential answer is inserted into the last cell of the matrix, and each pattern matcher runs. Here, the matcher labelled "remaining shapes after pairing" will match: each upright triangle in the first cell of a row or column is paired with a flipped version in the second cell, and the remaining triangles are checked to see if they match those of the third cell. Other matchers may also match the inserted choice, creating a more complex pattern. In the end, each potential answer will have a list of matchers associated with it, and the one with the longest list of matchers is selected. For this problem, the agent would choose the first answer choice. Because the problem would be solved in Phase 2, Phase 3 would not execute.

Agent 1 performed exceptionally well, correctly answering 101 of the 123 problems (88 of the 100 problems for credit). Agent 1's general method of generating a representation based on prior shape knowledge also reflects the most common approach used in the class (as well as an approach used in prior literature, e.g. O'Donoghue, Bohan, & Keane 2006); however, Agent 1's classification of multiple problem types goes beyond what the majority of agents attempt and plays a large role in its success.

Connecting with computational creativity, Agent 1 possesses the ability to creatively generate its own answers. Presently, Agent 1 operates by substituting each answer choice in the empty frame and evaluating its degree of fit to the problem's transformations; however, implicit here is the idea of an 'optimal' fit for the remaining frame. Were the agent deprived of the answer choices, it could instead generate the optimal solution for the empty frame. Agent 1 is limited in this regard, however, in that it could only produce solutions that are comprised of the shapes in its shape library; Agent 1 cannot deal with novel shapes.

## Agent 2: Shape-Agnostic Transformation Recognition

The second agent, Agent 2, operates in two stages. First, the agent detects and analyzes individual objects to produce a propositional representation, similar to Agent 1. The agent uses the individual properties to find relationships between objects in pairs of frames, and chooses the answer that best fits the relationships that are found. Agent 2's high-level process thus resembles Agent 1's in its initial phase of translating imagistic representations into propositional ones; however, it differs in that it does not rely on prior shape knowledge. Agent 2 derives the structure and content of the problem from within the problem, rather than based on prior knowledge of shapes and features.

The agent begins by recording visual measurements for each object in the problem and using a simple clustering method to partition similar objects into shape groups. The agent records the width/height ratio of an object and the amount of whitespace "outside" of the object's boundaries in its cropped region. Without predefined knowledge of triangles and squares, the agent instead categorizes shapes based on these properties and gives them arbitrary names. For example, the agent may label all triangles as "shape1" and all squares as "shape2", even if the individual objects vary in size and other properties across the problem, based on these measurements. To account for variations in the measurements, objects are rotated to optimize an arbitrary scoring function. This also helps determine relative rotation angles between objects which are necessary in certain problems.

To take an example, in Figure 1, there are no overlapping objects in the frames. Individual objects are easily isolated, and the shapes of these objects are distinguished by the relative outside whitespace. Other properties, such as relative size and position, are also computed. In frames A and B, the agent records as the target relationship that the single object in frame B has the same shape (shape2) as both of the objects in frame A and the same size as the larger object in frame A. The agent then compares frame C with each answer frame to find the closest match to this relationship. An exact match is not possible because frame C contains two different shapes (shape1 and shape3) rather than a single shape. The correct answer, frame 2 with the large triangle (shape3), is chosen because it matches all aspects of the target relationship other than the object matching the shape of the smaller object. Thus, the concept of shape is used to mark objects as being different from or similar to other objects, and as long as the agent correctly observes those differences in the visual analysis portion it will have enough information to solve the problem.

The process for the problems in Figures 2 and 3 is similar, although the addition of rotating objects demands the

agent's rotation logic. For example, in the first frame of Figure 3, the two outer triangles are already at the "ideal" rotation angle and are given an angle value of 0 degrees, whereas the middle triangle would reach the same "ideal" value after being rotated 180 degrees. As noted before, the primary difference between Agent 1 and Agent 2 is that while Agent 1 relies on prior knowledge of shapes and their potential properties, Agent 2 takes a grounded method to identifying shapes in a frame. Thus, while Agent 1 will fail to recognize previously unseen shapes, Agent 2 is equipped to address previously unidentified shapes.

Agent 2 performed exceptionally well, correctly answering 83 of the 123 problems (78 of the 100 problems for credit). It is notable, though, that Agent 2's performance lagged behind on the 'Extra' problem sets; many of these sets included transformations, such as counting the sides of a shape, for which Agent 2's more visually-oriented method does not account. We also hypothesize Agent 2 would show greater success on problems featuring previously unseen shapes that humans could similarly address, but no such problems were included here.

Like Agent 1, Agent 2 can also generate novel answers rather than select them from a set of possible answers. The paragraph above acknowledged that on the problem presented in Figure 1, the most-obvious answer to Agent 2 is not present among the answer candidates. To have a 'most obvious' answer prior to examining the choices, Agent 2 must generate its own solutions. This also reveals how the presence of candidate answers can encourage creativity by introducing new constraints. It is creative to generate novel solutions from scratch, but it is also creative to generate arguments for available non-obvious solutions.

## Agent 3: Visual Heuristics

In contrast to Agents 1 and 2, Agent 3 does not derive any representation of the visual analogy problems. Agent 3 begins from the supposition that it is fundamental to reduce the input space to something both *manageable* and *meaningful* for the agent to be able to *compute* and *correctly guess* an answer from the given choices. Agents 1 and 2 do so by reducing the input space to a propositional representation; Agent 3 reduces the input space to sets of contiguous non-white pixels.

Agent 3 takes each possible answer choice and computes the likelihood it is correct. To do so, the agent takes a series of measurements capturing the relationship between each training pair, which is described by any two adjacent cells in the matrix. It then compares those measurements against each of the test-answer pairs, the combinations of any cell adjacent to the empty slot and each answer choice. Each comparison, if significant enough, casts a vote for the current answer as the likely answer with a weight directly proportional to the believed similarity of the cells. The most-voted answer is selected as the agent's answer.

Many relationship measurements were evaluated, such as grid-based similarity, histogram-based similarity, and affine transformations. After multiple iterations, few measures were needed to yield the best performance. In the final design, the agent only uses the following two measurements:

- **Dark pixel ratio**: the difference in percentage of the number of dark-colored pixels with respect to the total number of pixels in the contiguous pixel sets of two matrix cells.
- **Intersection pixel ratio**: the difference in percentage of the number of dark-colored pixels present at the same coordinates with respect to the total number of dark-colored pixels in both matrix cells for a given set of contiguous pixels.

For example, in Figure 1, the intersection pixel ratio would lead the agent to vote for the answers containing an outer square; this is analogous to the most logical answer to the problem, an outer square with the inner object removed. Counterintuitively, the correct answer is just the expanded triangle, but the agent would also vote for that answer based on the dark pixel ratio's similarity to the most logical answer. Hence, thanks to the simple metrics used, the agent is "immune" to problems that may appear deceiving at first glance or may involve convoluted transformations. Although for this particular example, the agent picked answer 6, the correct answer was evaluated to be only 6.76% less likely to be correct.

Agent 3 performed exceptionally well, correctly answering 102 of the 123 problems (82 of the 100 problems for credit). Agent 3 gave the most correct answers of any agent, although a greater proportion of its correct answers were previously-seen problems than Agent 1's similarly high performance. This may suggest that the iterations examining the effectiveness of multiple measures of similarity may have overfit the agent's reasoning to those problems, and that further development with more problems may expand the set of desirable measurements.

Unlike Agents 1 and 2, Agent 3 does not have the capability of generating an answer choice rather than selecting from a set of presented answer choices. This is because while Agents 1 and 2 operate under an implicit ranking of possible choices culminating in an ideal choice, Agent 3 might find numerous options equally ideal, and thus could generate thousands of candidate selections.

## Agent 4: Hybrid Reasoning

Agents 1 and 2 use propositional representations of the target problem while Agent 3 uses purely imagistic representations; Agent 4, by contrast, leverages both and takes a hybrid method. This method asks the question: can an agent quickly find patterns and relationships in a problem through a high-level visual comparison? If the agent can find high-level visual relationships quickly, it can efficient-

ly formulate a solution without any further propositional understanding of the problem. If no such visual relationships are found, the agent may look for lower level propositional relationships present in the problem.

Thus, Agent 4 starts by examining frames for visual relationships and transformations that can be quickly detected by visual inspection. The agent uses image similarity to detect rotation, vertical and horizontal reflection, the identity transformation, image addition, XOR, and NOR. If this process detects the presence of one of these relationships within a matrix problem, the agent generates a prospective solution and looks for a matching answer. For example, in Figure 1, the transformation between frame A and frame B would be identified through the XOR transformation, which searches for pixels present in only one of two frames. Similarly, in Figure 2, the transformation between frame A and frame B would be identified through the rotation transformation; the agent would (successfully) identify frame 3 as a frame that would complete the same rotation transformation when paired with frame C.

This imagistic method was successful in finding solutions to over 20% of the problems, and it was much more computationally efficient compared to extracting propositional representations from the images; this is notable in that it acknowledges the different levels of effort applied by humans in solving these problems. Results could be further improved by searching for more types of high-level relationships and transformations, by applying transformations at a lower granularity than at the image level, and by improving the image comparison. For example, at present, Agent 4 is unable detect the visual transformations between parts of frames in Figure 2.

This visual method has difficulty finding relationships that cannot be represented through affine transformations, such as problems involving prior knowledge of shapes and properties represented in the frames. When the agent is confronted with problems like these, it will try to find low-level relationships using contour recognition to identify shapes and object properties, ultimately leading to a method similar to Agent 1.

Agent 4 performed exceptionally well, correctly answering 79 of the 123 problems (66 of the 100 problems for credit). Although these scores are the lowest among these four agents, they are in the top 10% of agents submitted. Moreover, Agent 4 may represent the best approximation of human reasoning; humans can discuss problems in both visual and propositional terms (Kunda, McGreggor & Goel 2011), and Agent 4 similarly can do both.

As noted in the description above, during the first phase of its reasoning, Agent 4 generates prospective solutions and compares those prospective solutions to the answer choices. Thus, it already engages in creative answer generation and compares the generated answers to the candidate solutions.

## Discussion

Agents 1 and 2 above exemplify Hunt's (1974) analytical, propositional reasoning strategies for addressing RPM problems. Agent 1 extracts propositonal representations that describe the shapes, spatial relations, and transformations from the input images, and then operates on those representations. Agent 2 also extracts propositional representations, but these representations are grounded in the transformations between objects: it has no prior knowledge of shapes, but rather the ability to generate representations of the transformations themselves. Agents 3, on the other hand, exemplifies Hunt's "Gestalt" visual reasoning strategy for RPM. It uses visual abstractions over problems to approximate the answer even without precise knowledge of the transformations between frames. Agent 4 combines the two methods: it first leverages the immediately-identifiable "intuitive" answer that can be established from accessible visual transformations before resorting to more complex propositional reasoning strategies. Thus, Agent 4 demonstrates the possibility of creatively combining methods. As far as we know, the precise strategies used by these agents have not appeared in the literature on the RPM test.

These four agents, along with the 220 other agents developed over the course of this project, reflect the ability of AI agents to succeed on a test of human intelligence that relies on creative and flexible problem-solving. This experiment suggests that there may be no one single "right" problem-solving strategy for the RPM test, that creativity on the RPM test may entail a large number of problem-solving strategies, and that we have so far discovered only a subset of creative problem-solving strategies. Future research along these same lines will test future agents against the authentic RPM test; examine patterns of errors in agents' performance for comparison to human performance (Kunda et al. 2013) including atypical cognition (Kunda & Goel 2011); and better articulate the strengths and weaknesses of different methods (Lynn, Allik, & Irwing 2004; Kunda et al. 2013). We will also examine merging multiple agents into a single agent equipped with metacognitive ability to select among the different strategies, thus more closely approximating factors that determine human success on such tests (Keating & Bobbitt 1978).

## Conclusions

The RPM test admits many problem-solving methods, which in part is what makes it a good test of intelligence and creativity. The various problem-solving methods differ in both the knowledge representations and control of processing they use. In this paper we described a human computation strategy for acquiring novel problem-solving methods for addressing RPM-inspired visual analogy problems. This strategy resulted in the design of 224 AI agents for addressing 123 visual analogy problems. Some of the

agent designs were both novel and effective: we described four of these agent designs.

An important issue in computational creativity is how to acquire knowledge of creative methods. Our research suggests that human computation may be a useful strategy for this acquisition, especially when the computation comes from intelligent, educated, high-achieving humans who themselves are likely to do well on a creative task.

## Acknowledgements

We thank all 224 students in both the in-person and online sections of CS 7637 KBAI course at Georgia Tech in Fall 2014. Goel was the primary instructor of both sections; Joyner was the course developer and head TA of the online section; Lemmon, Graham, Martinez, and Bedwell were four students in the online course and developed agents 1, 2, 3, and 4, respectively.

We are grateful to Maithilee Kunda and Keith McGreggor for their prior work on which this project builds. We also thank the course's teaching team: Lianghao Chen, Amish Goyal, Xuan Jiang, Sridevi Koushik, Rishikesh Kulkarni, Rochelle Lobo, Shailesh Lohia, Nilesh More, and Sriya Sarathy. We also thank the anonymous reviewers of this paper: their comments truly helped improve the discussion.

## References

Bringsjord, S., & Schimanski, B. (2003). What is Artificial Intelligence? Psychometric AI as an answer. In *Procs*. *18th IJCAI*, 887-893.

Dastani, M., Indurkhya, B., & Scha, R. (2003). Analogical Perception in Pattern Completion. *JETAI 15*(4), 489-511.

Evans, T. (1967). A Program for the Solution of a Class of Geometric Analogy Intelligence-Test Questions. In M. Minsky (ed.) *Semantic Information Processing*. MIT Press.

Goel, A. & Joyner, D. (2014). CS7637: Knowledge-Based AI: Cognitive Systems [Online Course]. Retrieved from http://www.omscs.gatech.edu/cs-7637-knowledge-based-artificial-intelligence-cognitive-systems/

Goel, A. & Joyner, D. (2015). An Experiment in Teaching Cognitive Systems Online. Technical Report, Georgia Institute of Technology.

Goel, A., Kunda, M., Joyner, D., & Vattam, S. (2013). Learning about Representational Modality: Design and Programming Projects for Knowledge-Based AI. In *Fourth AAAI Symposium on Educational Advances in Artificial Intelligence*.

Howe, J. (2008). *Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business*. Crown.

Hunt, E. (1974). Quote the raven? Nevermore! In L. W. Gregg (Ed.), *Knowledge and Cognition*. 129-158. Hillsdale, NJ: Erlbaum.

Keating, D. , & Bobbitt, B. (1978). Individual and developmental differences in cognitive-processing components of mental ability. *Child Development*, 155-167.

Kirby, J., & Lawson, M. (1983). Effects of strategy training on progressive matrices performance. *Contemporary Educational Psychology*, *8*(2), 127-140.

Kunda, M., & Goel, A. (2011). Thinking in Pictures as a Cognitive Account of Autism. *Journal of Autism and Developmental Disorders, 41*(9), 1157-1177.

Kunda, M., McGreggor, K., & Goel, A. (2013). A Computational Model for Solving Problems from the Raven's Progressive Matrices Intelligence test using Iconic Visual Representations. *Cognitive Systems Research, 22*, 47-66.

Kunda, M., Soulieres, I., Rozga, A., & Goel, A. (2013). Methods for Classifying Errors on the Raven's Standard Progressive Matrices Test. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*, 2796-2801. Berlin, Germany.

Law, E., & von Ahn, L. (2011). *Human Computation*. Morgan & Claypool.

Lovett, A., Tomai, E., Forbus, K. & Usher, J. (2009). Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science 33*(7), 1192-1231.

Lynn, R., Allik, J., & Irwing, P. (2004). Sex differences on three factors identified in Raven's SPM. *Intelligence, 32*, 411-424.

McGreggor, K., Kunda, M., & Goel, A. (2014). Fractal and Ravens. *Artificial Intelligence 215*, 1-23.

O'Donoghue, D., Bohan, A., & Keane, M. (2006). Seeing Things: Inventive Reasoning with Geometric Analogies and Topographic Maps. *New Generation Computing 24* (3), 267-288.

Prade, H. & Richard, G. (2011). Analogy-Making for Solving IQ Tests: A Logical View. In *Procs*. *19th International Conference on Case-Based Reasoning*, 561-566. London, UK: Springer.

Ragni, M. & Neubert, S. (2014). Analyzing Raven's Intelligence Test: Cognitive Model, Demand, and Complexity. In H. Prade & G. Richard (Eds.) *Computational Approaches to Analogical Reasoning: Current Trends*, 351-370. Springer.

Raven, J., Raven, J. C., & Court, J. (1998). *Manual for Raven's Progressive Matrices and Vocabulary Scales*. San Antonio, TX: Harcourt Assessment.

Schwering, A., Krumnack, U., Kuhnberger, K-U, & Gust, H. (2009). Spatial cognition of geometric figures in the context of proportional analogies. In *Procs*. *Spatial Information Theory, Lecture Notes in Computer Science Volume 5756*, 18-35.

Snow, R., Kyllonen, P., & Marshalek, B. (1984). The topography of ability and learning correlations. *Advances in the Psychology of Human Intelligence*, *2*, 47-103.

# Accounting for Bias in the Evaluation of Creative Computational Systems: An Assessment of DARCI

**David Norton, Derrall Heath** and **Dan Ventura**
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
dnorton@byu.edu, dheath@byu.edu, ventura@cs.byu.edu

### Abstract

Recent investigations into the assessment and evaluation of "creative" systems in the field of computational creativity have disclosed several problems common to research within the field. We perform a practical evaluation of the latest iteration of the creative system, DARCI, attempting to address some of these problems using a specially designed, but generalizable, online human survey. Of note, we address the complications of evaluator bias that are present in all assessments of creativity. Using our evaluation, we show that within its narrow domain, DARCI is able to produce artifacts that are rated at least as favorably as human counter parts across five aspects of creativity. Further, these artifacts tend to be more surprising and perceived as more difficult to produce than those created by human artists.

## Introduction

Recent investigations into the assessment of "creative" systems in the field of computational creativity have disclosed several problems common to research within the field. The first problem is properly focusing assessments to the intended scope of a given creative system: how much should an evaluation focus on the artifacts themselves—*weak computational creativity*—and how much should it focus on the processes involved in creating the artifacts—*strong computational creativity* (al Rifaie and Bishop 2012)? The second problem is determining measurable assessment criteria that can be used to determine if one version of a creative system is an improvement over another, or to compare two different creative systems (Colton et al. 2014). The third problem is empirically grounding the ambiguous terminology that is commonly used to describe and assess creative systems (Brown 2014). The fourth problem is picking, or designing, the best methodology to actually carry out the assessment of a system (Jordanous 2014). The fifth problem, and one that is not addressed in detail by researchers in the field, is compensating for the effects of bias inevitably introduced by human evaluators when assessing creative systems.

While the researchers exploring these issues have presented tantalizing theoretical solutions, few have implemented practical solutions (a noted exception is Jordanous' meta-evaluation of existing evaluation methodologies (2014)). In practice, as each of the researchers have noted, there is no straightforward solution to any of these problems. Here we perform a practical evaluation of the latest iteration of the DARCI system, attempting to address some of these problems using a specially designed, but generalizable, online human survey. Of note, we address the complication of bias introduced by human evaluators that is unaccounted for in current assessments of creativity.

There has been some reticence in the community towards conducting human surveys as a means of evaluation. Brown notes that human surveys often have wide variance making them difficult to incorporate into established models of creativity (2014). In a study comparing several methods of evaluation, Jordanous concludes that human surveys were the least correct of the methods she explored (2014). She suggests that this was because participants, unsure of the definition of creativity, evaluated systems based on other factors. However, anonymous online surveys can quickly gather many responses from individuals outside of the computational creativity community. Having this outside opinion is valuable as it reduces biases that those within the community inevitably bring to assessments. We evaluate DARCI through such a survey, but, in order to reduce participant confusion and response variance, ask participants to evaluate a variety of explicitly defined artifact qualities (that correspond to requirements for creativity) rather than asking them to directly evaluate the system's creativity.

Brown stresses the inadequacy of human surveys as empirically grounding assessments since we don't have an understanding of what the human responses mean (2014). In order to gain that understanding on some level, we develop a standard for judging the artifact qualities that we measure. The standard is created by having survey participants assess human artifacts (the standard) in addition to DARCI's.

In order to evaluate a creative system from a *strong computational creativity* standpoint, Colton et al. argue that the process by which a system produces artifacts, in addition to the artifacts themselves, must be evaluated (2014). While our survey questions do focus on the artifacts, some are designed to glean opinions about DARCI's creative process. Unfortunately, in order for survey takers to evaluate this process, the survey cannot be blind. Participants in the survey will know that they are evaluating an artificial system, and bring with that knowledge unwanted biases. These biases may be negative if the viewer feels that art is an inherently human affair that automatically renders a computer's efforts

invalid. Or, they may be positive if the viewer feels that the computer has an unfair disadvantage and should thus be graded on a curve. Another possible source of positive bias is potential viewer familiarity with computational creativity, or even DARCI itself, and a concomitant desire for the study to succeed.

In order to evaluate DARCI's creative process while taking into consideration the effects of evaluator bias, we design the survey to detect the level of human/computer bias in each survey taker. We then use this information to determine the effects of survey taker bias and adjust our conclusions from the survey accordingly.

## DARCI and Artifact Creation

DARCI is composed of several subsystems, each with its own creative potential, and each designed to perform an integral step of image creation from conception of an idea, to design, to various phases of implementation, to curation. The most complete subsystem, and the one that is the focus of this paper, is called the *image renderer*. The image renderer uses a genetic algorithm to discover a sequence of image filters for rendering an image composition (produced by another subsystem) so that it will reflect a given description (selected from yet another subsystem).

DARCI is designed to produce a rendering for a given source image that reflects a given adjective(s) in an *interesting* way. As detailed in previous research, by *interesting* we mean that the rendering is different enough from the source image so as to satisfy the creativity requirement of originality while not being too different from the source image so as to satisfy the creativity requirement of functionality (Norton, Heath, and Ventura 2014).

To produce its artifact, DARCI first uses a system of genetic algorithms to build a pool of candidate artifacts from which to select the final rendering. Once these candidates have been created, DARCI uses a heuristic to rank them and then selects the top ranked candidate as the final artifact.

### Candidate Artifact Creation

DARCI begins by training a binary artificial neural network (ANN) for the given adjective. This neural network, called here the *adjective ANN*, is trained to associate 51 image features with the adjective using standard backpropagation and a training set of hand-labeled images. The 51 image features describe a variety of image qualities including color, lighting, texture, and local interest points, and were chosen from a larger set of 198 features using forward feature selection as described by Norton et al. (2015). Many of these image features are the result of psychological studies analyzing the connection between color and various affective words (Ou et al. 2004; Wang, Yu, and Jiang 2006; Machajdik and Hanbury 2010). Others summarize local interest point data that is typically reserved for object detection in images (Norton, Heath, and Ventura 2015). Still other features come from a publicly available[1] set of widely accepted global image features (King, Ng, and Sia 2004).

[1] http://appsrv.cse.cuhk.edu.hk/~miplab/discovir/

Once the adjective ANN is trained, DARCI uses a genetic algorithm to discover the configuration and parameter settings of Photoshop-like filters for rendering the source image to reflect the given adjective. Candidate filter sequences are evaluated by applying them to the source image and using the resulting image as input to the adjective ANN. The output of the adjective ANN is the fitness score. To increase the variety of renderings discovered by the genetic algorithm, speciation is introduced by including sub-populations.

After a number of generations of evolution (in our case 100) the renderings corresponding to the ten highest scoring filter sequences discovered per sub-population are returned. In these experiments, we use six sub-populations, yielding 60 images. These select images are ordered by fitness, then added to the pool of candidate artifacts one at a time beginning with the most fit image. Images are only added to the candidate artifacts if they are determined to be sufficiently unique. To identify those artifacts that are not sufficiently unique, the system calculates the normalized cosine similarity between the 51-element feature vector of each potential candidate and the feature vector for each existing candidate. If the similarity is greater than some threshold, the potential candidate is considered redundant and not added to the candidate pool. For our experiments, based on preliminary observations, we set this threshold to 0.95.

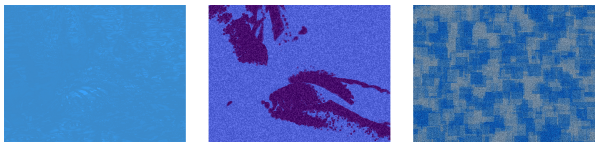
Once the candidate artifacts have been selected, another epoch of evolution is performed. This time a neural network we call the *novelty ANN* is trained to distinguish images novel to DARCI (the hand-labeled images mentioned previously) from those produced by the system (the pool of candidate artifacts). This process is similar to the process employed by Machado et al. in training NEvAr to create novel images (2007).

A new genetic algorithm is initialized using the combined output of the novelty ANN and the adjective ANN as the fitness function. To combine the output of the two neural nets, the system selects the minimum output of the two classifiers as described by Norton et al. (2014). The genetic algorithm performs 100 generations of evolution using the new fitness function. This forces DARCI to produce images that reflect the given adjective *and* are distinct from the images produced earlier. As before, the most fit artifacts are added to the pool of candidate artifacts, provided they are not redundant.

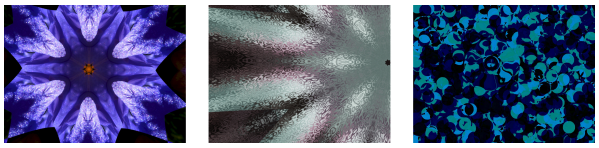
This process is repeated for several epochs, each adding increasingly varied images to the pool of candidate artifacts as the system attempts to optimize the changing fitness function. For our experiments, we perform a total of 8 epochs including the initial novelty-ANN-free $0^{th}$ epoch. Figure 1 illustrates how candidate artifacts vary from epoch to epoch during one experiment with the adjective "cold" using the image of Figure 2 as the source image.

### Candidate Artifact Curation

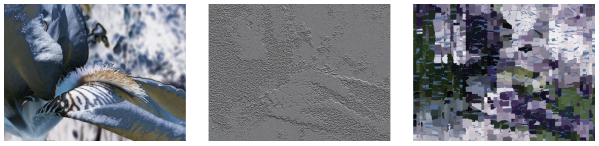
Once the candidate pool has been created, DARCI selects a single rendering to present as the finished product. Curating the candidates consists of two phases. In the first phase, DARCI ranks the candidates by their similarity to the source image and selects the top 10% (see Figure 3a - 3c), increas-



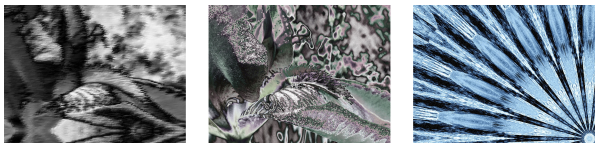
(a) Epoch 0



(b) Epoch 1



(c) Epoch 2

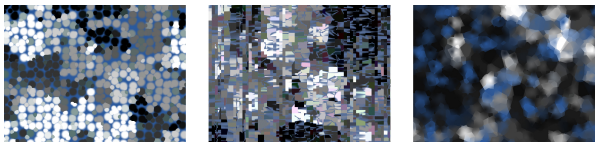


(d) Epoch 3

(e) Epoch 4

(f) Epoch 5

(g) Epoch 6



(h) Epoch 7

Figure 1: Sample artifacts from each epoch of the candidate building process for the adjective "cold" and source image in Figure 2. Note that since the candidate pool is empty during epoch 0, the novelty ANN is not used in the genetic algorithm's fitness function for this epoch.

Figure 2: The source image for all experiments in this paper.

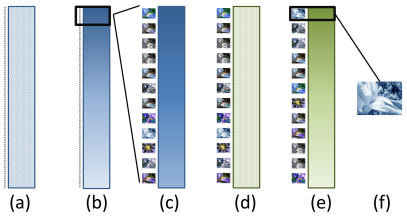
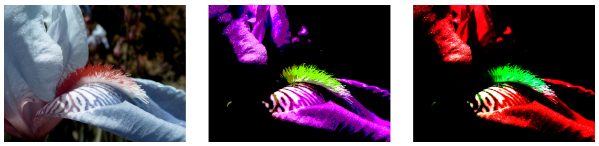



Figure 3: The curation process for selecting a final artifact from the pool of candidates (represented by a colored bar). (a) Each artifact in the pool of candidates is assigned a score of similarity to the source image (in this case Figure 2). (b) The candidates are ranked by this score (depicted by the bar's gradient). (c) The top 10% of ranked artifacts are chosen for the next phase of curation. (d) The remaining artifacts are given a score of how well they match the given adjective. (e) The artifacts are ranked by the new score. (f) The top image is selected as the final artifact to be returned.

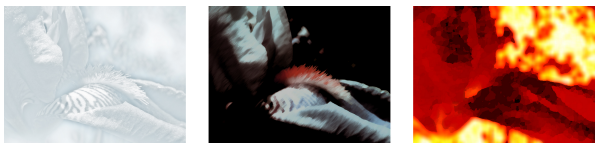
ing the chance that the final rendering will make noticeable use of the source image.

During curation, similarity to the source image is calculated to preserve the content, rather than the color, of the source image. Color usage has been shown to correlate with the affect of images (Wang, Yu, and Jiang 2006; Li and Chen 2009; Norton, Heath, and Ventura 2013), and we would actually like the color of the image to change in order to match the adjective description while keeping major objects within the source image recognizable. Therefore, similarity is calculated by first extracting a 1000-element histogram of visual words from the source image and each candidate artifact (visual words are quantized local image features commonly used in content-based image retrieval approaches (Sivic and Zisserman 2003)). The similarity between two images is calculated by taking the cosine similarity of the images' visual word histograms, as this similarity function has previously been used to successfully preserve the source image (Norton, Heath, and Ventura 2014).

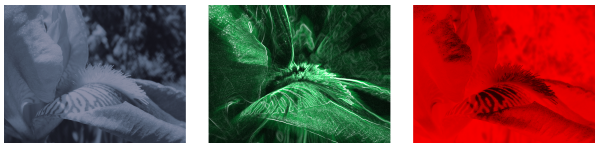
In the second phase of curation, DARCI ranks the remaining candidates by their association with the given adjective using the adjective ANN (see Figure 3d - 3f). The highest ranked image is then selected as the final artifact. This second phase occurs after over-filtered images have been removed in order to increase the chance that the final artifact reflects the given adjective and to reduce the possibility of returning an under-filtered image.



(a) Human 1



(b) Human 2

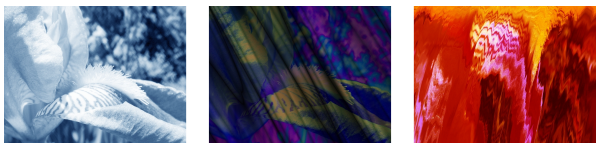


(c) Human 3

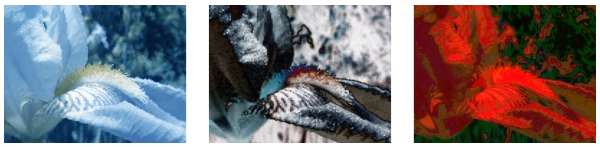
(d) Human 4

Figure 4: Renderings of Figure 2 created by four human artists. The renderings were created to depict, from left to right, the adjectives “cold”, “eerie”, and “violent”.

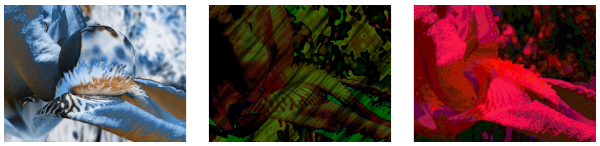
(a) *DARCI 1



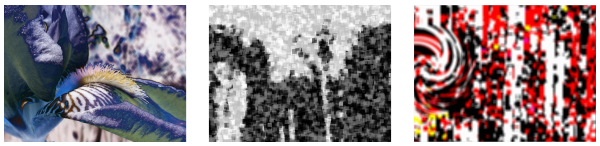
(b) *DARCI 2



(c) *DARCI 3



(d) *DARCI 4



(e) DARCI 5

(f) DARCI 6

(g) DARCI 7

Figure 5: Renderings of Figure 2 created by DARCI. The first four sets (*) were selected for the study. The renderings were created to depict, from left to right, the adjectives “cold”, “eerie”, and “violent”.

## Commissions

For our experiments, we commissioned DARCI and four human volunteers to produce renderings of the photograph in Figure 2 that depict it as “cold”, “eerie”, and “violent”, respectively. These adjective were chosen because DARCI is able to associate them with images effectively (Norton, Heath, and Ventura 2015), they are affective, and they haven’t been used extensively in previous studies involving DARCI. In order to keep the rendering tools available to DARCI and the human artists as similar as possible, human artists were restricted to a subset of tools found in software packages used for photo manipulation.

All four human volunteer artists have experience working with photo manipulation software, and, for grounding, they were shown examples of human-produced renderings from a previous study. The 12 images they produced for our study are shown in Figure 4.

We commissioned DARCI seven times for each of the three adjectives. Each commission produced one artifact as outlined in the previous section. In order to increase output diversity across these commissions, the error threshold used in training the neural networks was varied for several commissions. To match the number of human commissions, we selected four of the artifacts DARCI produced for each adjective. We made the final decision to ensure varied artifacts and to eliminate potential outliers. Figure 5 shows all of the artifacts produced by DARCI and notes our chosen images.

## Online Survey

In order to easily gather many responses, the survey was anonymous and online. To our knowledge, prior to taking the survey all participants were informed that the survey

was to help with research regarding DARCI, "a computer program we created". Furthermore, the survey began by informing volunteers that "the results will be used in research exploring creativity in computational systems".

The survey was separated into two parts. The first part was designed to detect any pre-existing human/computer bias in the survey taker as well as any bias the survey taker may have towards our research in particular (given the survey preface and disclosure of our system). The second part was designed to gather survey takers' opinions about the renderings created by DARCI and the human artists.

**Part 1** Volunteers were presented with 15 pairs of images from which they would indicate their preferences. All images were created by applying random filters from DARCI's toolset to random source images and selecting intriguing and abstract creations from the thousands of random images. In order to limit the number of factors survey takers would be required to consider when making their selection, we paired images together that seemed similar in some respect. These image pairs were presented to volunteers in a random order with random labels. The labels indicated that one of the images was created by a human, and the other was created by a computer program. For each pair, the volunteers were asked which they thought was the better image and given only 10 seconds to respond. Since the images were randomly labeled as "human" or "computer", unbiased volunteers should pick the "human" and "computer" options approximately equally.

**Part 2** All volunteers were randomly assigned to one of three experiments: *blind*, *basic*, or *detailed*. The experiments were identical except for the amount of information that was presented to each volunteer. In all three experiments volunteers were given the following instructions:

> In this part, you will be presented with a total of seven images. You will be asked to indicate your impressions of each image.
>
> Each image was created by either a human artist or a computer program called DARCI. The images were created using digital tools to modify a specific source photograph so that it reflected a given word.
>
> As an example, observe how an artist modified the following source photograph so that it reflected the word "happy".

In the *blind* experiment, volunteers were never given the name of DARCI (it was obfuscated from the above instruction) and were not told which images were produced by DARCI and which were produced by a human artist. In the *basic* experiment, volunteers were told the name of DARCI and which images were produced by DARCI. In the *detailed* experiment, volunteers were not only told which images were created by DARCI, they were also given a detailed (for the layman) description of how DARCI produced its images. This description was followed by a simple one question quiz to assess comprehension.

Aside from the noted differences, the three experiments proceeded identically. Survey takers were presented with the source photograph (Figure 2), noted as such, and then six random images presented in random order: one image from DARCI and one from a human artist for each of the three adjectives ("cold", "eerie", and "violent"). Only six of the twenty-four possible images were presented to reduce fatigue. Volunteers were required to evaluate each image by indicating how strongly they agreed or disagreed with a series of 7-point Likert items. To assist with these items, volunteers were always allowed to view the source photograph.

For all images, except the source, the Likert items were (*adjective* taking the place of the appropriate adjective):

> "I like this image." (*like*)
>
> "This image is *adjective*." (*adjective*)
>
> "This image is a surprising modification of the source photograph." (*surprising*)
>
> "This image would be difficult to create from the source photograph." (*difficult*)
>
> "This image makes good use of the source photograph." (*use*)

For the source image we asked about all three adjectives, and omitted the three items that referred to the source.

Participants were not asked to explicitly assess the creativity of artifacts since personal notions of creativity vary widely. Instead, these five items were chosen to succinctly capture certain qualities required to attribute creativity to a system via the artifacts it produces, and to a small extent, its creative process. Norton et al. have shown that a similar set of Likert items are reliable (using Cronbach's alpha) and correlate with participants' opinions of creativity as measured by an additional Likert item explicitly for "creativity" (2013).

Researchers in computational creativity have identified several attributes necessary to attribute creativity or, as Colton has stated, not attribute un-creativity to a system. These attributes include Colton's creative tripod (*appreciation*, *imagination*, and *skill*) (2008), Ritchie's 18 criteria defined by functions of *quality*, *novelty*, and *typicality* (2007), Jordanous' 14 components of creativity (2012), and the American Psychological Association's *functionality* and *originality* attributes.

Many of these attributes relate to the Likert items in the survey. The *like* item relates to the attributes of skill, quality, functionality, and Jordanous' 'domain competence' and 'value' components. *Adjective* relates to the attributes of functionality and Jordanous' 'intention and emotional involvement' and 'social interaction and communication' (particularly in the *detailed* experiments) components. *Surprising* relates to the attributes of novelty, originality, and Jordanous' 'originality' and 'value' components. *Difficult* relates to the attributes of skill and Jordanous' 'domain competence' component, and emphasizes the creation process. Finally, *use* relates to the attributes of functionality, skill, and quality. Since DARCI produces artifacts, all of the Likert items relate to Jordanous' 'generation of results' component, and for the *detailed* experiment where the creative process is disclosed, all of the items relate to Jordanous' 'progression and development', 'thinking and evaluation', and 'variety, divergence, and experimentation' components.

## Results

After removing results from volunteers who indicated that they had either taken the survey before or viewed someone else taking the survey, 284 completed surveys remained. An

additional 46 surveys in various stages of completion were collected and included in calculating applicable results. 100 volunteers were assigned to the *blind* experiments, 111 to the *basic* experiment, and 106 to the *detailed* experiment. For evaluation, results from volunteers who failed the comprehension question were removed from the *detailed* results and added to the *basic* results. This was 68 of the 106 volunteers assigned to the *detailed* experiment.

## Bias

A volunteer's bias was calculated by subtracting the number of images they preferred labeled with "computer" from those labeled with "human" in the first part of the survey. Thus, a positive score indicates a bias in favor of humans. Since the images were randomly labeled, the average bias of all test takers should have been close to 0 if there was no bias. However, the average bias was 0.901 with a standard error of 0.185, indicating a small but substantial bias either towards humans or against DARCI.

When analyzing results from the second part of the survey, we averaged the scores (between 1 and 7) for each Likert item across all artifacts produced by either humans or DARCI for each group of experiments. These results, with standard error, can be seen in Figure 6.

In order to discover the effect of bias on the results in the second part of the survey, we calculated the Pearson correlation coefficient, $r$, between bias and the average Likert item scores for *blind*, *basic*, and *detailed* experiments. A positive correlation between bias and a particular item for a given artist (human or DARCI) would indicate that a bias towards humans (or against DARCI) is correlated with an increase in the item score for the artist. Table 1 shows these correlation values and their $p$-values (calculated with a two tailed Student's $t$-test) for the three experiments.

Only the *detailed* experiment contained a correlation that was statistically significant to $p < 0.05$. That was a positive correlation with the *difficult* item in human produced images. This means that volunteers with a bias towards humans tended to give humans a boosted score for *difficulty* when they understood how DARCI produced images. Even though none of the other correlations were statistically significant, it should be noted that in the two most informed experiments, the correlations were generally more positive towards humans and more negative towards DARCI (as one might expect). But, the lack of significance indicates that bias did not have a substantial impact on most results.

While one might expect no correlation between bias and scores in the *blind* experiment, there was a clear trend towards negative correlation across *all* items, both for humans and DARCI (see Table 1). None of these correlation values were statistically significant, but the fact that almost all of the correlations were negative suggests that there may indeed be an overall negative correlation. This would imply that those with a bias in favor of humans tended to give all images a lower score when they didn't know who produced them. Perhaps these volunteers were concerned that an image might be produced by DARCI. It would be interesting to investigate this phenomenon in future studies.

| **Human** | *blind* | | *basic* | | *detailed* | |
|---|---|---|---|---|---|---|
| Item | $r$ | $p$-value | $r$ | $p$-value | $r$ | $p$-value |
| like | -0.087 | 0.399 | 0.044 | 0.591 | 0.160 | 0.357 |
| adjective | -0.089 | 0.389 | -0.011 | 0.892 | -0.059 | 0.737 |
| surprising | -0.043 | 0.677 | 0.123 | 0.130 | 0.179 | 0.303 |
| difficult | 0.019 | 0.851 | 0.102 | 0.208 | 0.428 | 0.010 |
| use | -0.154 | 0.133 | 0.050 | 0.539 | 0.295 | 0.085 |

| **DARCI** | blind | | basic | | detailed | |
|---|---|---|---|---|---|---|
| Item | $r$ | $p$-value | $r$ | $p$-value | $r$ | $p$-value |
| like | -0.047 | 0.651 | -0.060 | 0.465 | 0.070 | 0.690 |
| adjective | -0.076 | 0.462 | -0.0117 | 0.150 | -0.012 | 0.944 |
| surprising | -0.045 | 0.661 | 0.068 | 0.405 | -0.212 | 0.222 |
| difficult | -0.098 | 0.342 | -0.036 | 0.662 | -0.117 | 0.502 |
| use | -0.073 | 0.480 | 0.002 | 0.983 | 0.029 | 0.869 |

Table 1: The Pearson correlation coefficient, $r$, and associated $p$-value, between volunteer bias and item scores for the three experiments (*blind*, *basic*, *detailed*). Positive correlation indicates that a bias towards humans is correlated with an increase in item score.

## Evaluation

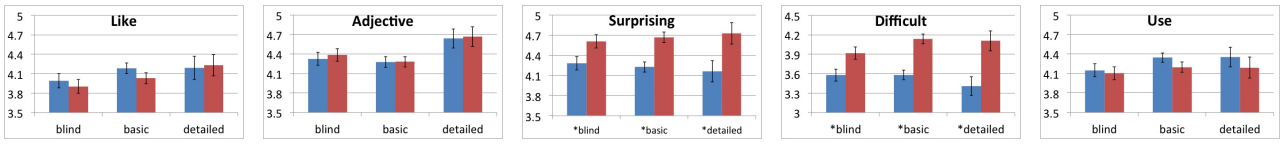
The average scores of the source image across its four Likert items were 5.873 (*like*), 2.260 (*cold*), 1.870 (*eerie*), and 1.377 (*violent*). Looking at Figure 6b we see that both humans and DARCI were able to reflect the adjectives more effectively in their artifacts than did the original source (though at the cost of a lower "like" score).

While the Likert scale is one of the most common evaluation tools used in psychology and marketing research, it has come under criticism for the unintended effects that it can introduce, including cultural biases, memory effects, and the loss of individual subjectivity when the scale is averaged across participants. Recently, it has been demonstrated that ranking or preference questionnaires have fewer negative effects (Yannakakis and Hallam 2011) and that converting from a rating scale to preferences can reduce some of the undesired effects of Likert questionnaires (Martínez, Yannakakis, and Hallam 2014).

To augment the rating-based results of Figure 6, individual survey takers' preferences were calculated from their Likert scores. For each Likert item and for each participant, we performed a pairwise comparison of all images reviewed by the survey taker. We tabulated which images scored higher (were preferred) and when ties occurred in these pairwise comparisons. To summarize the results, we have indicated the percentage of all pairwise tests for each item where human art was preferred, DARCI's art was preferred, and when ties occurred (Figure 7).

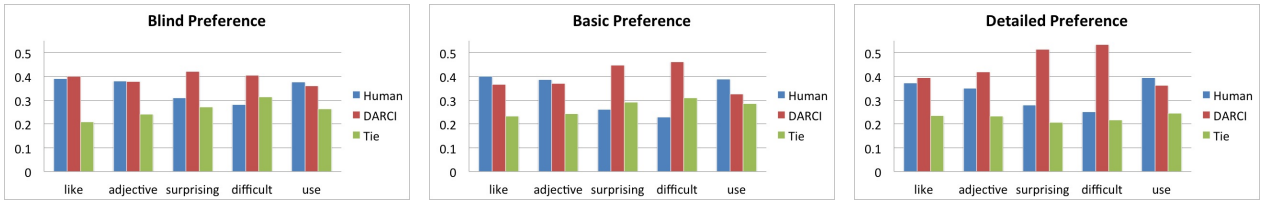
Looking at Figures 6 and 7 we see that DARCI clearly scored higher than human artists in the *surprising* and *difficult* categories while humans did not score substantially higher than DARCI in any category. These trends persisted across all experiments despite the overall human bias of the volunteers. In Figure 6, statistically significant differences ($p < 0.05$ using a two tailed Student's $t$-test) between humans and DARCI are starred (*).

While purely quantitative, these results suggest that




(a) (b) (c) (d) (e)

Figure 6: The average scores of each Likert item across all artifacts produced by either humans (blue, left) or DARCI (red) for each group of experiments (with standard error). (*) indicates statistical significance between human and DARCI results.




(a) (b) (c)

Figure 7: The result of every pairwise test, after converting Likert ratings to pairwise preferences. For each survey taker, pairwise tests were conducted between every combination of images produced by a human and those produced by DARCI.

within this constrained domain of digital visual art, DARCI is capable of producing renderings that are comparable to human renderings in terms of appeal, while being significantly more surprising and unusual. This is more than just a functional evaluation of DARCI's artifacts, it's also an evaluation of the creation process. The fact that DARCI scored higher than humans in the *difficulty* category suggests that volunteers felt that DARCI's artifacts required some skill to create. Additionally, volunteers given details about DARCI's creation process responded to its artifacts very similarly to how those volunteers not given the details responded—understanding how DARCI functioned did not diminish the way the artifacts were perceived.

Somewhat surprisingly, the additional information provided to some of the survey participants had minimal effect on their responses. There was no statistically significant difference between the results of any of the experiments except between the *basic* and *detailed* experiments in the *adjective* category (note the increased scores for both DARCI and human for the *detailed* experiment of Figure 6b). In this one case, understanding how DARCI produced artifacts influenced how volunteers perceived the meaning of the images produced by both DARCI and humans. Since the *detailed* group was told that DARCI learned to associate images with words through training by human teachers, volunteers may have realized that all of the images they were evaluating were essentially examples of what their peers associated with the given adjective. In other words, we suggest that volunteers were incorporating Jordanous' 'social interaction and communication' component into their evaluation.

Table 2 shows the top six images in each category for the three experiments. Refer to Figures 4 and 5 to view the actual images. Of note, DARCI's artifacts have a slightly greater representation amongst the highly rated images.

## Conclusions

We have described recent improvements to a computational system, DARCI, that generates renderings of images so that they reflect an adjective and have presented a human-survey-based instrument designed to evaluate DARCI's artifacts and creation process while taking participant bias into consideration. The instrument uses human artists' artifacts as a baseline for analyzing DARCI's results. Such a survey could be generalized to many computational systems, though it would need to be tailored to the specific domain in question.

By analyzing the survey results, we have shown that across each of our criteria for creativity, DARCI's artifacts were rated comparably to artifacts produced by humans. Of note, DARCI's images were generally considered more surprising and more difficult to create than their human counterparts. DARCI's performance in the evaluation persisted even when volunteers (shown to be biased against DARCI) were aware of the process used to create the images.

While these results look remarkable on paper, we must note that creativity is still ill-defined and our survey questions are clearly a simplification of what it means to be creative. We must also acknowledge that the artifacts were very specific in nature and the human artists were heavily restricted in there creative process in order to make the comparison to DARCI fair. In a more practical setting, humans would have far fewer restrictions and would undoubtedly produce more interesting images. Finally, we must acknowledge that the four sets of DARCI's artifacts used in the survey were selected from seven sets by a human—though more than half of DARCI's artifacts were included.

Despite these limitations, the results clearly indicate a system capable of performing on par with humans within the restricted domain. These results will also act as a baseline for testing future improvements to the system.

| *blind* | *basic* | *detailed* |
|---|---|---|
| Like | | |
| DARCI 4 “cold” | DARCI 4 “cold” | DARCI 3 “cold” |
| DARCI 3 “cold” | Human 1 “cold” | Human 3 “eerie” |
| Human 1 “cold” | Human 4 “violent” | Human 1 “cold” |
| Human 2 “cold” | DARCI 3 “cold” | Human 4 “violent” |
| Human 4 “cold” | Human 4 “eerie” | Human 2 “cold” |
| DARCI 2 “eerie” | Human 4 “cold” | DARCI 2 “eerie” |
| Adjective | | |
| DARCI 2 “cold” | DARCI 1 “cold” | DARCI 3 “cold” |
| DARCI 4 “cold” | DARCI 3 “cold” | Human 2 “cold” |
| DARCI 3 “cold” | DARCI 2 “cold” | DARCI 2 “cold” |
| DARCI 1 “cold” | Human 2 “eerie” | Human 3 “eerie” |
| Human 2 “violent” | Human 2 “cold” | DARCI 1 “cold” |
| Human 2 “cold” | DARCI 4 “cold” | DARCI 4 “eerie” |
| Surprising | | |
| Human 3 “eerie” | DARCI 2 “violent” | DARCI 1 “violent” |
| DARCI 2 “violent” | DARCI 1 “violent” | Human 3 “eerie” |
| DARCI 1 “violent” | Human 3 “eerie” | DARCI 2 “violent” |
| DARCI 2 “eerie” | DARCI 2 “eerie” | DARCI 1 “cold” |
| DARCI 4 “cold” | DARCI 4 “cold” | Human 2 “violent” |
| Human 2 “violent” | Human 2 “violent” | DARCI 4 “eerie” |
| Difficult | | |
| DARCI 1 “violent” | DARCI 2 “violent” | DARCI 1 “violent” |
| DARCI 2 “violent” | DARCI 1 “violent” | Human 3 “eerie” |
| DARCI 2 “eerie” | Human 3 “eerie” | DARCI 2 “violent” |
| Human 2 “violent” | DARCI 2 “eerie” | Human 2 “violent” |
| Human 3 “eerie” | Human 2 “violent” | DARCI 2 “eerie ” |
| Human 2 “eerie” | DARCI 4 “cold” | DARCI 4 “violent” |
| Use | | |
| Human 3 “eerie” | DARCI 4 “cold” | Human 4 “eerie” |
| Human 4 “eerie” | Human 1 “cold” | Human 1 “cold” |
| DARCI 4 “cold” | Human 4 “violent” | Human 3 “eerie” |
| DARCI 3 “cold” | Human 4 “cold” | DARCI 1 “cold” |
| Human 1 “cold” | DARCI 1 “cold” | DARCI 3 “cold” |
| Human 4 “violent” | DARCI 2 “eerie” | DARCI 1 “eerie” |

Table 2: The top six images (based on Likert rating) for each item across the three experiments. Refer to Figures 4 and 5 to view images.

## References


al Rifaie, M., and Bishop, M. 2012. Weak vs. strong computational creativity, computing, philosophy and the question of bio-machine hybrids. In *Proceedings of the AISB Symposium on Computing and Philosophy*.

Brown, O. 2014. Empirically grounding the evaluation of creative systems: Incorporating interaction design. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S.; Pease, A.; Corneli, J.; Cook, M.; and Llano, T. 2014. Assessing progress in building autonomously creative systems. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. *Creative Intelligent Systems: Papers from the AAAI Spring Symposium* 14–20.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4:246–279.

Jordanous, A. 2014. Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proceedings of the 5th International Conference on Computational Creativity*.

King, I.; Ng, C. H.; and Sia, K. C. 2004. Distributed content-based visual information retrieval system on peer-to-peer network. *ACM Transactions on Information Systems* 22(3):477–501.

Li, C., and Chen, T. 2009. Aesthetic visual quality assessment of paintings. *IEEE Journal of Selected Topics in Signal Processing* 3:236–252.

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.

Machajdik, J., and Hanbury, A. 2010. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the International Conference on Multimedia*, 83–92.

Martínez, H. P.; Yannakakis, G. N.; and Hallam, J. 2014. Dont classify ratings of affect; rank them! *IEEE Transactions on Affective Computing* 5:314–326.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2):106–124.

Norton, D.; Heath, D.; and Ventura, D. 2014. Autonomously managing competing objectives to improve the creation and curation of artifacts. In *Proceedings of the 5$^{th}$ International Conference on Computational Creativity*.

Norton, D.; Heath, D.; and Ventura, D. 2015. Annotating images with emotional adjectives using features that summarize local interest points. *IEEE Transactions on Affective Computing, in submission*.

Ou, L.-C.; Luo, M. R.; Woodcock, A.; and Wright, A. 2004. A study of colour emotion and colour preference. Part I: Colour emotions for single colours. *Color Research and Application* 29:232–240.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Sivic, J., and Zisserman, A. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, 1470–1477.

Wang, W.-N.; Yu, Y.-L.; and Jiang, S.-M. 2006. Image retrieval by emotional semantics: A study of emotional space and feature extraction. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, 3534–3539.

Yannakakis, G. N., and Hallam, J. 2011. Rating vs. preference: A comparative study of self-reporting. *Affective Computing and Intelligent Interaction* 6974:437–446.

# Quantifying Creativity in Art Networks

**Ahmed Elgammal and Babak Saleh**
Department of Computer Science
Rutgers University
{elgammal,babaks}@cs.rutgers.edu
digihumanlab.rutgers.edu

**Abstract**

This paper proposes a computational framework for assessing the creativity of products, such as paintings, sculptures, poetry, etc. The proposed computational framework is based on constructing a network between creative products and using this network to infer about the originality and influence of its nodes. Through a series of transformations, we construct a Creativity Implication Network. We show that inference about creativity in this network reduces to a variant of network centrality problems which can be solved efficiently. We apply the proposed framework to the task of quantifying creativity of paintings (and sculptures). We experimented on two datasets with over 62K paintings to illustrate the behavior of the proposed framework.

## Introduction

The field of computational creativity is focused on giving the machine the ability to generate human-level "creative" products such as computer generated poetry, stories, jokes, music, art, etc., as well as creative problem solving. An important characteristic of a creative agent is its ability to assess its creativity as well as judge other agents' creativity. In this paper we focus on developing a computational framework for assessing the creativity of products, such as painting, sculpture, etc. We use the most common definition of creativity, which emphasizes the originality of the product and its influential value (Paul and Kaufman 2014a). In the next section we justify the use of this definition in contrast to other definitions. The proposed computational framework is based on constructing a network between products and using it to infer about the originality and influence of its nodes. Through a series of transformations, we show that the problem can reduce to a variant of network centrality problems, which can be solved efficiently.

We apply the proposed framework to the task of quantifying creativity of paintings (and sculptures). The reader might question the feasibility, limitation, and usefulness of performing such task by a machine. Artists, art historians and critics use different concepts to describe pantings. In particular, elements of arts such as space, texture, form, shape, color, tone and line. Artists also use principles of art including movement, unity, harmony, variety, balance, contrast, proportion, and pattern; besides brush strokes, subject matter, and other descriptive concepts (Fichner-Rathus 2008). We collectively call these concepts artistic concepts. These artistic concepts can, more or less, be quantified by today's computer vision technology. With the rapid progress in computer vision, more advanced techniques are introduced, which can be used to measure similarity between paintings with respect to a given artistic concept. Whether the state of the art is already sufficient to measure similarity in meaningful ways, or whether this will happen in the near or far future, the goal of this paper is to design a framework that can use such similarity measures to quantify our chosen definition of creativity in an objective way. Hence, the proposed framework would provide a ready-to-use approach that can utilize any future advances in computer vision that might provide better ways for visual quantification of digitized paintings. In fact, we applied the proposed framework using state-of-the-art computer vision techniques and achieved very reasonable automatic quantification of creativity on two large datasets of paintings.

One of the fundamental issues with the problem of quantifying creativity of art is how to validate any results that the algorithm can obtain. Even if art historians would agree on a list of highly original and influential paintings that can be used for validation, any algorithm that aims at assigning creativity scores will encounter three major limitations: I) Closed-world limitation: The algorithm is only limited to the set of paintings it analyzed. It is a closed world for the algorithm where this set is every thing it has seen about art history. The number of images of paintings available in the public domain is just a small fraction of what are in museums and private collections. II) Artistic concept quantification limitation: the algorithm is limited by what it sees, in terms of the ability of the underlying computer vision methods to encode the important elements and principles of art that relates to judging creativity. III) Parameter setting: the results will depend on the setting of the parameters, where each setting would mean a different way to assign creativity scores with different interpretation and different criteria. However, these limitations should not stop us from developing and testing algorithms to quantify creativity. The first two limitations are bound to disappear in the future, with more and more paintings being digitized, as well as with the continuing advances in computer vision and machine learning. The third limitation should be thought of as an advan-

tage, since the different settings mean a rich ability of the algorithm to assign creativity scores based on different criteria. For the purpose of validation, we propose a methodology for validating the results of the algorithm through what we denote as "time machine experiments", which provides evidence of the correctness of the algorithm.

Having discussed the feasibility and limitations, let us discuss the value of using any computational framework to assess creativity in art. For a detailed discussion about the implications of using computational methods in the domain of aesthetic-judgment-related tasks, we refer the reader to (Spratt and Elgammal 2014). Our goal is not to replace art historians' or artists' role in judging creativity of art products. Providing a computational tool that can process millions of artworks to provide objective similarity measures and assessments of creativity, given certain visual criteria can be useful in the age of digital humanities. From a computational creativity point of view, evaluating the framework on digitized art data provides an excellent way to optimize and validate the framework, since art history provides us with suggestions about what is considered creative and what might be less creative. In this work we did not use any such hints in achieving the creativity scores, since the whole process is unsupervised, i.e., the approach does not use any creativity, genre, or style labels. However we can use evidence from art history to judge whether the results make sense or not. Validating the framework on digitized art data makes it possible to be used on other products where no such knowledge is available, for example to validate computer-generated creative products.

## On the Notion of Creativity

There is a historically long and ongoing debate on how to define creativity. In this section we give a brief description of some of these definitions that directly relate to the notion we will use in the proposed computational framework. Therefore, this section is by no means intended to serve as a comprehensive overview of the subject. We refer readers to (Taylor 1988; Paul and Kaufman 2014b) for comprehensive overviews of the different definitions of creativity.

We can describe a person (e.g. artist, poet), a product (painting, poem), or the mental process as being creative (Taylor 1988; Paul and Kaufman 2014a). Among the various definitions of creativity it seems that there is a convergence to two main conditions for a product to be called "creative". That product must be novel, compared to prior work, and also has to be of value or influential (Paul and Kaufman 2014a). These criteria resonate with Kant's definition of artistic genius, which emphasizes two conditions "originality" and being "exemplary" [1]. Psychologists would not totally agree with this definition since they favor associating creativity with the mental process that generates the product (Taylor 1988; Nanay 2014). However associating creativity with products makes it possible to argue in favor of "Computational Creativity", since otherwise, any computer product would be an output of an algorithmic process and not a result of a creative process. Hence, in this paper we stick to quantifying the creativity of products instead of the mental process that create the product.

Boden suggested a distinction between two notions of creativity: psychological creativity (P-creativity), which assesses novelty of ideas with respect to its creator, and historical creativity (H-creativity), which assesses novelty with respect to the whole human history (Boden 1990). It follows that P-creativity is a necessary but not sufficient condition for H-creativity, while H-creativity implies P-creativity (Boden 1990; Nanay 2014). This distinction is related to the subjective (related to person) vs. objective creativity (related to the product) suggested by Jarvie (Jarvie 1986). In this paper our definition of creativity is aligned with objective/H-creativity, since we mainly quantify creativity within a historical context.

## Computational Framework

According to the discussion in the previous section, a creative product must be *original*, compared to prior work, and valuable (*influential*) moving forward. Let us construct a network of creative products and use it to assign a creativity score to each product in the network according to the aforementioned criteria. In this section, for simplicity and without loss of generality, we describe the approach based on a network of paintings, however the framework is applicable to other art or literature forms.

### Constructing a Painting Graph

Let us denote by $P = \{p_i, i = 1 \cdots N\}$ a set of paintings. The goal is to assign a creativity score for each painting, denoted by $C(p_i)$ for painting $p_i$ . Every painting comes with a time label indicating the date it was created, denoted by $t(p_i)$. We create a directed graph where each vertex corresponds to a painting. A directed edge (arc) connects painting $p_i$ to $p_j$ if $p_i$ was created before $p_j$. Each directed edge is assigned a positive weight (we will discuss later where the weights come from), we denote the weight of edge $(p_i, p_j)$ by $w_{ij}$. We denote by $W_{ij}$ the adjacency matrix of the painting graph, where $W_{ij} = w_{ij}$ if there is an edge from $p_i$ to $p_j$ and 0 otherwise. Note that according to this definition, a painting is not connected to itself, i.e., $w_{ii} = 0, i = 1 \cdots N$. By construction, $w_{ij} > 0 \rightarrow w_{ji} = 0$, i.e., the graph is anti-symmetric.

To assign the weights we assume that there is a similarity function that takes two paintings and produces a positive scalar measure of affinity between them (higher value indicates higher similarity). We denote such a function by $S(\cdot, \cdot)$

[1] Among four criteria for artistic genius suggested by Kant, two describe the characteristic of a creative product "That genius 1) is a talent for producing that for which no determinate rule can be given, not a predisposition of skill for that which can be learned in accordance with some rule, consequently that originality must be it's primary characteristic. 2) that since there can also be original nonsense, its products must at the same time be models, i.e., exemplary, hence, while not themselves the result of imitation, they must yet serve others in that way, i.e., as a standard or rule for judging." (Guyer and Wood 2000)-p186

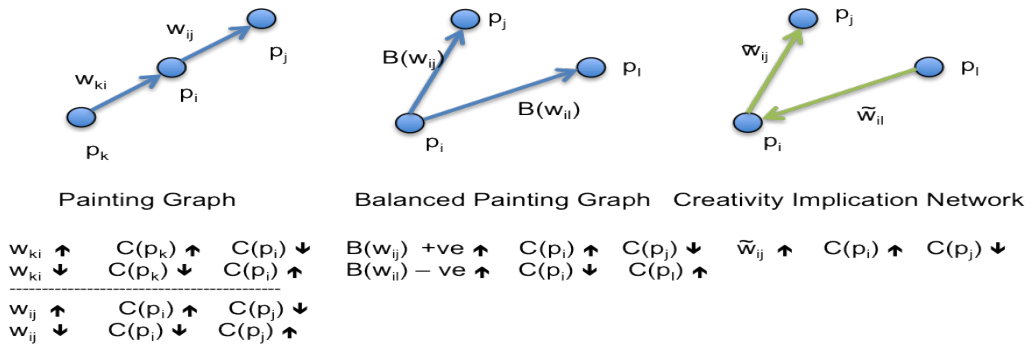



Figure 1: Illustration of the construction of the Creativity Implication Network: blue arrows indicate temporal relation and orange arrows indicate reverse creativity implication (converse).

and, therefore,

$$w_{ij} = \begin{cases} S(p_i, p_j) & \text{if } t(p_i) < t(p_j). \\ 0 & \text{otherwise.} \end{cases}$$

Since there are multiple possible visual aspects that can be used to measure similarity, we denote such a function by $S^a(\cdot,\cdot)$ where the superscript $a$ indicates the visual aspect that is used to measure the similarity (color, subject matter, brush stroke, etc.) This implies that we can construct multiple graphs, one for each similarity function. We denote the corresponding adjacency matrix by $W^a$, and the induced creativity score by $C^a$, which measure the creativity along the dimension of visual aspect $a$. In the rest of this section, for the sake of simplicity, we will assume one similarity function and drop the superscript. Details about the similarity function will be explained in the next section.

## Creativity Propagation

Giving the constructed painting graph, how can we propagate the creativity in such a network? To answer this question we need to understand the implication of the weight of the directed edge connecting two nodes on their creativity scores. Let us assume that initially we assign equal creativity indices to all nodes. Consider painting $p_i$ and consider an incoming edge from a prior painting $p_k$. A high weight on that edge ($w_{ki}$) indicates a high similarity between $p_i$ and $p_k$, which indicates that $p_i$ is not novel, implying that we should lower the creativity score of $p_i$ (since $p_i$ is subsequent to $p_k$ and similar to it) and increase the creativity score of $p_k$. In contrast, a low weight implies that $p_i$ is novel and hence creative compared to $p_k$, therefore we need to increase the creativity score of $p_i$ and decreases that of $p_k$.

Let us now consider the outgoing edges from $p_i$. According to our notion of creativity, for $p_i$ to be creative it is not enough to be novel, it has to be influential as well (some others have to imitate it). This indicates that a high weight, $w_{ij}$, between $p_i$ and a subsequent painting $p_j$ implies that we should increase the creativity score of $p_i$ and decrease that of $p_j$. In contrast, a lower weight implies that $p_i$ is not influential on $p_j$, and hence we should decrease the score for $p_i$ and increase it for $p_j$. These four cases are illustrated in Figure 1. A careful look reveals that the two cases for the incoming edges and those for the outgoing edges are in fact the same. *A higher weight implies the prior node is more influential and the subsequent node is less creative, and a lower weight implies the prior node is less influential and the subsequent node is more creative.*

## Creativity Implication Network

Before converting this intuition to a computational approach, we need to define what is considered high and low for weights. We introduce a balancing function on the graph. Let $m(i)$ denote a balancing value for node $i$, where for the edges connected to that node a weight above $m(i)$ is considered high and below that value is considered low. We define a balancing function as a linear function on the weights connecting to each node in the form

$$B_i(w) = \begin{cases} w - m(i) & \text{if } w > 0. \\ 0 & \text{otherwise.} \end{cases}$$

We can think of different forms of balancing functions that can be used. Also there are different ways to set the parameter $m(i)$ with different implications, which we will discuss in the next section. This form of balancing function basically converts weights lower than $m(i)$ to negative values. The more negative the weight of an edge the more creative the subsequent node and the less influential the prior node. The more positive the weight of an edge the less creative the subsequent node and the more influential the prior node.

The introduction of the negative weights in the graph, despite providing a solution to represent low weights, is problematic when propagating the creativity scores. The intuition is, a negative edge between $p_i$ and $p_j$ is equivalent to a positive edge between $p_j$ and $p_i$. This directly suggests that we should reverse all negative edges and negate their values. Notice that the original graph construction guarantees that an edge between $p_i$ and $p_j$ implies no edge between $p_j$ and $p_i$, therefore there is no problem with edge reversal. This process results in what we call *"Creativity Implication Network"*. We denote the weights of that graph by $\tilde{w}_{ij}$ and its adjacency matrix by $\widetilde{W}$. This process can be described

mathematically as

$$\begin{aligned} B(w_{ij}) > 0 &\rightarrow \tilde{w}_{ij} = B(w_{ij}) \\ B(w_{ij}) = 0 &\rightarrow \tilde{w}_{ij} = 0 \\ B(w_{ij}) < 0 &\rightarrow \tilde{w}_{ji} = -B(w_{ij}) \end{aligned}$$

The Creativity Implication Network has one simple rule that relates its weights to creativity propagation: *the higher the weight of an edge between two nodes, the less creative the subsequent node and the more creative the prior node.* Note that the direction of the edges in this graph is no longer related to the temporal relation between its nodes, instead it is directly inverse to the way creativity scores should propagate from one painting to another. Notice that the weights of this graph are non-negative.

## Computing Creativity Scores

Given the construction of the Creativity Implication Network, we are now ready to define a recursive formula for assigning creativity scores. We will show that the construction of the Creativity Implication Network reduces the problem of computing the creativity scores to a traditional network centrality problem. The algorithm will maintain creativity scores that sum up to one, i.e., the creativity scores form a probability distribution over all the paintings in our set. Given an initial equal creativity scores, the creativity score of node $p_i$ should be updated as

$$C(p_i) = \frac{(1-\alpha)}{N} + \alpha \sum_j \tilde{w}_{ij} \frac{C(p_j)}{N(p_j)}, \tag{1}$$

where $0 \leq \alpha \leq 1$ and $N(p_j) = \sum_k \tilde{w}_{kj}$. In this formula, the creativity of node $p_i$ is computed from aggregating a fraction $\alpha$ of the creativity scores from its outgoing edges weighted by the adjusted weights $\tilde{w}_{ij}$. The constant term $(1-\alpha)/N$ reflects the chance that similarity between two paintings might not necessarily indicate that the subsequent one is influenced by the prior one. For example, two paintings might be similar simply because they follow a certain style or art movement. The factor $1-\alpha$ reflects the probability of this chance. The normalization term $N(p_j)$ for node $j$ is the sum of its incoming weights, which means that the contribution of node $p_j$ is split among all its incoming nodes based on the weights, and hence, $p_i$ will collect only a fraction $\tilde{w}_{ij}/\sum_k \tilde{w}_{kj}$ of the creativity score of $p_j$.

The recursive formula in Eq 1 can be written in a matrix form as

$$C = \frac{(1-\alpha)}{N}\mathbf{1} + \alpha\widetilde{\widetilde{W}}C, \tag{2}$$

where $\widetilde{\widetilde{W}}$ is a column stochastic matrix defined as $\widetilde{\widetilde{W}}_{ij} = \tilde{w}_{ij}/\sum_k \tilde{w}_{kj}$, and $\mathbf{1}$ is a vector of ones of the same size as $C$. It is easy to see that since $\widetilde{\widetilde{W}}$, $C$, and $\frac{1}{N}\mathbf{1}$ are all column stochastic, the resulting scores will always sum up to one. The creativity scores can be obtained by iterating over Eq 2 until conversion. Also a closed-form solution for the case where $\alpha \neq 1$ can be obtained as

$$C^* = \frac{(1-\alpha)}{N}(I - \alpha\widetilde{\widetilde{W}})^{-1}\mathbf{1}. \tag{3}$$

A reader who is familiar with social network analysis literature might directly see the relation between this formulation and some traditional network centrality algorithms. Eq 2 represents a random walk in a Markov chain. Setting $\alpha = 1$, the formula in Eq 2 becomes a weighted variant to eigenvector centrality (Borgatti and Everett 2006), where a solution can be obtained by the right eigenvector corresponding to the largest eigenvalue of $\widetilde{\widetilde{W}}$. The formulation in Eq 2 is also a weighted variant of Hubbell's centrality (Hubbell 1965). Finally the formulation can be seen as an inverted weighted variant of the Page Rank algorithm (Brin and Page 1998). Notice that this reduction to traditional network centrality formulations was only possible because of the way the Creativity Implication Network was constructed.

## Originality vs. Influence

The formulation above sums up the two criteria of creativity, being original and being influential. We can modify the formulation to make it possible to give more emphasis to either of these two aspects when computing the creativity scores. For example it might be desirable to emphasize novel works even though they are not influential, or the other way around. Recall that the direction of the edges in Creativity Implication Network are no longer related to the temporal relation between the nodes. We can label (color) the edges in the network such that each outgoing edge $e(p_i, p_j)$ from a given node $p_i$ is either labeled as a subsequent edge or a prior edge depending on the temporal relation between $p_i$ and $p_j$. This can be achieved by defining two disjoint subsets of the edges in the networks

$$\begin{aligned} E^{\text{prior}} &= \{e(p_i,p_j) : t(p_j) < t(p_i)\} \\ E^{\text{subseq}} &= \{e(p_i,p_j) : t(p_j) \geq t(p_i)\} \end{aligned}$$

This results in two adjacency matrices, denoted by $\widetilde{W}^p$ and $\widetilde{W}^s$ such that $\widetilde{W} = \widetilde{W}^p + \widetilde{W}^s$, where the superscripts $p$ and $s$ denote the prior and subsequent edges respectively. Now Eq 1 can be rewritten as

$$\begin{aligned} C(p_i) &= \frac{(1-\alpha)}{N} + \\ &\alpha[\beta \sum_j \tilde{w}^p_{ij} \frac{C(p_j)}{N^p(p_j)} + (1-\beta)\sum_j \tilde{w}^s_{ij}\frac{C(p_j)}{N^s(p_j)}], \end{aligned} \tag{4}$$

where $N^p(p_j) = \sum_k \tilde{w}^p_{kj}$ and $N^s(p_j) = \sum_k \tilde{w}^s_{kj}$. The first summation collects the creativity scores stemming from prior nodes, i.e., encodes the originality part of the score, while the second summation collects creativity scores stemming from subsequent nodes, i.e, encodes influence. We introduced a parameter $0 \leq \beta \leq 1$ to control the effect of the two criteria on the result. The modified formulation above can be written as

$$C = \frac{(1-\alpha)}{N}\mathbf{1} + \alpha[\beta\widetilde{\widetilde{W^p}}C + (1-\beta)\widetilde{\widetilde{W^s}}C], \tag{5}$$

where $\widetilde{\widetilde{W^p}}$ and $\widetilde{\widetilde{W^s}}$ are the column stochastic adjacency matrices resulting from normalizing the columns of $\widetilde{W}^p$ and $\widetilde{W}^s$ respectively. It is obvious that the closed-form solution

in Eq 3 is applicable to this modified formulation where $\widetilde{\widetilde{W}}$ is defined as $\widetilde{\widetilde{W}} = \beta\widetilde{\widetilde{W^p}} + (1-\beta)\widetilde{\widetilde{W^s}}$.

## Creativity Network for Art

In this section we explain how the framework can be realized for the particular case of visual art.

*Visual Likelihood:* For each painting we can use computer vision techniques to obtain different feature representations for its image, each encoding a specific visual aspect(s) related to the elements and principles of arts. We denote such features by $f_i^a$ for painting $p_i$, where $a$ denotes the visual aspect that the feature quantifies. We define the similarity between painting $p_i$ and $p_j$, as the likelihood that painting $p_j$ is coming from a probability model defined by painting $p_i$. In particular, we assume a Gaussian probability density model for painting $p_i$, i.e.,

$$S^a(p_j, p_i) = Pr(p_j|p_i, a) = \mathcal{N}(\cdot; f_i^a, \sigma^a I).$$

It is important to limit the connections coming to a given painting. By construction, any painting will be connected to all prior paintings in the graph. This makes the graph highly biased since modern paintings will have extensive incoming connections and early paintings will have extensive outgoing connections. Therefore we limit the incoming connections to any node to at most the top $K$ edges (the $K$ most similar prior paintings).

*Temporal Prior:* It might be desirable to add a temporal prior on the connections. If a painting in the nineteenth century resembles a painting from the fourteenth century, we shouldn't necessarily penalize that as low creativity. This is because certain styles are always reinventions of older styles, for example neoclassicism and renaissance. Therefore, these similarities between styles across distant time periods should not be considered as low creativity. Therefore, we can add a temporal prior to the likelihood as

$$S^a(p_j, p_i) = Pr^v(p_j|p_i, a) \cdot Pr^t(p_j|p_i),$$

where the second probability is a temporal likelihood (what is the likelihood that $p_j$ is influenced $p_i$ given their dates) and the first is the visual likelihood. There are different ways to define such a temporal likelihood. The simplest way is a temporal window function, i.e., $Pr^t(p_j|p_i) = 1$ if $p_i$ is within K temporal neighbors prior to $p_j$ and 0 otherwise[2].

*Balancing Function:* There are different choices for the balancing function $B(w)$, as well as the parameter for that function. We mainly used a linear function for that purpose. The parameter $m$ can be set globally over the whole graph, or locally for each time period. A global $m$ can be set as the p-percentile of the weights of the graph, which is p-percentile of all the pairwise likelihoods. This directly means that p% of the edges of the graph will be reversed when constructing the Creativity Implication Graph.

[2]Alternatively, a Gaussian density can be use, $Pr^t(p_j|p_i) = \exp(-[t(p_i)-t(p_j)]^2/\sigma_t^2)$. However, adding such temporal Gaussian would complicate the algorithm since it will not be easy to estimate a suitable $\sigma_t$, specially the graph can have non-uniform density over the time line.

One disadvantage of a global balancing function is that different time periods have different distributions of weights. This suggests using a local-in-time balancing function. To achieve that we compute $m_i$ for each node as p% of the weight distribution based on its temporal neighborhood.

## Experiments and Results

### Datasets and Visual Features

*Artchive:* This dataset was previously used for style classification and influence discovery (Saleh et al. 2014). It contains a total of 1710 images of art works (paintings and sculptures) by 66 artists, from 13 different styles from 1412-1996, chosen from Mark Harden's Artchive database of fine-art (Harden ). The majority of the images are of the full work, while a few are details of the work.

*Wikiart.org:* We used the publicly available dataset of *"Wikiart paintings"*[3]; which, to the best of our knowledge, is the largest online public collection of artworks. This collection has images of 81,449 fine-art paintings and sculptures from 1,119 artist spanning from 1400-2000+. These paintings are from 27 different styles (Abstract, Byzantine, Baroque, etc.) and 45 different genres (Interior, Landscape, Portrait, etc.). We pruned the dataset to 62,254 western paintings by removing genres and mediums that are not suitable for the analysis such as sculpture, graffiti, mosaic, installation, performance, photos, etc.

For both datasets the time annotation is mainly the year. Therefore, it is not possible to tell which is prior between any pair of paintings with the same year of creation. Therefore no edge is added between their corresponding nodes.

We experimented with different state-of-the-art feature representations. In particular, the results shown here are using Classeme features (Torresani, Szummer, and Fitzgibbon 2010). These features were shown to outperform other state-of-the-art features for the task of style classification (Saleh et al. 2014). These features (2659 dimensions) provide semantic-level representation of images, by encoding the presence of a set of basic-level object categories (e.g. horse, cross, etc.), which captures the subject matter of the painting. Some of the low-level features used to learn the Classeme features also capture the composition of the scene.

### Example Results

We show qualitative and quantitative experimental results of the framework applied to the aforementioned datasets. As mentioned in the introduction, any result has to be evaluated given the set of paintings available to the algorithm and the capabilities of the visual features used. Given that the visual features used are mainly capturing subject matter and composition, sensible creativity scores are expected to reflect these concept. A low creativity score does not mean that the work is not creative in general, it just means that the algorithm does not see it creative with respect to its encoding of subject matter and composition.

Figures 2-top and 3 show the creativity scores obtained on the Artchive and Wikiart datasets respectively. Figure 2-bottom shows a zoom in to the period between 1850-1950 in

[3]http://www.wikiart.org/

the Artchive dataset, which is very dense in the graph[4]. In all figures we plot the scores vs. the year of the painting. The figures visualize some of the paintings that obtained high scores, as well as some with low scores (the scores in the plots are scaled). We randomly sampled points with low scores for visualization. A close look at the paintings that scored low (bottom) reveals the presence of typical subject matter that is common in the dataset, or in some cases the image presents an unclear view of a sculpture (e.g. Rodin 1889 sculpture in the bottom right). The general trend shows peaks in creativity around the time of High Renaissance (late 15th , early 16th century) and the late 19th and early 20th centuries, and a significant increase in the second half of the 20th century.

One of the interesting findings is the algorithm's ability to point out wrong annotations in the dataset. For example, one of the highest scoring paintings around 1910 was a painting by Piet Mondrain called " Composition en blanc, rouge et jaune," (see Figure 2). By examining this painting, we found that the correct date for it is around 1936 and it was mistakenly annotated in the Artchive dataset as 1910 [5]. Modrain did not start to paint in this grid-based (Tableau) style untill around 1920. So it is no surprise that wrongly dating one of Mondrain's tableau paintings to 1910 caused it to obtain high creativity score, even above the cubism paintings from that time. On the Wikiart dataset, one of the highest-scored painting was "tornado" by contemporary artist Joe Goode, which was found to be mistakenly dated 1911 in Wikiart [6]. A closer look at the artist biography revealed that he was born in 1937 and this painting was created in 1991[7]. It is not surprising for a painting that was created in 1991 to score very high in creativity if it was wrongly dated to 1911. These two examples, besides indicating that the algorithm works, show the potential of proposed algorithm in in spotting wrong annotations in large datasets, which otherwise would require tremendous human effort.

## Time Machine Experiment

Given the absence of ground truth for creativity, the aforementioned wrong annotations inspired us with a methodology to quantitatively evaluate the framework. We designed what we call "time machine" experiment, where we change the date of an artwork to some point in the past or some point in the future, relative to its correct time of creation. Then we compute the creativity scores using the wrong date, by running the algorithm on the whole data. We then compute the gain (or loss) in the creativity score of that artwork compared to its score using correct dating. What should we expect from an algorithm that assigns creativity in a sensible way? Moving a creative painting back in history would increase its creativity score, while moving a painting forward would decrease its creativity. Therefore, we tested three settings: I) Moving back to AD 1600: For styles that date after 1750, we set the test paintings back to a random date around 1600 using Normal distribution with mean 1600 and std 50 years, i.e. $N(1600, 50^2)$ . II) Moving forward to AD 1900: For the Renaissance and Baroque styles, we set the test paintings to random dates around 1900 sampled from $N(1900, 50^2)$. III) Wandering about AD 1600 (baseline): In this experiment, for the Renaissance and Baroque styles, we set the test paintings to random dates around 1600 sampled from $N(1600, 50^2)$.

Table 1: Time Machine Experiment

| Art movement | avg % gain/loss | % increase |
|---|---|---|
| Moving backward to AD 1600 | | |
| Neoclassicism | 5.78%±1.28 | 97%±4.8 |
| Romanticism | 7.52%± 2.04 | 98%± 4.2 |
| Impressionism | 14.66%± 2.78 | 99%±3.2 |
| Post-Impressionism | 16.82%±2.22 | 99%±3.1 |
| Symbolism | 15.2%±2.94 | 97%±4.8 |
| Expressionism | 16.83%±2.43 | 98%±4.2 |
| Cubism | 13.36%±2.43 | 89%±9.9 |
| Surrealism | 12.66%±1.82 | 95%±7.1 |
| American Modernism | 11.75%±2.99 | 84%±8.4 |
| Wandering around to AD 1600 | | |
| Renaissance | 0.68 %± 2.05 | 39%±5.7 |
| Baroque | 2.85%± 1.09 | 71%±19.7 |
| Moving forward to AD 1900 | | |
| Renaissance | -8.13%± 2.02 | 20%±10.5 |
| Baroque | -10.2%±2.03 | 0%±0 |

Table 1 shows the results of these experiments. We ran this experiment on the Artchive dataset with no temporal prior. In each run we randomly selected 10 test paintings of a given style and applied the corresponding move. We used 10 as a small percentage of the data set (less than 1%), not to disturb the global distribution of creativity. We repeated each experiment 10 times and reported the mean and standard deviations of the runs. For each style we computed the average gain/loss of creativity scores by the time move. We also computed the percentage of the test paintings whose scores have increased. From the table we clearly see that paintings from Impressionist, Post-Impressionist, Expressionist, and Cubism movements have significant gain in their creativity scores when moved back to 1600. In contrast, Neoclassicism paintings have the least gain, which makes sense, because Neoclassicism can be considered as revival to Renaissance. Romanticism paintings also have a low gain when moved back to 1600, which is justified because of the connection between Romanticism and Gothicism and Medievalism. On the other hand, paintings from Renaissance and Baroque styles have loss in their scores when moved forward to 1900, while they did not change much in the wandering-around-1600 setting.

[4]For Figure 2 a temporal window historical prior is uses. For Figure 3 no historical prior was used. For both, we set K=500,$\alpha$=0.15

[5]The wrong annotation is in the Artchive CD obtained in 2010. The current online version of Artchive has corrected annotation for this painting

[6]http://www.wikiart.org/en/search/tornado/1#supersized-search-318512 - accessed on Feb 28th, 2015

[7]http://www.artnet.com/artists/joe-goode/tornado-9-2Y7erPME95YlkhFp7DRWlA2

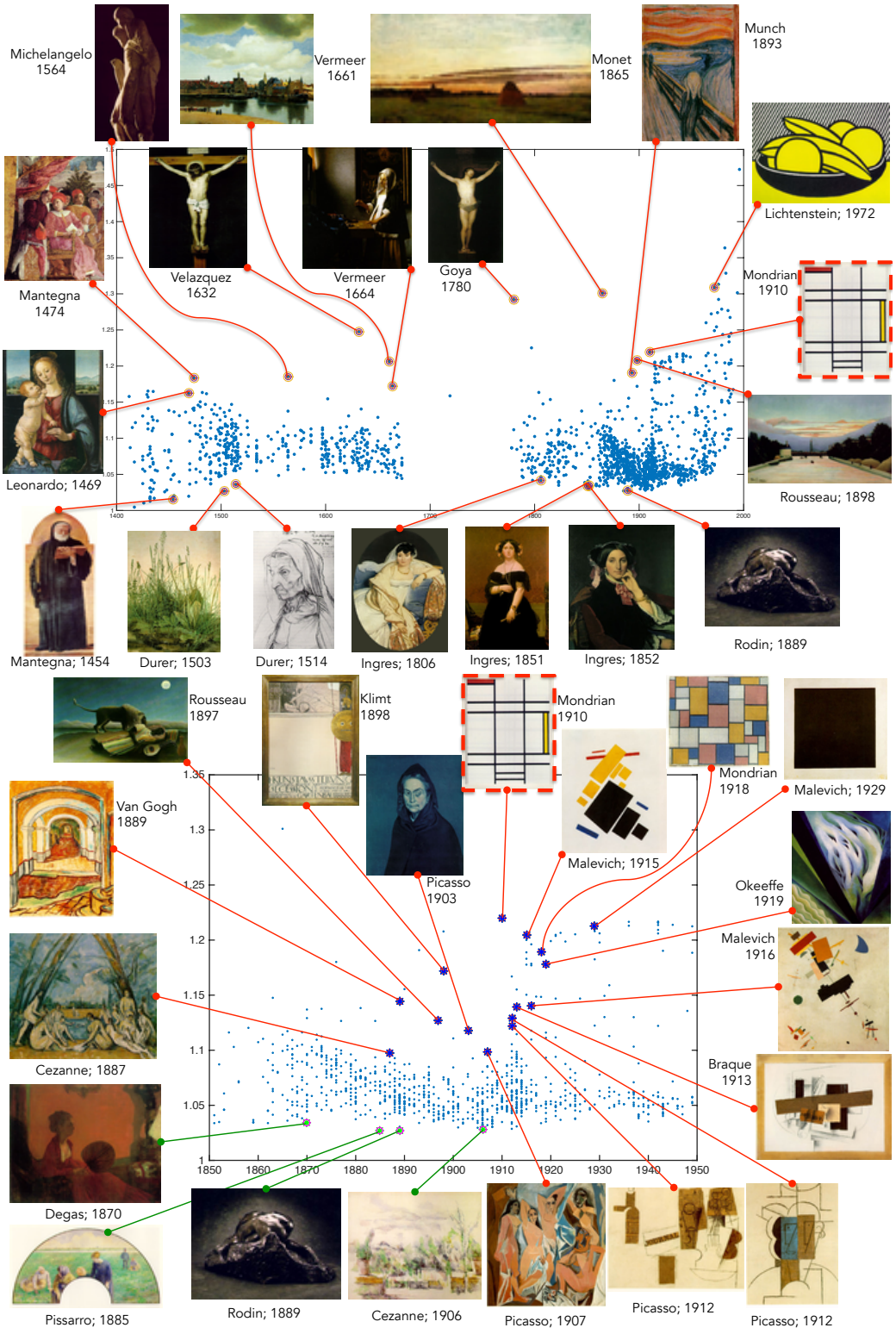



Figure 2: Top: Creativity scores for 1710 paintings from Artchive dataset. Bottom: zoom in to the period 1850-1950. Each point represents a painting. The thumbnails illustrate some of the paintings that scored relatively high or low compared to their neighbors. Only artist names and dates of the paintings are shown on the graph because of limited space. The red-dotted-framed painting by Piet Mondrain scored very high because it was wrongly dated to 1910 instead of 1936 in the dataset.

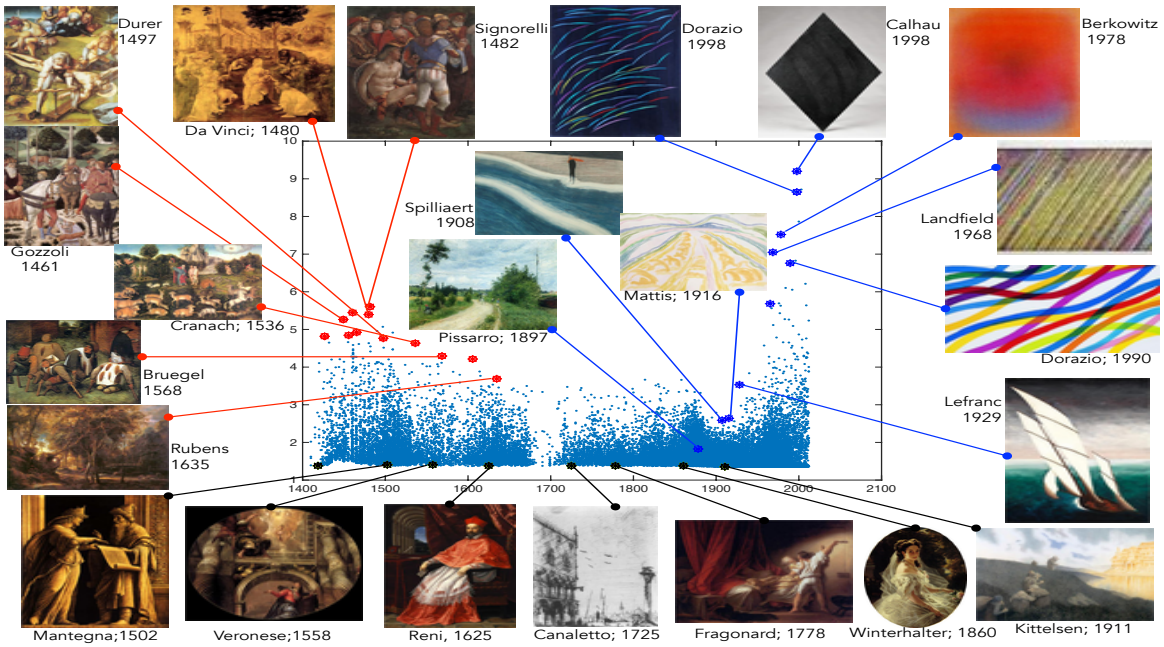



Figure 3: Creativity scores for 62K paintings from the wikiart.org dataset

## Conclusion and Discusion

The paper presented a computational framework to assess creativity among a set of products. We showed that, by constructing a creativity implication network, the problem reduces to a traditional network centrality problem. We realized the framework for the domain of visual art, where we used computer vision to quantify similarity between artworks. We validated the approach qualitatively and quantitively on two large datasets.

In this paper we focused on "creative" as an attribute of a product, in particular artistic products such as painting, where creativity of a painting is defined as the level of its originality and influence. However, the computational framework can be applied to other forms such as sculpture, literature, science etc. Quantifying creativity as an attribute of a product facilitates quantifying the creativity of the person who made that product, as a function over the creator's set of products. Hence, our proposed framework also serves as a way to quantify creativity as an attribute for people.

## References


Boden, M. A. 1990. *The creative mind: Myths and mechanisms*. Basic Books.

Borgatti, S. P., and Everett, M. G. 2006. A graph-theoretic perspective on centrality. *Social networks* 28(4):466–484.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1):107–117.

Fichner-Rathus, L. 2008. *Foundations of Art and Design*. Clark Baxter.

Guyer, P., and Wood, A. W. 2000. *Critique of the Power of Judgement. The Cambridge Edition of the Works of Immanuel Kant*. Cambridge University Press.

Harden, M. The artchive@http://artchive.com/cdrom.htm.

Hubbell, C. H. 1965. An input-output approach to clique identification. *Sociometry* 377–399.

Jarvie, I. 1986. The rationality of creativity. In *Thinking about Society: Theory and Practice*. Springer. 282–301.

Nanay, B. 2014. An experiential account of creativity. In Paul, E. S., and Kaufman, S. B., eds., *The Philosophy of Creativity: New Essays*. Oxford University Press.

Paul, E. S., and Kaufman, S. B. 2014a. Introducing the philosophy of creativity. In *The Philosophy of Creativity: New Essays*. Oxford University Press.

Paul, E. S., and Kaufman, S. B. 2014b. *The Philosophy of Creativity: New Essays*. Oxford University Press.

Saleh, B.; Abe, K.; Arora, R. S.; and Elgammal, A. 2014. Toward automated discovery of artistic influence. *Multimedia Tools and Applications - Special Issue on Multimedia for Cultural Heritage*.

Spratt, E. L., and Elgammal, A. 2014. Computational beauty: Aesthetic judgment at the intersection of art and science. In *ECCV 2014 Workshops, Part I, Proceedings of when vision meets art (VisArt) workshop, Lecture Notes on Computer Science number 8925*. Springer.

Taylor, C. W. 1988. Various approaches to and definitions of creativity. *The nature of creativity* 99–121.

Torresani, L.; Szummer, M.; and Fitzgibbon, A. 2010. Efficient object category recognition using classemes. In *ECCV*.

# Is Biologically Inspired Invention Different?

Ashok K. Goel
Design & Intelligence Laboratory, School of Interactive Computing, and
Center for Biologically Inspired Design, Georgia Institute of Technology
goel@cc.gatech.edu

**Abstract:** The paradigm of biologically inspired design views nature as a vast library of robust, efficient and multifunctional designs, and espouses the use nature as a source of analogues for inspiring novel designs in domains of interest such as architecture, computing, engineering, etc. Over the last generation, biologically inspired design has emerged as a major movement in engineering, architectural, and systems design, pulled in part by the need for environmentally sustainable design and pushed partly by the desire for creativity and innovation in design. An important question is whether biologically inspired design is fundamentally different from other kinds of analogy-based creative processes. This question is critical because the computational theories, techniques and tools we need to develop to support biologically inspired design depend on the nature of the task itself. In this paper, we first summarize some of our empirical findings about biologically inspired design, then derive a task model for it, and finally posit that biologically inspired design indeed is a novel methodology for multiple reasons.

## Biologically Inspired Design

The paradigm of biologically inspired design (also known as biomimicry, biomimetics and bionics) views nature as a vast library of robust, efficient and multifunctional designs, and espouses the use of nature as an analogue for designing technological systems as well as a standard for evaluating technological designs (Benyus 1997; French 1994; Gleich et. al. 2010; Turner 2007; Vincent & Mann 2002; Vogel 2000). This paradigm has inspired many famous designers in the history of design including Leonardo da Vinci, and in a wide variety of design domains ranging from architecture to computing to engineering to systems. However, over the last generation the paradigm has become a movement in modern design, pulled in part by the growing need for environmentally sustainable development and pushed partly by the desire for creativity and innovation in design. Thus, the study of biologically inspired design is attracting a rapidly growing literature, including patents (Bonser & Vincent 2007), publications (Lepora et al. 2013), and computational tools (Goel, McAdams & Stone 2014).

The Biomimicry Institute (2011) provides numerous examples of biologically inspired design. The design of windmill turbine blades mimicking the design of tubercles on the pectoral flippers of humpback whales is one example of biologically inspired design. As Figure 1 illustrates, tubercles are large bumps on the leading edges of humpback whale flippers that create even, fast-moving channels of water flowing over them. The whales thus can move through the water at sharper angles and turn tighter corners than if their flippers were smooth (Fish et al. 2011). When applied to wind turbine blades, they improve lift and reduce drag, improving the energy efficiency of the turbine.

**Figure 1: Design of windmill turbine blades to increase efficiency inspired by the tubercles on humpback whale flippers. (**The Biomimicry Institute 2011)

From the perspective of computational creativity, two characteristics of biologically inspired design are especially noteworthy. Firstly, biologically inspired design often is creative: its products, such as the windmill turbine blades illustrated in Figure 1, are novel, valuable, feasible, and non-obvious (even surprising at first). Secondly, the conceptual phase of biologically inspired design engages analogical transfer of knowledge from biological analogues to design problems in the domain of interest. The latter point raises an important question: is biologically inspired design fundamentally different from other kinds of analogy-based creative processes other than the obvious fact the source domain here is biology? This question is important because the computational theories, techniques and tools we need to develop to support biologically inspired design depend on the nature of the task. For example, Nagle (2014) describes an engineering-to-biology thesaurus that maps function terms used in engineering into equivalent function terms used in biology. The (implicit) assumption in the work on the engineering-to-biology thesaurus is that biologically inspired design is not very different from other analogy-based creative processes (e.g., Veale 2003), that if we could only bridge the vocabulary

gap between design and biology, we could borrow the rest from extant theories of design, analogy and creativity.

In this paper, we first summarize some of our empirical findings about biologically inspired design and then derive a Task Model for it. Finally, we will posit that biologically inspired design is a novel methodology for multiple reasons, and thus requires the development of new computational theories, techniques and tools.

## Research Methodology

Theories of biologically inspired design process can be *normative and prescriptive* or *descriptive and explanatory*. Vincent's et al.'s (2006) BioTRIZ theory, for example, is a normative and prescriptive account of biologically inspired design. In contrast, we have developed a descriptive and explanatory account. Thus, our research methodology consists of three major elements: *In situ* observations of biologically inspired design practices, task analysis of biologically inspired design, and comparison with current theories of design, analogy and creativity.

**Observations of Biologically Inspired Design Practices:** Given that the professional biologically inspired design community at present is nascent, sparse and diffused, we studied biologically inspired design practices in the Georgia Tech ME/ISyE/MSE/BME/BIOL 4740 course from 2006 through 2013 taken by ~350 students. This a yearly, interdisciplinary, project-based course on biologically inspired design taught jointly by biology and engineering faculty. The class is composed of mostly senior-level undergraduate students from biology, biomedical engineering, industrial design, industrial engineering, mechanical engineering, and a variety of other disciplines. Although it evolves a little every year, the course is consistently structured around lectures, found object exercises, journal entries, and one or more design projects. Some lectures discuss biological systems; some lectures focus on case studies of biologically inspired design; and some lectures formulate, analyze and critique problems for students to solve in small groups. Yen et al. (2011, 2014) provide a detailed account of the teaching and learning in the course.

**Task Analysis of Biologically Inspired Design:** Given our observations in the ME/ISyE/MSE/BME/BIOL 4740 classes from 2006 through 2013, we conducted a task analysis of the macrostructure of biologically inspired design practices. Crandall, Klein & Hoffman (2006) describe the methodology of task analysis in detail. Task analysis helps identify the task decomposition of a complex task, the methods used to accomplish the various subtasks in the task decomposition, and the contents of knowledge used by the different methods. For example, Chandrasekaran (1990) presents a high-level task analysis of the general design task while Goel & Chandrasekaran (1992) present a task analysis of the specific method of case-based design. In general, task analysis may describe the behaviors of an individual designer, the interactions among a team of designers, or the behaviors of a design team viewed as a unit. Although we are interested in all three levels of aggregation, in this work we focus on interdisciplinary design teams of biologists and engineers viewed as the unit of analysis. Our task analysis of biologically inspired design by interdisciplinary design teams generates a task model of biologically inspired design: the task model describes the processes and the knowledge used in biologically inspired design.

**Comparative Analysis with Theories of Design, Analogy and Creativity:** Given our task model of biologically inspired design, we compared it with theories of biologically inspired design such as BioTRIZ (Vincent et al. 2006) and Design Spiral (Baumeister et al. 2012). However, because of space limitations, here we will compare our task model only with BioTRIZ. We also compared our task model with established theories of analogical reasoning such as Gentner (1983), Hofstadter (1996), Holyoak & Thagard (1996), and Kolodner (1993). Again because of space limitations, here we will compare our task model only with Gentner's structure-mapping theory of analogy.

## Data

The ME/ISyE/MSE/BME/BIOL 4740 classes from 2006 through 2013 resulted in 83 extended, open-ended design projects. The 83 case studies of the design projects in the classes were the focal points of our data collection. The projects involved identification of a design problem of interest to the team and conceptualization of a biologically inspired solution to the identified problem. Each design project grouped together an interdisciplinary team of typically 4-5 students. Each team had at least one student with a biology background and a few from different engineering disciplines. Each design team also had at least one faculty member. Each team identified a problem that could be addressed by a biologically inspired solution, explored a number of solution alternatives, and developed a final solution design based on one or more biologically inspired designs. Each design team presented its final design to an interdisciplinary design jury. Goel et al. (2015) describe a digital library, called the Design Study Library (DSL), of all 83 case studies.

## Empirical Findings

**Cross-Domain Analogies:** By definition, biologically inspired design engages cross-domain analogies from biology to engineering Although we have observed that extended episodes of biologically inspired design involve both within domain and cross-domain analogies (Vattam, Helms & Goel 2010), it is the essentialness of cross-domain analogies that defines the paradigm of biologically inspired design.

**Problem-Driven and Solution-Based Analogies:** We observed the existence of two high-level analogical

processes for biologically inspired design based on two different starting points – *problem-driven analogy* and *solution-based analogy* (Helms, Vattam & Goel 2009). In the problem-driven analogical process, designers identify a problem that forms the starting point for subsequent problem solving. They usually formulate their problem in functional terms (e.g., stopping a bullet). In order to find biological sources for inspiration, designers "biologize" the given problem, i.e., they abstract and reframe the function in more broadly applicable biological terms (e.g., what characteristics do organisms have that enable them to prevent, withstand and heal damage due to impact?). Designers use a number of strategies for finding biological sources relevant to the design problem at hand based on the "biologized" question, and then they research the biological sources in greater detail. Important principles and mechanisms that are applicable to the target problem are then extracted to a solution-neutral abstraction and applied to arrive at a trial design solution.

On the other hand, in the solution-based analogical process, designers begin with a biological source of interest. The designers understand (or research) their biological source to a sufficient depth to support the extraction of deep principles from it. Then they find human problems to which the principle can be applied. Finally, they apply the principle to develop a design solution to the identified problem.

The two analogical processes have different characteristics. Compared to problem-driven analogical processes, solution-based analogical processes tend to exhibit not only design fixation but also a fixation on the structure of the biological design (Helms, Vattam & Goel 2009). Again compared to problem-driven processes, solution-based design processes also tend to more often result in the generation of multifunctional designs, i.e. where a single design principle meets multiple functional goals (Helms, Vattam & Goel 2009). In general, a single case study may contain both problem-driven and solution-based analogical processes.

**Problem Decomposition and Level of Abstraction of Biological Analogy:** Biologically inspired design engages decomposition of the target design problem as well as functional decomposition of the biological system that acts as a source analogy to the design problem (Vattam, Helms & Goel 2007). Problem decomposition and functional decomposition of course are familiar ideas in design (e.g., Brown & Chandrasekaran 1989; Chandrasekaran 1990; Dym & Brown 2012; French 1996; Simon 1996). However, these decompositions appear to play a special role in biologically inspired design. The decomposition of the target design problem and the functional decomposition of the source biological system help identify the appropriate level for the analogical transfer from the biological system to the design problem.

**Problem Decomposition and Compound Analogies:** Problem decomposition appears to play a second special role in biologically inspired design. We found that biologically inspired design often entails compound analogies in which a new design concept is generated by composing the results of multiple cross-domain analogies (Vattam, Helms & Goel 2008). This process of compound analogical design relies on an opportunistic interaction between the processes of memory and problem solving. In this interaction, the target design problem is decomposed functionally, solutions to different subfunctions in the functional decomposition are found through analogies to different biological systems retrieved from memory, and the overall solution is obtained by composing the solutions for achieving the different subfunctions. Thus, the subfunctions in the functional decomposition of the design problem act as probes into a memory of biological systems.

**Interactive Analogical Retrieval:** Most designers are novices at biology (just as many biologists are naïve about design). Thus, designers typically do not have a large number of biological analogues stored in their long-term memory. Instead, we found that designers searched online for biological cases analogous to the target problems. Based on our observations, this was one of the predominant approaches for finding biological cases that typically were in the form of biology articles. Designers reported using a range of online information environments to seek information resources about biological systems. These included: (1) online information environments that provided access to scholarly biology articles like Web of Science, Google Scholar, ScienceDirect, etc., (2) online encyclopedic websites like Wikipedia, (3) popular life sciences blog sites like Biology Blog, (4) biomimicry portals like AskNature, and (5) general web search engines like Google. We call this phenomenon interactive analogical retrieval (Vattam & Goel 2013).

**Serendipity in Biologically Inspired Design:** The coupling of design problems and biological analogues often is serendipitous. For example, a design team may formulate a design problem, then find itself unable to make progress on it, and thus suspend additional work on the problem. At a later time, while working on a different problem, the team may serendipitously come across a biological analogue that provides a solution to the earlier problem, and therefore switch to the earlier problem.

**Abstraction and Transfer of Design Patterns:** We found biologically inspired design engages abstraction and transfer of several kinds of design patterns. Design patterns are abstractions of design cases, including generic domain principles (Bhatta & Goel 1994) and generic teleological mechanisms – causal mechanisms that achieve specific types of functions (Bhatta & Goel 1996). In particular, we have so far studied three kinds of design patterns in biologically inspired design: domain principles, causal mechanisms for accomplishing specific functions types,

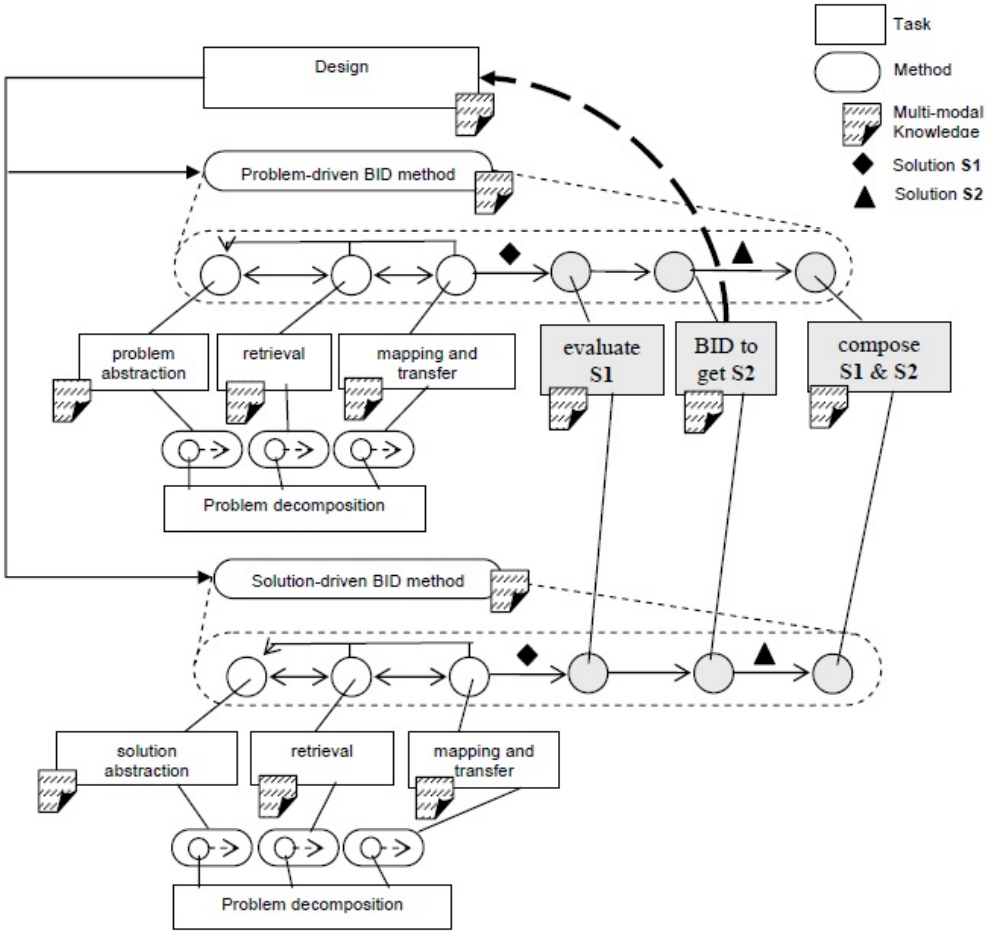



**Figure 2. A generic task model of biologically inspired design.**

and arrangements of structural components for accomplishing function types. We expect that there are many other types of design patterns yet to be discovered in biologically inspired design.

**Bridging Spatial and Temporal Scales:** Note that although the example in Figure 1 of this article is about product design at a spatial and temporal scale visible to the naked human eye, the scope of biologically inspired design is much larger. Thus, biologically inspired products may cover many spatial scales ranging from nanometers (e.g., biomolecules) to hundreds of kilometers (e.g., ecosystems), as well as many temporal scales ranging from nanoseconds to centuries. Often, a design pattern abstracted from a biological analogue may bridge across several spatial and temporal scales. For example, Weiler & Goel (2015) describe the crinkles on the surface of mitochondria cells as a source of analogy for designing human-scale devices for harvesting water from fog.

**Problem-Solution Co-Evolution***:* Conceptual design in biologically inspired design entails problem-solution co-evolution (Helms & Goel 2012). That is, the design process iterates between defining and refining the problem and the solution, with both the problem and the solution influencing each other (Maher & Tang 2003; Dorst & Cross 2001). As a solution (S) is developed and evaluated for a given problem (P), it reveals additional issues, spawning a new conceptualization of the problem (P+1). The process continues with the development of a new solution (S+1) and will iterate until a final solution is decided upon.

## Task Model

Figure 2 illustrates our generic task model of biologically inspired design based on the above findings. The overall *task* is design. This is accomplished by using two *methods:* problem-driven analogy and solution-driven analogy. Each method sets up *subtasks* like abstraction, retrieval, and mapping and transfer. Each subtask (e.g., retrieval) might, in turn, be accomplished by one of several methods (e.g., feature-based similarity matching for retrieval). *Knowledge* here refers to the knowledge used by a task or a method, for example, knowledge of design patterns. Note that

knowledge may be multimodal, for example, descriptive and depictive.

The problem-driven analogical process incorporates the design subtasks of problem formulation, problem reframing, biological solution search, defining biological solution, principle extraction and principle application. Similarly, the solution-based analogical process incorporates the design subtasks of defining biological solution, principle extraction, solution reframing, problem search, problem definition, and principle application. To avoid cluttering, Figure 2 illustrates only some of these subtasks of problem-driven and solution-based design.

Our task model of biologically inspired design also accounts for problem decomposition and compound analogies. In Figure 2, S1 represents the initial solution obtained. We add a new subtask "evaluate" to both problem-driven and solution-based methods. This subtask evaluates the initial solution S1 generated by a method. If the evaluation of S1 indicates that S1 addresses only a part of the design problem, then a new design sub-problem is spawned to address the remaining part(s) of the problem. Addressing the new sub-problem may lead to another partial solution S2. The subtask "compose" composes S1 and S2 to obtain a more complete solution to the original problem. For expediency, it is assumed here that subtask execution for compound analogy is sequential, represented by one-way arrows between the circles denoting the evaluation, designing and composition. The actual process may in fact involve much more complex interactions.

## Comparative Analysis

In this section, we compare the task model for biologically inspired design with both computational theories of analogical reasoning in creativity and creativity in biologically inspired design. Due to space limitations, here we will compare the task model only with Gentner's structure-mapping theory of analogy and Vincent et al.'s BioTRIZ theory of biologically inspired design.

**Structure Mapping:** Gentner's structure-mapping theory is one of the classical theories of analogy. Falkenhainer, Forbus & Gentner (1989) describe the structure-mapping engine, a computational implementation of the structure-mapping theory. Gentner & Markman (1997) discuss structure mapping as a more general theory of similarity and analogy. The process of analogical reasoning using structure mapping process starts with a target problem, and the method spawns the subtasks of retrieving a source analogue, finding mappings between the target problem and the source analogue, transfer of knowledge from the source to the target to generate a candidate solution, evaluation of the candidate solution, and storage of the new case in memory for potential reuse. The mapping task aligns the representations of the target problem and the source case – structure here refers to the structures of the two representations, and the principle of systematicity gives preference to higher-order relations.

A comparison of our task model of biological inspired design and the theory of analogical reasoning shows several similarities and differences:

- The structure-mapping theory of analogical reasoning is problem-driven. In contrast, biologically inspired design engages two distinct processes: problem-driven analogy and solution-based analogy.
- There are broad correspondences between some subtasks in the process of analogical reasoning and subtasks in the problem-driven analogical processes of biologically inspired design. For example, the "biological solution search" task in the problem-driven analogical process corresponds to the "retrieval" subtask in the structure-mapping theory. The aggregate of "defining biological solution," "principle extraction" and "principle application" subtasks in the problem-driven process corresponds to the "mapping" and "transfer" subtasks in the structure-mapping theory.
- On the other hand, there are subtasks in the problem-driven and solution-based analogical processes of biologically inspired design that are not directly matched by subtasks in the theory of analogical reasoning. In particular, the "problem abstraction" and "solution abstraction" subtasks in our task model of biologically inspired design that are preparatory to the subtasks of retrieval, mapping and transfer that follow.
- The structure-mapping theory of analogical reasoning does not itself address problem decomposition, but it can be extended to include problem decomposition, and, with it, the use of compound analogies that may potentially be at multiple levels of abstraction.
- While the structure-mapping focuses on the structure of the representations of the target problems and the solution analogues, our task model of biologically inspired design emphasizes the role of contents of knowledge, for example, the abstraction, acquisition, and use of knowledge of the design patterns.
- Most designers typically are novices in biology, and thus most biologically inspired designers rely on interactive analogical retrieval from online information sources. This is in contrast to the structure-mapping that assumes that the source analogues are available in the long-term memory of the agent.

**BioTRIZ:** Vincent et al.'s (2006) BioTRIZ is an information-processing theory of biologically inspired design derived from the earlier theory of engineering invention called TRIZ (Altshuller 1984). The TRIZ theory begins with a repository of design cases with known solutions, where each case is indexed by contradictions that arose in the original design situation. For example, consider a case in the repository that represents the design of an airplane wing. In this case the designer faces the contradiction of obtaining a material that is both strong and light-weight, and solves it using a solution, say $S_1$. This

case is then indexed by the contradiction "strong yet light-weight material." Additionally, if the particular solution $S_1$ belongs to a more general way of resolving contradictions of a particular kind, it may be categorized as a generic abstraction, such as "use porous materials" (to resolve the contradiction of strong yet light-weight material). TRIZ posits the existence of 40 generic ways of resolving conflicts, called *inventive principles*. The inventive principles were extracted by dropping the specifics of a particular case and domain and retaining the essence of how a particular class of contradictions is solved, so we can imagine each principle pointing to numerous cases (potentially belonging to different domains) in which that principle was used to resolve a conflict. The contradictions and the principles are organized in a contradiction matrix.

When the designer is presented with a design problem, she reformulates the problem to identify certain key contradictions in the requirements of the design. For each contradiction, she is reminded of a general inventive principle that is applicable for resolving that conflict. In addition to suggesting the essence of a solution for resolving that conflict, the inventive principle also points to a number of cases in which that general principle was instantiated. These cases can originate from domains different from the one in which the designer is currently working. TRIZ, however, does not address the issue of how transfer occurs.

Vincent et al. (2006) recently developed a modified version of TRIZ, called BioTRIZ, specifically for biologically inspired design. The primary difference between the two theories is a change in the features that compose the contradiction matrix. Whereas TRIZ defines 39 features with which to determine contradictions and index into inventive principles, the current version of BioTRIZ has six "operational fields": substance, structure, space, time, energy, and information.

A comparison of our task model and BioTRIZ reveals the following similarities and differences:

- Both BioTRIZ and our model address cross-domain analogies between biological and technological systems.
- BioTRIZ is a prescriptive theory of biologically inspired design, derived from best practices in mechanical engineering design. In contrast, our task model is a descriptive theory based on *in situ* observations of biologically inspired design.
- The processing in BioTRIZ is problem-driven. The processing in BioTRIZ always begins with a specification of a design problem. It does not directly address solution-based analogical process. Our task model accounts for both problem-driven and solution-based analogies.
- BioTRIZ does not directly address compound analogy. However, since a design problem may contain multiple contradictions, and the various contradictions may require the invocation of different principles, compound analogy appears to be feasible in BioTRIZ.

## So Is Biologically Inspired Design Different?

The above comparative analysis brings us to the question often asked by design theorists: is biologically inspired design different from other design paradigms? After all, analogical reasoning is used extensively in other design paradigms, and cross-domain analogies often are the basis of creativity in the other design paradigms. So is analogical reasoning in biologically inspired design different from analogical reasoning in other design paradigms, other than the obvious fact the source analogues are from biology? Or, put a little differently, what precisely makes biologically inspired design a new design paradigm from the perspective of analogy and creativity?

Note that the question here is not whether biological and technological systems are different. As Vincent et al. (2006) note, "biology and technology solve problems in design in rather different ways:" biological systems often use information for functions for which technological systems tend to use energy. French (1994) and Vogel (2000) make detailed analyses of the similarities and differences between biological and technological systems: biological systems in general tend to be more multifunctional than technological systems. Instead, the question here is: are the *processes* of analogical reasoning in biologically inspired design fundamentally different from that of other design paradigms?

Our task model offers some insights into what may make analogical reasoning in biologically inspired design different from analogical reasoning in other domains, thereby making biologically inspired design a new design paradigm:

1. Biologically inspired design by definition is based on cross-domain analogies. While many design processes in and out of biologically inspired design sometimes engage cross-domain analogies, and while biologically inspired design also frequently engages within domain analogies (Vattam, Helms & Goel 2010), insofar as we know there are not many other kinds of design that by definition are based on cross-domain analogies.
2. Biologically inspired design often entails compound analogies. In particular, the target design problem is decomposed functionally, solutions to different subfunctions in the functional decomposition are found through analogy to different biological systems retrieved from a functionally indexed memory, and the overall design solution is obtained by composing the solutions for achieving the different subfunctions. While problem decomposition could be introduced into the structure-mapping theory of analogical reasoning, compound analogy appears to be a stronger characteristic of biologically inspired design.
3. Biologically inspired design engages two different analogical design processes, namely, problem-driven analogy and solution-based analogy. We first observed these two analogical processes in our *in situ* studies of biologically inspired design in practice. Insofar as we know, all information-processing theories of analogy

(e.g., Dunbar 2001; Gentner 1983; Gick & Holyoak 1983; Goel 1997; Hofstadter 1996; Holyoak & Indurkhya 1992; Thagard 1996; Keane 1988; Kolodner 1993) focus on and emphasize problem-driven analogy. Further, insofar as we know, computational theories of all other kinds of design focus on and emphasize problem-driven design (e.g., Brown & Chandrasekaran 1989; Chandrasekaran 1990; Dym & Brown 2012; French 1996; Maher & Tang 2003; Simon 1996). Therefore, that biologically inspired design entails both problem-driven and solution-based analogies appears to be another definitional characteristic of biologically inspired design.

4. Most designers typically are novices in biology, and thus most designers rely on interactive analogical retrieval from online information sources while engaging in biologically inspired design. This is in contrast to all theories of analogical reasoning that assume that source analogues are available in the long-term memory of the agent.
5. In biologically inspired design, problems and solutions co-evolve. This is similar to creative processes in other design domains but in sharp contrast to current theories of analogical reasoning.

From the perspective of creativity in design, we should add that the question here is not binary. Most of the processes that occur in biologically inspired design also occur in other creative design. Instead, the difference lies in focus and emphasis. As an example, other types of creative design often engage cross-domain analogies irrespective of the design domains, but biologically inspired design is defined by cross-domain analogies.

## Conclusions

In this paper, we found that biologically inspired design indeed is a novel methodology for creative design for at least five reasons: (1) Biologically inspired design by definition engages cross-domain analogies. (2) Problems and solutions in biologically inspired design co-evolve. (3) Problem decomposition plays a fundamental role in biologically inspired design. (4) Biologically inspired design often involves compound analogy, entailing a complex interplay between the processes of problem decomposition and the processes of analogical retrieval from memory. (5) Biologically inspired design entails two distinct but related processes: problem-driven analogy and solution-based analogy. For this reason, we now prefer the term biologically inspired *invention*, as in the title of this paper: while design always starts with a problem, invention need not, sometimes starting with a solution and only later finding a problem, perhaps by serendipity.

These distinctions make for important differences in developing computational theories, techniques and tools for supporting biologically inspired design. For example, as we mentioned in the introduction, Nagle (2014) describes an engineering-to-biology thesaurus, with the (implicit) assumption that biologically inspired design is not very different from other analogy-based creative processes, that if we could only bridge the vocabulary gap between design and biology, we could borrow the rest from extant theories of design, analogy and creativity. However, if biologically inspired design is different, then we also need a different set of computational tools based on a different set of hypotheses. For example, Vattam & Goel (2013) describe Biologue, a computational tool for interactive analogical retrieval from online information sources that is based on the observation that analogical retrieval in biologically inspired design is situated online.

Further, our work on biologically inspired design indicates that research on computational creativity may need to develop new theories of analogical reasoning that incorporate a more dynamic, a more flexible view of cognition, including problem-driven and solution-based analogies, problem decomposition and compound analogies, interactive analogical retrieval, and problem-solution coevolution. This makes for an exciting research agenda in computational creativity.

**Acknowledgements:** This article is based in part on collaborative research with Michael Helms, Swaroop Vattam, Bryan Wiltgen, and Jeannette Yen (the primary instructor of the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course) over several years. This article has also benefited from discussions with Norbert Hoeller, Spencer Rugaber, and Filippo Salustri.

## References

Altshuller, G. (1984). *Creativity as an exact science*. Gordon and Branch Publishers, Luxembourg.

Baumeister, D., Tocke, R., Dwyer, J., Ritter, S., & Benyus, J. (2012) *Biomimicry Resource Handbook.* Biomimicry 3.8, Missoula, MT, USA.

Benyus, J. (1997) *Biomimicry: Innovation Inspired by Nature*. New York: William Morrow.

Bhatta, S., & Goel, A. (1994) Model-Based Discovery of Physical Principles from Design Experiences. *AIEDAM 8(2):113-123*.

Bhatta, S., & Goel, A. (1996) From Design Cases to Generic Mechanisms. *AIEDAM 10:131-136.*

Bonser, R., & Vincent, J.. (2007). Technology trajectories, innovation, and the growth of biomimetics. *In Procs. Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,* 221(10), 1177–1180.

Brown, D., & Chandrasekaran, B. (1989) *Design Problem Solving: Knowledge Structures and Control Strategies.* San Mateo, California: Morgan Kaufmann.

Chandrasekaran, B. (1990) Design problem solving: a task analysis. *AI Magazine 11(4):59-71.*

Crandall, B., Klein, G., & Hoffman, R. (2006). *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. MIT Press.

Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem-solution. *Design Studies*, 22, pp. 425-437.

Dunbar K. (2001) The Analogical Paradox. In Gentner, D, Holyoak, K.J., & Kokinov, B.N. (Eds.) *The Analogical*

*Mind: Perspectives from Cognitive Science.* MIT Press. Boston.

Dym, C., & Brown, D. (2012) *Engineering design: Representations and Reasoning.* 2nd Edition, Cambridge University Press.

Falkenhainer, B., Forbus, K., Gentner, D. (1989) The Structure-Mapping Engine: Algorithm and Examples. *Artificial Intelligence, 41(1): 1-63.*

Fish, F., Weber, P., Murray, M., & Howles, L. (2011) The Tubercles of Humpback Whale Fillers: Application of Bioinspired Technology. *Integrative and Comparative Biology* 51(1): 203-213.

French, M. (1994) *Invention and evolution: design in nature and engineering.* 2nd edition. Cambridge University Press.

French M. (1996). Conceptual design for engineers. 3rd ed. Springer-Verlag. London.

Gentner, D. (1983) Structure Mapping: A Theoretical Framework for Analogy. *Cognitive Science 7(2): 155-170.*

Gentner, D., & Markman, A. (1997) Structure Mapping in Analogy and Similarity. *American Psychologist 52(1): 45-56.*

Gick M, and Holyoak KJ (1983). Schema induction and analogical transfer. *Cognitive Psychology 15:1-38.*

Gleich, A. von, Pade, C., Petschow, U., & Pissarskoi, E. (2010). *Potentials and Trends in Biomimetics.* Berlin: Springer.

Goel, A. (2013) Biologically Inspired Design: A New Program for AI Research on Computational Sustainability. *IEEE Intelligent Systems* 28(3): 80-84.

Goel, A., & Bhatta, S. (2004) Use of Design Patterns in Analogy-Based Design. *Advanced Engineering Informatics* 18(2):85-94, 2004.

Goel, A., & Chandrasekaran, B. (1992) Case-Based Design: A Task Analysis. In *Artificial Intelligence Approaches to Engineering Design, Volume II: Innovative Design*, C. Tong and D. Sriram (editors), pp. 165-184, San Diego: Academic Press, 1992.

Goel, A., McAdams, D., & Stone, R. (editors, 2014) *Biologically Inspired Design: Computational Methods and Tools*, London, UK: Springer-Verlag.

Goel, A., Zhang, G., Wiltgen, B., Zhang, Y., Vattam, S., & Yen, J. (2015) On the Benefits of Digital Libraries of Case Studies of Analogical Design: Documentation, Access, Analysis and Learning. *AIEDAM* 29(2):215-227.

Helms, M., Vattam, S., & Goel, A. (2009) Biologically Inspired Design: Process and Products, *Design Studies*, 30(5):606-622.

Helms, M., & Goel, A. (2012), Analogical Problem Evolution in Biologically Inspired Design. In *Procs. 5th International Conference on Design Computing and Cognition*, College Station, Texas. Berlin:Springer.

Hofstadter, D., (editor, 1996) *Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought.* Harvester Wheatsheaf.

Holyoak, K., & Thagard, P. (1996) *Mental Leaps: Analogy in Creative Thought.* MIT Press.

Indurkhya, B. (1992). *Metaphor and cognition.* Dordrecht: Kluwer.

Keane M (1988) *Analogical Problem Solving*. Ellis Horwood. Chichester

Kolodner, J. (1993) *Case-Based Reasoning.* Morgan Kauffman.

Lepora, N., Verschure, P., & Prescott, T. (2013) The state of the art in biomimetics. *Bioinspiration & Biomimetics 8(1).*

Maher, M.L., & Tang, H. (2003) Co-evolution as a computational and cognitive model of design. *Research in Engineering Design, 14(1):47-64.*

Nagel, J. (2014) A Thesaurus for Bioinspired Engineering Design. In *Biologically Inspired Design: Computational Methods and Tools,* Goel, McAdams & Stone (editors), Springer.

Simon, H. (1996) *Sciences of the Artificial.* 3rd Edition. MIT Press.

The Biomimicry Institute (2011). http://biomimicry.org/ Last retrieved on April 28, 2011.

Turner, J. (2007). *The Tinkerer's Accomplice: How Design Emerges from Life Itself.* Harvard University Press.

Vattam, S., & Goel, A. (2013) Biological Solutions for Engineering Problems: Cross-Domain Textual Case-Based Reasoning in Biologically Inspired Design. In *Procs. 21st International Conference on Case-Based Reasoning*, July 2013, pp. 343-357.

Vattam, S., Helms, M., Goel, A. (2007) Biologically Inspired Innovation in Engineering Design: A Cognitive Study. Technical Report GIT-GVU-07-07, Graphics, Visualization & Usability Center, Georgia Institute of Technology.

Vattam, S., Helms, M., & Goel, A. (2008) Compound Analogical Design: Interaction Between Problem Decomposition and Analogical Transfer in Biologically Inspired Design. In *Proc. Third International Conference on Design Computing and Cognition,* Atlanta, June 2008.

Vattam, S., Helms, M., & Goel, A. (2010) A Content Account of Creative Analogies in Biologically Inspired Design. *AIEDAM* 24: 467-481.

Veale, T. (2003) The Analogical Thesaurus. In *Procs. 15th Innovative Applications of AI Conference (IAAI-2003),* pp. 137-142.

Vincent J., Bogatyreva O., Bogatyrev N., Bowyer A, Pahl A. (2006) Biomimetics: its practice and theory. *Journal of the Royal Society Interface*, 3, 471–482.

Vincent, J., & Mann, D. (2002) Systematic Transfer from Biology to Engineering. *Philosophical Transactions of the Royal Society of London*, 360: 159-173.

Vogel, S (2000) *Cat's Paws and Catapults: Mechanical Worlds of Nature and People*. W.W. Norton and Company.

Weiler, C., & Goel, A. (2015) From Mitochondria to Water Harvesting: A Case Study in Biologically Inspired Design. *IEEE Potentials,* 34(2): 38-43.

Yen, J., Weissburg, M., Helms, M., & Goel, A. (2011) Biologically inspired design: a tool for interdisciplinary education. In *Biomimetics: Nature-Based Innovation, Y. Bar-Cohen (editor),* Taylor & Francis.

Yen, J., Helms, M., Tovey, C., Weissburg, M., & Goel, A. (2014) Adaptive Evolution of a Biologically Inspired Design Course. In *Biologically Inspired Design: Computational Methods and Tools,* Goel, McAdams & Stone (editors), pp. 153-200, London: Springer.

# The role of blending in mathematical invention

**F. Bou**
**M. Schorlemmer**
IIIA - CSIC
Barcelona

**J. Corneli**
Computing
Goldsmiths, London

**D. Gómez-Ramírez**
Cognitive Science
Osnabrück

**E. Maclean**
**A. Smaill**
Informatics
Edinburgh

**A. Pease**
Computing
Dundee

**Abstract**

We model the mathematical process whereby new mathematical theories are invented. Here we explain the use of conceptual blending for this purpose, and show examples to illustrate the process in action. Our longer-term goal is to support machine and human mathematical creativity.

## Introduction

We are concerned with creativity in mathematics: creativity as evinced by human and artificial mathematicians, individually and collectively.

Work on *conceptual blending* has been much influenced by Fauconnier and Turner (1998, 2002). More recently, the centrality of conceptual blending to creativity has been stressed by Turner (2014), where he writes:

> . . . the human spark comes from our advanced ability to *blend* ideas to make *new* ideas. Blending is the origin of ideas. (Turner, 2014, p 2)

The claim is that blending in this sense is a general human cognitive ability, and as such applies to mathematics, as much as to art, poetry, music and so on (see for example Turner (2005)).

The place of mathematics and the sciences among creative endeavours has been stressed by the literary critic George Steiner:

> It is in mathematics and the sciences that the concepts of creation and of invention, of intuition and of discovery, exhibit the most immediate, visible force.
>
> Steiner (2001, p 145)

Blending involves recognising features common to mathematical concepts, even when expressed in different terminology. The role of mathematical analogy in creative mathematics is well expressed by Weil (1960), and a general plea for analogical reasoning within science in (Arbib and Hesse, 1986).

We are investigating *computational* accounts of mathematical creativity, taking conceptual blending as a key ingredient. The work of Goguen (1999, 2005) has provided a general framework for comparison of conceptual spaces, and computation of blends. This enables the use of richer representation formalisms, and so is closer to contemporary mathematics than previous computational realisations of blending, such as in Pereira (2007).

This paper deals with the creative process in mathematics, as modelled along the lines above. We focus on the use of blending within a single process, searching for blends satisfying some evaluation criteria, from the starting point of some given conceptual spaces.

While cognitive issues are important to us, this paper is focused on issues in representation and representation change; there are however brief comments on cognition in the conclusions.

We start by providing some background, followed by an example to illustrate the components involved in our approach. A historical example based on Georg Cantor's work follows. The most extended example was carried out by a pure mathematician (D. Gómez-Ramírez), working in a domain close to his own; in this case, the blend mechanism threw up some unexpected properties, which provoked new work by the mathematician.

Subsequently we give some more speculative thoughts on where this work can go in the future, by considering Galois theory as a test-bed. Finally, we discuss the evaluation of work along these lines, and give some conclusions.

## Background

### Blending in Mathematics

Lakoff and Núñez (2000) are among the first to present a cognitive account of the origin and development of mathematical ideas,[1] arguing against the "romance of mathematics" in which mathematics is presented as an ever-increasing set of universal, absolute, certain truths which exist independently of humans. They present the thesis that human mathematics is grounded in bodily experience of a physical world, and mathematical entities inherit properties which objects in the world have, such as being consistent or stable over time. Exploring the physical world of object collection might lead to concepts like the empty collection and rules like "adding a collection of $n$ objects to an empty

[1] This is lamented by Lakoff and Núñez, who claim that (prior to their work), "there was still no discipline of mathematical idea analysis from a cognitive perspective" (Lakoff and Núñez, 2000).

collection yields a collection with $n$ objects". People then form grounding metaphors between the physical world and an abstract mathematical world, allowing us to project from everyday experiences onto abstract concepts, thus leading to the concept of zero and the axiom that $n+0 = n$. Lakoff and Núñez posit that blending different mathematical metaphors leads to more complex ideas (see also Alexander (2011)).

Alongside this account of mathematical cognition, mainstream contemporary mathematics has developed its own methodology and foundations, enjoying an exceptional place among scientific disciplines. Its methods, objects of study and sometimes astonishing results have widespread, if not universal, acceptance.

In conclusion, mathematics is a scientific discipline having not only a fundamental cognitive component, necessary in its development, but also possessing a collection of general principles and structures going beyond a particular school of thought. Among these general processes we want to highlight in this paper the importance that conceptual blending has in mathematics, incorporating both cognitive and mathematically specific aspects in order to create new mathematical concepts.

## Terminology for conceptual blending

Our notion of conceptual blending is informed by Category theory, and highly influenced by Goguen's work on concepts (Goguen, 2005). In this paper we use the terminology below, and elucidate the terminology by means of a running example – discovering a version of the integers (in the sense of providing a partial approach to the genuine integers) using blending.

**Conceptual spaces** are partial and temporary representational structures which are constructed on the fly when talking about a particular situation, which are informed by the knowledge structures associated with a domain. These are influenced by Boden's idea of a concept space which is mapped, explored and transformed by transcending mapped boundaries (Boden, 1977), and form the input spaces to our blend.

As an example of two conceptual spaces, consider one as a theory NAT – a theory of the natural numbers, and FUNC – a theory of a total unary function with an inverse. We will refer back to these theories in this exposition.

(Many-sorted) First-order **Axioms** are the criteria which will be used here to delineate the conceptual spaces. The axiomatic method has been a fundamental aspect of mathematical research since Euclid, and various axiom changes have led to revolutions in mathematics. For instance, rejecting the parallel postulate opened up fascinating new areas of non-Euclidean geometry.

The precise formulations for NAT and FUNC can be found in Listings 1 and 2. Notice that these formulations obviously refer to partial representations of the genuine concepts employed by mathematicians. In the conceptual space with theory NAT, an example of an axiom is $\forall x.\neg\ 0 = s(x)$ – that is that zero is not a successor element. The conceptual space with theory FUNC has an axiom $\forall x.\ f(finv(x)) = x$.

**Signature morphisms** between conceptual spaces are mappings from the symbols of the source conceptual space into the symbols of the other conceptual space. For example NAT contains a function $\lambda x : Nat.\ s(x)$ that maps $x$ to its successor, and FUNC contains a function defined over a set $X$ that maps each element to an image $\lambda x : X.\ f(x)$. A theory $G$ with a morphism to both NAT and FUNC might contain a function $\lambda x : N.\ func(x)$ that takes every number in some set $N$ to its image under *func*. When we show a mapping we write this as

$$s \quad \leftarrow_{\phi(G,\text{NAT})} \quad func \quad \rightarrow_{\phi(G,\text{FUNC})} \quad f \qquad (1)$$

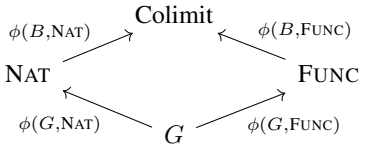
$$Nat \quad \leftarrow_{\phi(G,\text{NAT})} \quad N \quad \rightarrow_{\phi(G,\text{FUNC})} \quad X \qquad (2)$$

The mapping $\phi(G, \text{NAT})$ is a signature morphism from $G$ to NAT. Note that associated types are also mapped.

**Input Spaces** refer to two or more conceptual spaces of interest.

**Generic spaces** are conceptual spaces that possess commonality between input spaces.

**Colimits** are conceptual spaces representing a blend of input spaces with respect to a given generic space and a set of signature morphisms. These are uniquely computed given a generic space and a set of morphisms. Here is a diagrammatic representation of such a computation in our example using theories NAT and FUNC :




The conceptual space represented by the Colimit is often referred to as the *blend*.

**Internal Evaluation** constitutes a variety of techniques to determine whether a computed colimit is viable as a conceptual space. In our example, since the conceptual spaces are mathematical theories, we can exploit the notion of consistency. This is a way of evaluating whether a blend is not only creative, but also valid. In the example of theories NAT and FUNC, the computed blend is inconsistent due to the emergent axioms in the computed colimit. The only type existing within the colimit is from now on referred to as $\mathbb{Z}$ to distinguish it from the natural numbers. Notice that in the colimit it holds that:

$$\forall x : \mathbb{Z}.\neg\ zero = s(x)$$

$$\forall x : \mathbb{Z}.\ s(sinv(x)) = x\ .$$

This is an inconsistency, as from the second axiom we see that there is an element for which 0 is the successor.

**Weakening** refers to the process of weakening the input theories by removing symbols or axioms. If we remove the axiom

$$\forall x : Nat.\neg\ zero = s(x)$$

then the resulting computed colimit contains a mathematical theory which is consistent.

Martinez et al. (2014) provides an algorithm to explore the space of blends resulting from given input spaces and a given generic space, where weakening is achieved by omitting axioms. The algorithm returns the blends which are consistent, and maximally so, among those in this space of

blends. This algorithm assumes that consistency of relevant theories can be checked, so is not always effective.

**Running the blend** refers to elaborating or completing a mathematical theory. Sometimes there are missing definitions which need to be discovered. For example in the new theory the following axiom appears

$$\forall x, y : \mathbb{Z}.\ s(x) + y \quad = \quad s(x + y)\ ,$$

but we also are interested in theorems such as

$$\forall x, y : \mathbb{Z}.\ \mathit{sinv}(x) + y \quad = \quad \mathit{sinv}(x + y)\ .$$

Finding suitable theorems is an example of running the blend, and from which it is possible to discover and prove theorems such as

$$\forall x, y : \mathbb{Z}.\ \mathit{sinv}(x) + s(y) \quad = \quad x + y\ .$$

### Technologies

The approach explained above corresponds to Goguen's proposal (Goguen, 1999) for implementing blending, but slightly simplified (as in Kutz, Neuhaus, Mossakowski, and Codescu (2014)): we use the normal colimit construction, rather than $\frac{3}{2}$-colimits (both described in Goguen (1999)).

Additionally we assume that the conceptual spaces involved are given using a CASL specification (Astesiano et al., 2002) and that the morphisms are theorem preserving (i.e. map theorems to theorems). The reason for these assumptions is that in these cases it is well-known how to compute colimits: the colimit specification essentially corresponds to the disjoint union of the two target conceptual spaces except for not repeating the symbols given in the common source conceptual space. Moreover, we will be using the HETS system (Mossakowski, Maeder, and Lüttich, 2007) to compute such colimits. The code for the implemented examples in this paper is available on-line.[2]

The use of CASL specifications means that we deal with first-order logic; CASL is supported in the HETS system, and colimits here can be computed in the current implementation of HETS. Although higher-order logic (with Henkin semantics) is available in HETS (indeed in CASL) and the colimits are well-known to exist (because higher-order in this form is reducible to many-sorted first-order logic), it is worth noticing that the calculus of such colimits is not currently available in HETS. This restricts the formalisms that can be used directly for our purposes, where computation of colimits is central to our approach.

## Blending and the infinite

### Example Revisited – the Integers

As a first demonstration of the machinery involved in blending mathematical theories, we consider combining a theory of natural numbers with the concept of the inverse of a function to obtain the integers. Let us assume a simple partial axiomatisation of the natural numbers (without order axioms) as shown in Listing 1, and call this theory NAT. Now let us also define a simple theory which introduces the concept of a function with an inverse as shown in Listing 2, and call this theory FUNC.

[2] See: https://github.com/ewenmaclean/ICCC2015_hetsfiles

**spec** NAT =
**sort** *Nat*
**ops** *zero* : *Nat*;
*s* : *Nat* → *Nat*;
__+__ : *Nat* × *Nat* → *Nat*
∀ *x*, *y* : *Nat*
• $s(x) = s(y) \Rightarrow x = y$
• $\neg\ zero = s(x)$
• $s(x) + y = s(x + y)$
• $zero + y = y$
**end**

Listing 1: A theory of the natural numbers without order

**spec** FUNC =
**sort** *X*
**op** $f : X \to X$
**op** $\mathit{finv} : X \to X$
∀ *x* : *X*
• $f(\mathit{finv}(x)) = x$
• $\mathit{finv}(f(x)) = x$
**end**

Listing 2: A theory with a function and its inverse defined

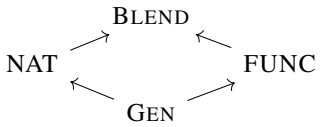
**Identifying a Generic Space** In order to incorporate the notion of blending here we want to be able to identify a "generic" component of each theory and compute the colimit. We can use the HDTP system (Gust, Kühnberger, and Schmidt, 2006; Schmidt, 2010) to discover a common theory and signature morphism between symbols in the two theories NAT and FUNC. The Generic theory GEN contains a sort *N* and a function *func*, and the morphisms from the Generic theory to NAT and FUNC are:

$$s \quad \leftarrow_{\phi(G,\mathrm{NAT})} \quad \mathit{func} \quad \rightarrow_{\phi(G,\mathrm{FUNC})} \quad f \qquad (3)$$

$$\mathit{Nat} \quad \leftarrow_{\phi(G,\mathrm{NAT})} \quad N \quad \rightarrow_{\phi(G,\mathrm{FUNC})} \quad X \qquad (4)$$

Here the successor function is identified in the mapping with the function in the theory FUNC.

**Computing the Colimit** The HETS system (Mossakowski et al., 2007) can then be exploited to find a new theory by computing the colimit:




This generates the theory shown in Listing 3 (for the sake of understanding it is used *p*, for predecessor, instead of *sinv*).

**Removal of Inconsistencies** This theory is automatically determined to be inconsistent due to the axioms

$$\forall x : \mathbb{Z}.\neg\ zero = s(x) \qquad (5)$$

$$\forall x : \mathbb{Z}.\ s(p(x)) = x \qquad (6)$$

**spec** SPEC =
**sort** $N$
**op** $\_+\_ : N \times N \to N$
**op** $p : N \to N$
**op** $s : N \to N$
**op** $zero : N$
$\forall x, y : N \bullet s(x) = s(y) \Rightarrow x = y$
$\forall x : N \bullet \neg\, zero = s(x)$
$\forall x, y : N \bullet s(x) + y = s(x + y)$
$\forall y : N \bullet zero + y = y$
$\forall x : N \bullet s(p(x)) = x$
$\forall x : N \bullet p(s(x)) = x$
**end**

Listing 3: An inconsistent partial approach to the integers (without order)

Removal of the limiting axiom (5) from Listing 1 results in generating a blend theory which is very similar to what we understand to be the integers as shown in Listing 4.

**spec** SPEC =
**sort** $N$
**op** $\_+\_ : N \times N \to N$
**op** $p : N \to N$
**op** $s : N \to N$
**op** $zero : N$
$\forall x, y : N \bullet s(x) = s(y) \Rightarrow x = y$
$\forall x, y : N \bullet s(x) + y = s(x + y)$
$\forall y : N \bullet zero + y = y$
$\forall x : N \bullet s(p(x)) = x$
$\forall x : N \bullet p(s(x)) = x$
**end**

Listing 4: A consistent partial approach to the integers (without order)

**Running the Blend** Running the blend refers to discovering definitions or adding axioms to flesh out the blend. In the example of the version in Listing 4, the definition of plus needs to be extended to understand how to calculate with the predecessor function:

$$p(x) + y = p(x + y)$$

from which theorems such as

$$p(x) + s(y) = x + y$$

can be proved.

## Potential and actual infinity

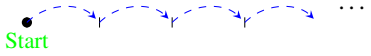
Some of the ideas of Lakoff and Núñez (2000) have been reworked by the authors, with increased emphasis on conceptual blending. In particular, the analysis of mathematical infinity, given in metaphorical form as the "Basic Metaphor of Infinity" (BMI) in Lakoff and Núñez (2000), is represented in blend form in Núñez (2005) as the "Basic Mapping of Infinity" (so, still "BMI").

We show here how this blend works out in our setting. The BMI suggests that the notion of completed infinity, in particular the possibility of transfinite numbers in the sense of Cantor, comes from a blend of the notion of completed, finite process with that of a potentially infinite and endless process.

Thus take two corresponding input spaces, given by CASL specifications **FinEnd** and **Inf** corresponding to the following diagrams

**FinEnd:**


**Inf:**




- **FinEnd:** Completed Iterative Processes are those that from some initial state, terminate in a final state after a finite number of state transitions. One such case is chosen.
- **Inf:** Infinite Iterative Processes are those that continue indefinitely to change state.

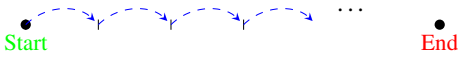
In both cases, the arrows indicate steps of the processes, and the process states are in a discrete linear order indicated by left-to-right order in the diagrams.

The generic space **Gen** simply identifies the start states, the notion of process step, and the linear ordering of states.

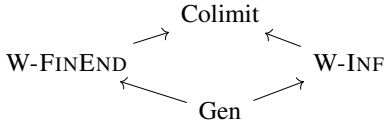
Now we can compute the blend of these spaces, which includes new features taken from both of the input spaces. This blend is *inconsistent*, for the following two reasons:

1. the number of states is finite (from **FinEnd**), and infinite (from **Inf**);
2. there both is an end state (from **FinEnd**) and is no end state (from **Inf**).

Search through the possibilities of weakening the input spaces by omitting as few axioms as possible among those involved in an inconsistency reveals the possibility of a structure with infinitely many states (from **Inf**) and an end state (from **FinEnd**). Computing the colimit from the weakened input spaces **W-FinEnd**, **W-Inf** gives a theory corresponding to this diagram:




Thus we have a blend as in the earlier examples:




## Prime Ideals as a blend

### Introduction

One of the most fundamental concepts of modern mathematics, which is the basis of commutative algebra and a seminal ingredient of the language of schemes in modern algebraic

geometry, is that of *prime ideal* (Grothendieck and Dieudonné, 1971; Eisenbud, 1995).

The terminology "prime ideal" relates to the older notion of "prime number". The initial aim of this work was to look for a blend between prime numbers (from the integers) and the ideals of a commutative ring, to see what would emerge. It turned out that *the blend process*, along with providing a definition for prime ideals, also *suggested an unexpected concept in the context of rings*, namely what will be called Containment Division Rings (CDR). In turn, this prompted questions and proofs about this concept – thus running the blend (space prevents description of this step in this paper).

We present a first blend involving weakening, followed by a second blend from fuller input spaces, where the emergent concept of CDR appears.

## The first conceptual space

Let $(R, +.*, 0, 1)$ be a commutative ring with unity (see the formal definition and examples in Eisenbud (1995)). Now, $R$ can be understood as the sort containing the elements of the corresponding commutative ring with unity. An ideal $I$ is a subset of $R$ satisfying the following axiom:

$$(\forall i, j \in I)(\forall r \in R)(i + (-j) \in I \wedge r * i \in I).$$

Let us define a unary relation (predicate) *isideal* on the set (sort) of subsets of $P(R)$ corresponding to this definition. Now, we define

$$\mathrm{Id}(R) = \{A \in P(R) : \mathit{isideal}(A)\}.$$

Ideals are "multiplied" using the following definition:

$$I \cdot_\iota J = \left\{ \sum_{k=1}^{n} i_k \cdot j_k : n \in \mathbb{N} \wedge i_1, \ldots, i_n \in I \wedge j_1, \ldots, j_n \in J \right\}.$$

In other words, $I \cdot_\iota J$ is the smallest ideal extending the set $\{i \cdot j : i \in I \wedge j \in J\}$.

The key property that we want to keep in the blend is the one saying that this operation $\cdot_\iota$ has a neutral element $1_\iota$, which can be seen as an additional notation for the ring. On the other hand, we want to see the containment relation $\subseteq$ as a binary relation over the sort $\mathrm{Id}(R)$.

Summarizing, our first conceptual space consists of sorts $R, \mathrm{Id}(R)$ and $P(R)$; operations $+, *, 0_R, 1_R, 1_\iota$ and $\cdot_\iota$; and the relations $\subseteq$ and *isideal*.

Let us denote this space by $\mathbb{I}$.

## The second conceptual space

Let $\mathbb{Z}$ be the set of the integer numbers. Here, we choose any partial axiomatization of them including at least the fact that $(\mathbb{Z}, *, 1)$ is a commutative monoid. We define also an upside-down divisibility relation $\lfloor$ defined as $e \lfloor g := g|e$, i.e. there exists an integer $c$ such that $e = c * g$. Let us define a unary relation *isprime* on $\mathbb{Z}$ as follows: for all $p \in \mathbb{Z}$, $\mathit{isprime}(p)$ holds if $p \neq 1$ and:

$$(\forall a, b \in \mathbb{Z})\ \big((ab \lfloor p) \rightarrow (a \lfloor p \vee b \lfloor p)\big).$$

Besides, we define the set (sort) of the prime numbers as

$$\mathit{Prime} = \{p \in \mathbb{Z} : \mathit{isprime}(p)\}$$

In the CASL language, we consider $\mathbb{Z}$ as the sort of the integer numbers, $*$ as a binary operation, *prime* as a predicate and $\lfloor$ as a binary relation, any of them defined over the sort $\mathbb{Z}$. We denote this conceptual space by $\mathbb{P}$.

## The Generic Space

The generic space $\mathbb{G}$ consists of a set (sort) $G$ with a binary operation $*_G$, a neutral element $S$ and a binary relation $\leq_G$.

## The Blending Morphisms

The morphism to $\mathbb{I}$ uses:
$\varphi(G) = \mathrm{Id}(R), \varphi(*_G) = *_\iota, \varphi(S) = 1_\iota$ and $\varphi(\leq_G) = \subseteq$;
the morphism to $\mathbb{G}$ uses:
$\delta(G) = \mathbb{Z}, \delta(*_G) = *, \delta(S) = 1$ and $\delta(\leq_G) = \lfloor$.

## The Axiomatization of the Blending

A straightforward colimit construction based on the input and generic spaces above yields a consistent space with properties inherited both from the prime elements into the integers and from the ideals of commutative rings; one of the concepts is a notion of prime ideals, another is that of CDR.[3] Here we describe briefly a weakening of the given spaces that makes the resultant blend more generally applicable.

From the properties defining the integers we transfer into the blend only the fact that $\mathbb{Z}$ is a set with a binary operation $*$ having 1 as neutral element and $\lfloor$ as a binary relation, without taking into account its formal definition.

Now after computing the colimit, we obtain that any element $P \in G$ (i.e., an ideal of $S$) satisfies the predicate *isprime* if and only if

$$P \neq S \wedge (\forall X, Y \in G = \mathrm{Id}(S))(X \cdot_\iota Y \subseteq P \rightarrow (X \subseteq P \vee Y \subseteq P)).$$

Thus, the predicate *isprime* turns out to be the predicate characterizing the primality of ideals of $S$ and the set (sort) $Prime$ turns out to be the set of prime ideals of $S$.

Using the weakened input spaces, the blending space consists of the axioms assuring that $S$ is a commutative ring with unity, $G$ is the set of ideals of $S$, *isprime* is the predicate specifying primality for ideals of $S$ and *Prime* is the collection of all prime ideals of $S$.

## Implementation for prime ideals over CDR-s as a blend

In this section we construct the concept of prime ideal over a CDR as a blend of the conceptual space of ideals of a commutative ring with unity and the conceptual space of the former second conceptual space where the axiom defining the upside-down divisibility relation is restored.

It is worth mentioning again that the definition of CDR-s was obtained after doing this implementation and therefore it could be seen as a form of "creative" result coming from the blending process.

After computing the corresponding colimit in HETS and interpreting "RingElt" as the sort containing the elements of the ring $S$, the theory defining the blend corresponds to the axioms defining a CDR (*S*), the set of all its ideals (*Generic*), the set all its prime ideals (*SimplePrime*) and a primality predicate (*IsPrime*). We present in Listing 5 just the

[3] A ring $R$ is a Containment Division Ring (CDR) if for all ideals $I$ and $J$ of $R$, $I \subseteq J$ if and only if $J$ divides $I$ (i.e. there exists an ideal $U$ such that $I = U \cdot_\iota J$).

theory corresponding to the colimit (omitting details of ring axioms and ideal generation).

```
spec SPEC =
    sorts Generic, RingElt, SimplePrime, SubSetOfRing
    sorts SimplePrime < Generic,IdGeneric < SubSetOfRing
    ops 0, 1, S : RingElt
    op __*__ : RingElt × RingElt → RingElt
    op __+__ : RingElt × RingElt → RingElt
    op __x__ : Generic × Generic → Generic
    pred IsIdeal : SubSetOfRing
    pred IsPrime : Generic
    pred __isIn__ : RingElt × SubSetOfRing
    pred gcont : Generic × Generic
    pred __generates__ : RingElt × Generic
    ∀ I : SubSetOfRing • I ∈ Generic ⇔ IsIdeal(I)
    ∀ x : Generic • x x S = x
    ∀ x : Generic • S x x = x
    ∀ A, B : Generic
    • gcont(A, B) ⇔ ∀ a : RingElt • a isIn A ⇒ a isIn B
    ∀ x, y : RingElt • x + y = y + x
    %% and further ring axioms ...
    ∀ I : SubSetOfRing
    • IsIdeal(I)
      ⇔ ∀ a, b, c : RingElt
          • ((a isIn I ⇒ a isIn S) ∧ 0 isIn I)
            ∧ (a isIn I ∧ c isIn S ⇒ c * a isIn I)
            ∧ (a isIn I ∧ b isIn I ∧ c isIn S ∧ b + c = 0
                ⇒ a + c isIn I)
    ∀ a : RingElt; A : Generic
    %% and axioms for generates and x ...
    ∀ x, y : Generic • gcont(x, y) ⇔ ∃ c : Generic • x = y x c
    ∀ p : Generic • p ∈ SimplePrime ⇔ IsPrime(p)
    ∀ p : Generic
    • IsPrime(p)
      ⇔ (∀ a, b : Generic
          • gcont(a x b, p) ⇒ gcont(a, p) ∨ gcont(b, p))
        ∧ ¬ p = S
end
```

Listing 5: Colimit for prime ideals over CDR-s

## A Challenge Example for Blending

### Computational Creativity via Blending

The examples shown thus far in the paper have been examples of blending in mathematics whose mechanisation has helped to identify some novel and unexpected results. The blending itself was a one-stage process where human input was required to identify the input concepts. A more ambitious aim of the approach of applying blending to the problem of computational creativity in mathematics, is to allow search to be done over multiple blends and for the *process* of blending to be controlled mechanically. In this section we describe very informally a mathematical domain that seems in some ways a natural candidate for a blending approach.

### Galois Theory

Galois theory develops a relationship between a polynomial $p(x)$ with coefficients in some field $F$, the extension of $K$ of $F$ (written "$K/F$") containing all of the roots of $p(x)$ in the algebraic closure of $F$, and the group $\mathbf{Gal}(K)$ of automorphisms of $K/F$ that fix the elements of $F$. The fundamental theorem of Galois theory states that there is a bijection between the subfields of $K/F$ and the subgroups of $\mathbf{Gal}(K)$; namely, subgroups correspond to their fixed fields. Using this correspondence, properties of polynomials can be derived, most famously the fact that quintic polynomials cannot be solved by algebraic operations and the extraction of roots.

We do not propose to reconstruct much of the theory here, but note that already in this basic account there are several steps that seem compellingly "blend-like."

In the first place, for field extension, $E$ is an extension of $F$ if $F$ is a subfield of $E$. We could derive the extension relationship from the input concepts $E$ and $F$ by "taking everything additional from $E$ and adding it to $F$." This is made specific in the process of *adjoining* elements, which simply means to augment the field with all fractions of formal finite sums and products of the adjoined elements with coefficients in the base field.

Second, the notion of the *splitting field* of a polynomial, namely the special extension $K/F$ containing all of the roots of $p(x)$. This could be formed conceptually by combining the concept "*the roots of a polynomial $p(x)$ with coefficients in a field $F$*" and the concept "*a field extension $E/F$ formed by adjoining certain elements to $F$*."

As above, we could then form the concept of $\mathbf{Gal}(K)$ by blending at the conceptual level. This time, there would be several constituent pieces: "*the roots of a polynomial $p(x)$ with coefficients in a field $F$*," "*the splitting field of $p(x)$*," "*the group of automorphisms of a field extension $E$*," "*the automorphisms that fix $F$*."

Finally, assuming that we have built $\mathbf{Gal}(K)$ in this fashion, we would like to know some of its properties. Consider the claim that *elements of $\mathbf{Gal}(K)$ permute the roots of $f$*. This time, instead of being purely conceptual, we want to work at the *process* level, and consider before-and-after descriptions of the result of applying $\varphi \in \mathbf{Gal}(K)$ to some $r$ with the property $p(r) = 0$. This is similar in some ways to the "Riddle of the Buddhist Monk", popularised by Koestler (1964), which is cited as an example of the power of blending.[4] However, this time the generic space is not a simple geometric machine, but rather an algebraic machine with several moving parts.

The proof of the claim is as follows. If $p(r) = 0$, then $\varphi p(r) = \varphi 0$. Since $\varphi$ is an automorphism, $\varphi 0 = 0$; and furthermore $\varphi$ distributes over the sums and products that make up the polynomial $p(x)$ and fixes its coefficients, therefore $\varphi p(r) = p(\varphi r)$. Chaining the equalities together, we have $p(\varphi r) = 0$.

[4] "A Buddhist monk begins at dawn one day walking up a mountain, reaches the top at sunset, meditates at the top for several days until one dawn when he begins to walk back to the foot of the mountain, which he reaches at sunset. Making no assumptions about his starting or stopping or about his pace during the trips, prove that there is a place on the path which he occupies at the same hour of the day on the two separate journeys."

In short, the proof is a fairly direct result of combining the definitions. Goguen (1992) suggests that "combination is colimit." Can we realise the proof through (one or several) colimit operations? And is there anything special about this proof? Apart from these more theoretical questions, the foregoing discussion raises the following technical issues:

**Field Extension** When reasoning about polynomials, it is useful to distinguish the three separate types – those of $E$, those of $F$ and those of $E/F$ as a supertype. Using blending machinery removes the distinction between these types.

**Splitting Field Extension Theorem** A challenging but creative step is to discover the theorem that extending $F$ *only* with the roots of $f(x)$ forms a field.

**Automorphisms** As mentioned in the background section, currently there is no way of computing colimits if automorphisms are characterised in higher-order logic. An alternative specification, or an implementation of colimit computation for higher-order logic is needed.

## Evaluation and Outlook

### Review of the current offering

(a) We began the paper with the reconstruction of certain mathematical objects, showing the technical feasibility of the approach.

(b) The more advanced example at the centre of the paper illustrates how this sort of reconstruction relates to mathematical practice.

(c) A future-oriented example exposes some technical challenges, while suggesting that blending could offer a novel approach to computer mathematics.

### Broader issues in evaluation

In addition to motivating a further investigation of the role blending can play in proofs, Galois theory, discussed above, is paradigmatic for other reasons. This discussion draws on the early 20th Century writings of Albert Lautman on the philosophy of mathematics and the subsequent interpretation of this work by Gilles Deleuze. It uses these ideas to propose an approach to embedding evaluation within the system itself.

Concerning the common features of Galois theory, class field theory, and the development of the universal covering surface in Riemann geometry, (Lautman, 2011, p. 126) writes:

> What is characteristic of the movement of the theories that will be considered here is the existence of an end conceived in advance as a term of the ascent.

This is reminiscent of our notion of internal evaluation that apply to the blend. To illustrate, let us briefly imagine how we would use blending techniques to move from porcupine+lion to the perfected *porculione*. Here, instead of field automorphisms that preserve mathematical structure and fix certain designated elements, we would look for mappings that preserve other properties that exist in the underlying domain. Porculiones would presumably have four feet, would be mammals, and would be omnivores; they should also be viable living creatures.

(Deleuze, 1994, pp. 227–228) follows Lautman in enthusiastically endorsing the Galoisian approach to mathematics:

> [T]he fact that an equation cannot be solved algebraically, for example, is no longer discovered as a result of empirical research or by trial and error, but as a result of the characteristics of the groups and partial resolvents which constitute the synthesis of the problem and its conditions (an equation is solveable only by algebraic means – in other words, by radicals, when the partial resolvents are binomial equations and the indices of the groups are prime numbers). The theory of problems is completely transformed and at last grounded, since we are no longer in the classic master-pupil situation where the pupil understands and follows a problem only to the extent that the master already knows the solution and provides the necessary adjunctions. For, as Georges Verriest remarks, the group of an equation does not characterise at a given moment what we know about its roots, but the objectivity of what we do not know about them. Conversely, this non-knowledge is no longer a negative or an insufficiency but a rule or *something to be learnt* which corresponds to a fundamental dimension of the object.

Although there is a commonality between blending and the Galoisian approach insofar as progressive refinement carries us toward a "perfected" conclusion, Deleuze's enthusiasm about the pedagogical situation would be significantly cooled here. It would seem, in many of our examples, that we only make progress "to the extent that the master already knows the solution and provides the necessary adjunctions."

However, this apparent infelicity may be less of a thick obstacle than it would initially appear. What seems to be most needed is a notion of a *question* inside the system. This would recover Lautman's basic thrust: "Scientific or not, every question has built in some assumptions about the form of the answer" (Larvor, 2011). In short, an experimental approach in which the system *asks* and *answers* questions would embed key aspects for evaluation in the system itself.

### Future work

The idea of using blending to carry out steps in a proof would provide a useful training ground for further development. The primary problem is: If blending is the realisation of "combinatorial creativity" how will we avoid being swamped by the combinatorial explosion of possible things to combine? The first challenge is thus fitting different mathematical components together in a sensible manner. A related challenge would apply when modifying the system to selectively experiment with the rules it uses. The objective in this case would be for the system to learn to associate different (useful) techniques with different types of problems.

## Conclusions and Remarks

The examples presented in this paper trace the development of the blending approach. The current paper begins with reconstructions, but also quickly shows how computed blends

can suggest new mathematical definitions and concepts of interest to practising mathematicians. The analysis offered here shows that this work is a building block that will be useful for future developments that are able to reason more flexibly about mathematical problems – and systematically find and propose new concepts and problems.

In future work, we will look more at the cognitive issues raised in this work. In particular, the use of *image schemas* can give a link between the computational and representational approach taken here, and the cognitive claims coming from authors such as Fauconnier and Turner, and Johnson. Here the work of Mandler and Canovás (2014) and Hedblom, Kutz, and Neuhaus (2014) gives an idea of how these underlying cognitive primitives can be expressed in logical form, and can thus play an explicit role in our modelling of creativity in mathematics.


## Acknowledgements

The authors are grateful to the referees for constructive comment. The project COINVENT 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-Open Grant number: 611553.


## References


Alexander, J. (2011). 'Blending in mathematics'. *Semiotica*, *2011*(187), 1–48.

Arbib, M. A., and Hesse, M. B. (1986). *The construction of reality*. Cambridge, England: Cambridge University Press.

Astesiano, E. et al. (2002). 'CASL: the common algebraic specification language'. *Theoretical Computer Science*, *286*(2), 153–196.

Boden, M. A. (1977). *Artificial intelligence and natural man*. Harvester Press.

Deleuze, G. (1994). *Difference and repetition*. Translated by Paul Patton. London: Bloomsbury Academic.

Eisenbud, D. (1995). *Commutative algebra with a view toward algebraic geometry*. Graduate Texts in Mathematics. Springer.

Fauconnier, G., and Turner, M. (1998). 'Conceptual integration networks'. *Cognitive Science*, *22*(2), 133–187. Extended version 2001 on-line.

Fauconnier, G., and Turner, M. (2002). *The way we think: conceptual blending and the mind's hidden complexities*. Basic Books.

Goguen, J. A. (1992). 'Sheaf semantics for concurrent interacting objects'. *Mathematical Structures in Computer Science*, 159–191.

Goguen, J. A. (1999). 'An introduction to algebraic semiotics, with application to user interface design'. In *Computation for metaphors, analogy, and agents* (Vol. 1562, pp. 242–291). LNCS. Springer.

Goguen, J. A. (2005). 'What is a concept?' In *Conceptual structures: common semantics for sharing knowledge* (Vol. 3596, pp. 52–57). LNAI. Springer.

Grothendieck, A., and Dieudonné, J. (1971). *Eléments de géométrie algébrique I* (seconde édition). Springer.

Gust, H., Kühnberger, K.-U., and Schmidt, U. (2006). 'Metaphors and heuristic-driven theory projection (HDTP)'. *Theoretical Computer Science*, *354*, 98–117.

Hedblom, M., Kutz, O., and Neuhaus, F. (2014). 'On the cognitive and logical role of image schemas in computational conceptual blending'. In A. Lieto et al. (Eds.), *Proceedings of the second international workshop on artificial intelligence and cognition (AIC 2014)* (Vol. 1315, pp. 110–121). CEUR Workshop Proceedings.

Koestler, A. (1964). *The act of creation*. Hutchinson.

Kutz, O., Neuhaus, F., Mossakowski, T., and Codescu, M. (2014). 'Blending in the Hub – towards a collaborative concept invention platform'. In *Proceedings of the fifth international conference on computational creativity, ICCC 2014*.

Lakoff, G., and Núñez, R. (2000). *Where mathematics comes from: how the embodied mind brings mathematics into being*. New York: Basic Books.

Larvor, B. (2011). 'Albert Lautman: Dialectics in Mathematics'. *Foundations of the Formal Sciences VII*.

Lautman, A. (2011). *Mathematics, ideas and the physical real*. Translated by Simon Duffy. A&C Black.

Mandler, J. M., and Canovás, C. P. (2014). 'On defining image schemas'. *Language and Cognition*, *6*, 510–532.

Martinez, M. et al. (2014). 'Algorithmic aspects of theory blending'. In *12th international conference on Artificial Intelligence and Symbolic Computation*. Sevilla, Spain.

Mossakowski, T., Maeder, C., and Lüttich, K. (2007). 'The Heterogeneous Tool Set'. In O. Grumberg, and M. Huth (Eds.), *Tacas 2007* (Vol. 4424, pp. 519–522). Lecture Notes in Computer Science. Springer.

Núñez, R. (2005). 'Creating mathematical infinities: the beauty of transfinite cardinals'. *Journal of Pragmatics*, *37*(10), 1717–1741.

Pereira, F. C. (2007). *Creativity and artificial intelligence: a conceptual blending approach*. Applications of Cognitive Linguistics. Mouton de Gruyter.

Schmidt, M. (2010). *Restricted higher-order anti-unification for heuristic-driven theory projection* (PICS-Report No. 31-2010). Univ. Osnabrück. Germany.

Steiner, G. (2001). *Grammars of creation*. London: Faber and Faber.

Turner, M. (2005). 'Mathematics and narrative'. In *International conference on mathematics and narrative*. Mykonos, Greece.

Turner, M. (2014). *The origin of ideas: blending, creativity and the human spark*. Oxford: OUP.

Weil, A. (1960). 'De la métaphysique aux mathématiques'. *Sciences*. in (Weil, 1979, pp 408–412).

Weil, A. (1979). *Œuvres scientifiques/collected papers*. Corrected second printing. New York: Springer.

# Unweaving The Lexical Rainbow:

## Grounding Linguistic Creativity in Perceptual Semantics

**Tony Veale[1], Khalid Alnajjar[2]**
[1]School of Computer Science and Informatics, University College Dublin, Belfield D4, Ireland.
[2]Department of Computer Science, University of Helsinki, Finland.

**Abstract**

The challenge of linguistic creativity is to use words in a way that is novel and striking and even whimsical, to convey meanings that remain stubbornly grounded in the very same world of familiar experiences as serves to anchor the most literal and unimaginative language. The challenge remains unmet by systems that merely shuttle or arrange words to achieve novel arrangements without concern as to how those arrangements are to spur the processes of meaning construction in a reader. In this paper we explore a problem of lexical invention that cannot be solved without an explicit model of the perceptual grounding of language: the invention of apt new names for colours. To solve this problem we shall call upon the notion of a *linguistic readymade*, a phrase that is wrenched from its original context of use to be given new meaning and new resonance in new settings. To ensure that our linguistic readymades, which owe a great deal to Marcel Duchamp's notion of *found art*, are anchored in a consensus model of perception, we introduce the notion of a lexicalized colour stereotype.

## Call me but [X], and I'll be new baptized

*What's in a name? that which we call a rose*
*By any other name would smell as sweet;*

-- Juliet, in William Shakespeare's *Romeo and Juliet*

Shakespeare wrote that a rose by any other name would smell just as sweet. From a chemical perspective he was certainly correct: a rose retains all of its olfactory qualities no matter what we choose to call it. Yet as a talented poet, Shakespeare often exploited the power of words to evoke fond memories, to arouse the imaginations and to stir the emotions of his audience. It is certainly true that the word "*rose*" obtains its warm associations and poetic resonance from its perceptual qualities – its deep red color, silky texture and sweet fragrance – but it is surely just as true that this flower would not be so beloved of poets if its established name were a lexical eyesore like "*goreweed*", "*bloodwort*", "*thorngore*," "*prickstem*" or "*turdblossom*."

Names are important. We choose them not just to serve as unique identifiers, but as evocative signs that are more than mere symbols. Steve Jobs chose the name "*Apple*" for his new technology venture to exploit the wholesome familiarity of its conventional meaning, a ubiquitous fruit that is seen as natural, attractive and unthreatening. *Apple Corp*. continues to make good use of this naming motif in its products, ranging from the Apple GS (nicknamed the *Granny Smith*) to the Apple *Macintosh* (a type of apple) to the Apple *Newton* (referencing both the popular myth of Isaac Newton and the falling apple than inspired him, and a fruit-filled cookie that is popular with children). The technology company *Sun Microsystems* chose its name to be a signifier of light, solidity and power, while *Oracle* chose its name to evoke all that is wise and knowledgable. *Cisco* is evocative of the freedoms one associates with the company's home city, San Francisco, while *Google* has benefited from seeing its name go from being a noun (a static thing) to a verb (a dynamic action). A good name cannot save a bad product, but it can help to make a good product great. Conversely, a poor choice of name can only add to the woes of a weak product. Though there are surely many reasons for the failure of Microsoft's "*Zune*", the fact that so many who care to speak of it can only remember the product as *Microsoft's answer to the iPod* suggests that its name was a big part of the problem.

We also use names to divide up the colour spectrum into shareable bundles of perceptual experiences. We all know what is meant by the words "*red*" or "*green*" but we also appreciate that such simple names subsume a wealth of possible tones and tints. Insofar as each color variant has its own uses, it deserves its own name. The *Pantone* company, a provider of colour palettes to industry, uses functional alphanumberic names for its many variations. Poets are more evocative, and anchor their chosen names in our shared experiences of a shared physical world. So when, in the Iliad, Homer describes the colour of morning light with the epithet *rosy-fingered dawn*, he succeeds in conveying a very specific shade of red by grounding his description in the familiar colour stereotype of the *rose*. A lexical stereotype is any lexicalized idea that can evoke a range of qualities, perceptual or otherwise. But one must be careful when using such dense descriptors. Homer's frequent use of the epithet "*wine-dark sea*" has led many a scholar to the edge of rational explanation, to question not just Homer's visual sense (he is traditionally believed

to have been blind, if indeed he was a single individual), but also ancient nautical conditions (e.g. to posit red tides, dense with rust-hued algae) and even the colour of ancient Greek wine (dark blue, perhaps, if heavily diluted with alkaline water). Yet the simplest answer is that which does not ask us to question our colour stereotypes: Homer really did mean to imply that the sea – at dusk, under an auspicious red sky – looked as dark and *red* as red wine.

With creativity we aim to be fresh and orginal, yet it is familiarity that lies at the heart of creativity. Conversely, it is obviousness, not familiarity, that is the antithesis of creativity, for to be creative one must knowingly exploit familiar ideas in non-obvious ways. Indeed, psychologists have long argued that a grounding in familiar stereotypes should guide the appreciation of new ideas, leading Giora *et al*. (2004) to advance, and empirically verify, the theory of *Optimal Innovation*. This theory argues that novelty is, in itself, neither sufficient for creativity nor a reliable benchmark of creativity. For Giora, an optimal innovation is any novel turn that contains the recognizable seeds of its familiar origins, as when a witty phrase is seen as a clever variation on a familiar expression, or a novel name can be decomposed into familiar elements. A colour name such as *Jealous Monster*, for a shade of green, would be an optimal innovation in this sense if it is appreciated as a variation on Shakespeare's *Green ey'd monster, jealousy*. So too are technology names that knowingly borrow – in the fashion of *Apple Corp*. – from the world of fruit. Thus, *BlackBerry* and the *Raspberry Pi* each nod to *Apple Corp*. while emphasizing their berry-like *petiteness*.

For a modern connoisseur of colours and colour names, a paintshop catalogue proves to be a more diverse source of evocative names than a book of verse. After all, paint manufacturers have a vested interest in selling more than emulsified RGB codes. So like poets, paint makers craft names that are dense in emotion and poetic resonance, to sell an entire colour "experience" to aspirational buyers. Why else name a paint colour *Soho Loft* or *Eton Mist*? The colour spectrum is free, and available to anyone with eyes, while paint makers all have access to much the same technologies. But names add value that can make a colour desirable, allowing manufacturers to sell feelings in a can. Paint catalogues are thus filled with colour names such as *Mocha Cream*, *Oyster Shell*, *Harvest Sun*, *Toffee Crunch*, *Vintage Plum* and *Almond Butter*, each a name that can stir the appetite as much as the imagination. Paint makers compete to find the most marketable names for what are virtually the same RGB codes, so that one maker's *Pale Liqueur* is another's *Baked Biscotti* or *Crème Caramel*.

Our colour preferences serve as superficial expressions of deeper personality traits, or at least we feel this to be so when we stake out claims to favorite colours or ask others about theirs. On Twitter, an automated bot that generates a random RGB code and a corresponding colour swatch every hour has attracted almost 30,000 human followers. The outputs of this Twitterbot, named *@everycolorbot*, are frequently favorited and re-tweeted, not because users are drawn to specific RGB hexcodes, but because of what the corresponding colours say about their own aesthetics. Similarly, the website *colourlovers.com* invites its users to express their *loves* for (i.e., to vote for) specific colours and RGB codes. Users of the site may also invent their own names for specific codes, and cluster these codes into recommended palettes. Rather like a vast paint catalogue, the site is a trove of insightful data on the creative naming strategies we humans use to lexicalize our favorite hues.

In this paper we seek to automate the creative task of inventing new names for specific colours and RGB codes. The task is interesting not just because humans find it so, or because name invention is a creative industry in itself; rather, the task interests us here primarily because it offers us a framework to explore issues of perceptual grounding in linguistic creativity. Much like *@everycolorbot*, our solution is implemented as an autonomous bot on Twitter. Yet this new Twitterbot is not a mere generator of random RGB codes, but an inventor of meaningful, perceptually-grounded names for its chosen colours. These names are grounded via a large inventory of colour stereotypes, and this database of stereotypes constitutes a reusable result of this research that we make available to others. To ensure that all names are semantically and syntactically well-formed as linguistic constructs, we also exploit the notion of a *linguistic readymade*, a Duchampian idea in art in which something – a physical object or even a phrase – is taken from its conventional context of use and placed in a new context that gives it new meaning and new relevance.

## The memory be green, and that it us befitted

There is both a science and an art to creative naming (see Keller, 2003), for though we want our new names to seem effortlessly apt, their creation often requires considerable amounts of search, filtering, evaluation and refinement. So while inspiration can arise from almost any source, a small number of reliable generative strategies dominate. Punning, for instance, is popular as a naming strategy for non-essential services or products that exude informality. Puns thus proliferate in the names of pet shops and pet services (e.g., *Indiana Bones and the Temple of Groom*, *Hairy Pop-Ins*), hair salons (*Curl Up & Dye*), casual food emporia (*Thai Me Up*, *Jurassic Pork*, *I Feel Like Crêpe*, *Custard's Last Stand*, *Tequila Mockingbird*) or any small business that relies on a memorable hook to direct future footfall (*Lawn Order*, *Sew It Seams*, *Sofa So Good*). As innovations, punning names are optimal in the sense of Giora *et al*. (2004), insofar as they ground themselves in the cozy familiarity of an idiom ("*so far, so good*") or a popular TV show ("*Law and Order*") or a film ("*Indiana Jones and the Temple of Doom*") and give their audience the thrill of recognition when first they encounter them. Computational Creativity (CC) has had notable successes with punning (Binsted and Ritchie, 1997; Hempelmann, 2008), leading Özbal and Strapparava (2012) to obtain promising results for a pun-based automated naming system. With tongue placed firmly in anesthetized cheek, these authors suggest that the punning name *Fatal Extraction* might be used to add humour to a dentist's

advertisement, or that a vendor of cruise holidays might find use for a slogan like *Tomorrow is Another Bay* (though not *Die Another Bay*).

Newly invented names may often take the form of new words, or *neologisms*. One especially productive strategy for neologism creation is the *portmanteau* word, or formal blend, in which a new word is stitched together from the lexical clippings of two others. A good *Frankenword* (the word is itself a portmanteau of "Frankenstein" + "*word*") will contain identifiable components of both ingredients, as in "*spork*" ("spoon"+"fork"), "*brunch*" ("breakfast" + "lunch") or "*digerati*" ("digital"+ "literati"). Veale (2006) presents an automated approach to harvesting neologistic portmanteaux from Wikipedia and for assigning plausible interpretations using the site's link topology. For instance, as the *Feminazi* Wikipedia page links to that of *feminist* and *Nazi*, and each denotes a kind of person, a "*Feminazi*" is assumed to be a formal blend of a feminist and a Nazi. Butnariu and Veale (2006) later describe a system, named *Gastronaut*, that invents and evaluates its own neologistic portmanteaux, by combining morphemes of Greek origin (e.g. "*gastro-*", "*-naut*") to which it assigns lexical glosses (e.g. "gastro-"→*food*, "-naut→*traveller*|*explorer*). As this system can propose a phrasal gloss for each portmanteau it invents (e.g. proposing "*food traveller*" for *gastronaut*), it uses the presence of this phrase on the Web to validate the linguistic usefulness of the corresponding neologism.

Özbal & Strapparava (2012) use a portmanteau strategy to propose salient names for products and their qualities; e.g., their system proposes "*Televi*sun" for an extra-bright television, as *sun* is an oft-used stereotype for brightness. Smith *et al.* (2014) present a semi-automatic collaborative portmanteau creator, called *Nehovah,* that uses synonyms of the input words in its formal blends, as well as relevant phrases gleaned from sites such as *www.thetoptens.com*. This diversity of lexical sources allows *Nehovah* to invent portmanteau words that do not contain clippings from *any* of its inputs, but to clip words that are nonetheless salient. Özbal and Strapparava also use word associations in their formal blends, to propose names such as *Eatalian* ("Eat" + "Italian") and *Pastarant* ("Pasta" + "Restaurant") for Italian eateries, the first of which names a real restaurant.

Creative naming, like modern art, is often a matter of wholesale appropriation: we reuse an existing product that is not itself original, but use it in a new context that makes it fresh again. Consider the name *Fifty Shades of Grey* for a hair salon that aims to imbue dye jobs with sex appeal, or the name *The Master and Margherita* for a pizzeria. The movie *The Usual Suspects* takes its striking title from an immensely quotable line from the movie *Casablanca*, the film *Pretty Woman* takes its title from a song by Roy Orbison, while the movie *American Pie* is named after a song by Don McLean. Veale (2012) refers to this kind of appropriation as a *linguistic readymade*, after the *found art* movement launched by Marcel Duchamp in 1917 with his *Fountain* – a signed urinal exhibited as a work of art.

Veale (2011,2012) generalizes this approach to creative text appropriation into a computational paradigm named CIR: *Creative Information Retrieval*. CIR is based on the observation that much of what is deemed creative in language is either a wholesale reuse of existing linguistic forms – *linguistic readymades* – or a coherent patchwork of modified readymades. CIR provides a non-literal query language to permit creative systems to retrieve suitable readymades with appropriate meanings from a corpus of text fragments such as the Google n-grams (Brants and Franz, 2006). For example, the CIR query operator *@Adj* matches any word/idea that is stereotypically associated with the property *Adj*, and so the query *"@cold @cold*" retrieves bigrams whose first and second words denote a stereotype of coldness, such as "*robot fish*" or "*January snow*". The retrieved phrases may never have been used figuratively in their original contexts of use, but they can now be re-used to evocatively convey coldness in novel witticisms, similes and epithets. Veale (2011) uses CIR as a flexible middleware layer in a robust model of affective metaphor interpretation and generation that also combines metaphors to generate poetry. Veale (2012) uses CIR in a generative model of irony, to invent ironic similes such as "*as threatening as a wet whisper*" and "*as strong as a cardboard tank*"). A key advantage of using linguistic readymades for automated invention – perhaps the single biggest reason to exploit readymades – is that, as phrases, their syntactic and semantic well-formedness has already been well-attested in the outputs of human authors.

We exploit CIR middleware here as a means of finding readymade colour names in the Google n-grams. That is, we seek out attested phrases that may evocatively suggest a colour, regardless of whether these phrases were ever used to name a colour in any of their original contexts of use (which, of course, an n-gram model cannot tell us). We use a large inventory of lexicalized colour stereotypes to permit CIR to find these candidate phrases, and employ a mapping from stereotypes to RGB hexcodes to derive a composite colour from their individual colour ingredients. Having established a mapping from colour readymades to colour codes, a perceptual Twitterbot can then creatively name the colours it wishes to showcase in its tweets.

## If Snow Be White

CIR offers users a range of non-literal query operators, of which **@** is perhaps the most useful for metaphor retrieval but also the most knowledge-dependent. For **@** is only as useful as its stock of stereotypical associations – such as that *fridges*, *winter*, *fish* and *ice* are each *cold* or that *suns*, *flames*, *ovens* and *deserts* are all *hot* – will allow. Veale (2013) outlines a semi-automated approach to acquiring these associations from similes found on the Web, such as "*hot as an oven*" and "*as cold as winter*". While a number of these similes identify popular colour stereotypes, such as that lemons are yellow ("*as yellow as a lemon*"), night is black, grass is green and snow is almost always white, we require a considerably more substantial inventory of colour stereotypes if we are going to extract a diversity of readymade colour names from the Google n-grams.

Basic colour words like "*red*" and "*blue*" are often used

as simple, descriptive adjectives, while more subtle hues call for longer adjectival forms. For example, hyphenated compounds, such as "*cherry-red*" and "*nut-brown*", are commonplace in English and easily harvested from Web texts or from large databases of Web n-grams. Consider the following matches for the CIR query "^*noun* - red" in the Google 3-grams (^*noun* matches any noun):

*blood* - red (3-gram frequency: *57,932*)
*ruby* - red (3-gram frequency: *16,366*)
*cherry* - red (3-gram frequency: *15,667*)
*rose* - red (3-gram frequency: *14,513*)
*brick* - red (3-gram frequency: *11,676*)
*flame* - red (3-gram frequency: *2,874*)
*coral* - red (3-gram frequency: *2,371*)

Each of the nouns in the modifier-first position above denotes a familiar stereotype of *red*ness. But the 3-grams also provide problematic matches, such as the following:

*tallahassee* - red (3-gram frequency: *172,082*)
*lemon* - red (3-gram frequency: *5,486*)
*mahogany* - red (3-gram frequency: *1,029*)

*Tallahassee*, a place name, does not denote a stereotype of redness in the same way as e.g., the place name *Mars*. Rather, it is a conventionalized name for a specific shade of red, while *lemons* have no association at all with *red* in the popular imagination. *Lemon-red* most likely denotes a blend then, of *red* and lemon-*yellow*, rather than the name of a stereotypical source of redness. It takes knowledge of the world to distinguish such n-grams – undesirable *near misses* – from the desirable *hits* of earlier n-gram matches.

We broaden our n-gram retrieval net by using the CIR query "^*noun* - ^*colour*", where ^*noun* matches any noun and where ^*colour* matches any member of the set {*red, blue, green, yellow, orange, brown, purple, black, white, grey, pink*}. To keep the hits, such as *coral-red*, and to discard the misses, such as *lemon-red*, we must manually filter all retrieved matches. Since our aim is to construct a high-quality resource with extensive reuse value, manual filtering is a good investment of effort. We think it is better to build a near-perfect resource with manual effort than to design a one-off learning algorithm that would do the job imperfectly yet take longer to implement and test. A day of manual effort yields a filtered set of 801 compound adjectives, ranging from *acid-green* to *zinc-white* with hues such as *sulfur-yellow*, *tandoori-red* and *whale-blue* in between. But a more arduous task awaits.

We must now assign a representative RGB code to each colour stereotype. For instance, we assign #E53134 to *tandoori-red* but #FD5E53 to *sunset-red*. This mapping of colour stereotypes to colour hexcodes provides the perceptual grounding for each stereotype and so must be performed with great care. The *encycolorpedia.com* site and others are used to explore possible RGB codes for each stereotype, and human judgment is used in each case in the selection of the most apt colour code. We use RGB as a coding system for its popularity and simplicity, as RGB codes can later be converted into one's preferred coding scheme, such as LAB (see Hunter, 1948), whose dimensions offer a better of model of human perception. The result of this manual effort is a map that associates each of our 801 colour stereotypes with an apt RGB code.

## And summer's green all girded up in sheaves

These lexicalized stereotypes are the building blocks with which we can build novel colour names. Conversely, they are the identifiable signifiers of colour that we can use to recognize the potential of arbitrary readymades to suggest and name specific colours. As noted earlier, we choose to view the invention of colour names as a *readymade art* task, in which coherent, existing phrases are ripped from their original contexts of use – where they are unlikely to name a colour – and given new life as apt colour names.

For CIR purposes, we construct the ad-hoc set ^*stereo* to hold the names of all of our colour stereotypes, from *acid* to *zucchini*. The simple CIR query "^*stereo* ^*stereo*" can now retrieve all bigram phrases from the Google n-grams in which both modifier and head suggest a colour. Consider the matching bigram "*chocolate espresso*" (freq =2,548). As the stereotype *chocolate*-brown maps to the RGB code #7B3F00, and the stereotype *espresso*-black maps to #393536, a creative system can infer that the colour named by "*chocolate espresso*" will have an RGB code that sits somewhere on the line connecting #7B3F00 to #393536 in RGB space. Veale (2011) demonstrates how phrases like "*chocolate espresso*" are retrieved from the Google n-grams because the stereotypes for *chocolate* and *espresso* have shared properties, such as *smooth* and *dark*, allowing a system named the *Jigsaw Bard* to invent the simile "*as smooth and dark as a chocolate espresso*." In effect, what we aim to achieve here is the generation of novel similes that have discernible perceptual groundings.

The CIR query "^*stereo* ^*stereo*" retrieves 5,841 bigram phrases from the Google 2-grams, from "*lemon tree*" (frequency="3,236") and "*honey mustard*" (freq=3,120) to "*Brick Park*" (freq=40) and "*Bear Shadow*" (freq=40). When this query is applied to the Google 1-grams – by splitting complex unigrams into their lexical parts – an additional 5,666 unigram readymades are found, ranging from "*honeymoon*" (frequency=2,410,981, which may be interpreted as a pale blend of *honey-yellow* and *moon-white*) to "*firemelon*" (freq=200, perhaps naming a blend of *fire-red* and *melon-orange*). The least frequent names also tend to be the most enigmatic. Consider "*braincloud*" (freq=201), which suggests a striking name for a shade of gray, or "*demonmilk*", "*coralstar*" and "*bananadragon*". These seem to have been crafted by another person in another context to name some idea or thing; now they can be used again, this time to provocatively name a colour.

These readymades are not manually filtered for quality, and so, as CIR cannot disambiguate word-senses in n-grams, it may retrieve phrases that use colour stereotypes in non-stereotypical senses. For instance, CIR retrieves

"*Holly Hunter*" (an actress, but also a potential blend of holly-*red* and hunter-*green*) and "*Tiger Woods*" (a famous golfer, but also, potentially, a tawny blend of tiger-*orange* and wood-*brown*). Recall that the ultimate artistic value of a readymade lies in its ability to be re-interpreted with a new meaning or a new resonance. An orange-brown colour named *Tiger Woods* would be not just apt then, but humorously apt, and we should embace this serendipity.

Each readymade can be assigned a potential RGB code at its moment of retrieval, by employing a parameterized mixture model to the RGB codes of its lexical ingredients. For a readymade like "*chocolate espresso*", whose words denote nearby points in RGB space, we can simply split the difference and average the colours, so that chocolate espresso is a mix of 50% chocolate-brown (#7B3F00) and 50% espresso-black (#393536). When these components denote more distant colours/codes, it is necessary to bring linguistic *and* perceptual intuition to bear on them. For instance, we can expect "*chocolate forest*" (freq=153) to denote a different hue than "*forest chocolate*" (freq=170). The rules of compounding suggest that "*forest chocolate*" denotes a kind of chocolate, and that its colour should be perceived as a brown hue. In contrast, as "*chocolate*" is a modifier, not a head, in "*chocolate forest*", we expect this name to denote some variation of (forest) green. As such, *forest chocolate* should contain as much *forest-green* as one can put into it while keeping it an identifiable brown, while *chocolate forest* should contain as much *chocolate-brown* as is possible while achieving a green hue overall.

The assignment of colours to readymade phrases is one side of the coin, of which the naming task is the flip side. Given an RGB color code, a creative naming system must assign an apt and original name to this code. This is the specific task that we focus on in this paper.

## O, speak again, bright angel!

Suppose one wanted a creative Twitterbot to respond to the postings of another bot, such as *@everycolorbot*. In this case, our responsive bot could await new tweets from *@everycolorbot*, extract the RGB code from each, and generate a catchy name for this colour to tweet as an apt response. Alternately, our bot could invent its own names for much loved colours on *colorlovers.com*, to compete with names already invented by human users of the site.

Suppose our CC bot is given the RGB code #FCF9F0, a code which corresponds to a very pale yellow hue and which, on *colorlovers.com* has received 69 loves (and the name "*vanilla ice cream*" from one of the site's users). The colours of the RGB space can be arranged on a *colour wheel* (see Jennings, 2003), in which the three primaries (Red, Green and Blue) are found at equidistant points on the circumference of a circle, with all possible secondary and intermediate colours arranged between the corresponding points for their color ingredients. Locating #FCF9F0 on the colour wheel, we consider this to be the dominant colour in a scheme of three colours, comprising this and its two near-neighbors, #FCF3F0 and #F9FCF0. This arrangement is called an *analogous* colour scheme (Pentak, 2010), as it forms a trio of adjacent colours that bear an analogical relationship to the related hues that one sees in nature, such as the changing colours of the leaves in Autumn. We thus refer to #FCF3F0 and #F9FCF0 as *analogous* colours of our dominant colour, #FCF9F0. A colour scheme such as this allows a CC system to find adjacent colours that appear to match well because they are often found together in the real world. Moreover, we can use a pair of analogous colours to find a readymade name for the dominant colour they bracket on the colour wheel, one that is both perceptually *and* linguistically apt.

For each analogous colour, our system seeks out the most appropriate colour stereotype. But first, we convert all relevant RGB codes into the equivalent CIE LAB code (Sharma 2003:29-32). The CIE LAB space is perceptually uniform, so any change δ in a CIELAB code induces a uniform change δ' in the perceptibility of the equivalent colour. The *Delta E CIE76* distance function can now be used to measure the distance between a given colour and that associated with any colour stereotype term. Thus, for instance, the *Delta E CIE76* distance between #FCF3F0 and *seashell-white* (#FFF5EE) is 2.17, while the distance between #F9FCF0 and pearl-white (#F7FBEF) is 0.55. As it happens, these two stereotypes – *seashell-white* and *pearl-white* – are the closest available colour stereotypes for the analogous colour pair #FCF3F0 and #F9FCF0.

Multiple readymades may each combine the words "*pearl*" and "*seashell*" in various ways. But as neither of the unigrams *pearlseashell* or *seashellpearl* is attested in the Google 1-grams, the system cannot choose a solid compound for a name. But the Google 2-grams do attest to the bigrams "*pearl seashell*" (freq=1,383) and "*seashell pearl*" (freq=5,633), and also attest to the plural bigram "*seashell pearls*" (freq=421). To maximize its chances of choosing a phrase that is semantically and syntactically well-formed, the system most prefers to choose attested unigram names, as these are most likely to have been coined as names; if it cannot find an attested unigram, it prefers a plural bigram, such as "*seashell pearls*", as these are more likely to have been coined as a *modifier:head* construction; if it cannot find an attested plural bigram, it settles for the most frequent bigram (e.g. *seashell pearl*"). In this case, it opts for the plural bigram "*seashell pearls*" and chooses its singular form, "*seashell pearl*" as a name.

A glance through any paint catalogue reveals that the most popular paint names are those that appeal to our love of nature, to our appetites, or to our aspirations. So paint names often use naming elements that denote a natural kind (*tree*, *pearl*, *forest*, *sea*, etc.), a food or drink (*toffee*, *butter*, *almond*, *espresso*, etc.) or a distinctive culture or place (*China*, *Persian*, etc.). So words such as *tandoori* and *kangaroo* tick two boxes at once. We may filter our readymade names by their adherence to this scheme, and choose only those phrases that use a colour stereotype that suggests a natural kind, food, drink, culture or place. The *Thesaurus Rex* Web service of Veale and Li (2013) can be used to provide fine-grained categorizations of colour

stereotypes (such as *kangaroo*, *butter*, *pearl*, etc.) and to filter possible readymades by the categories they evoke. The filter employed by a naming system determines its aesthetic sensibility, and different systems may exhibit different aethetic senses. One can imagine a system that prefers poetic names, smutty names, provocative names (e.g. *cocainestar* for a whiteish hue) or fantastic names (e.g. *alienbrain* for a gray-green hue). In the following experiments, our system employs the paintshop-friendly *natural-animal-food-drink-culture* filter described above.

## Beauty doth varnish age, as if new-born

To evaluate the quality and aptness of the readymade phrases that we repurpose as attractive new colour names, we compare these automatic names to those assigned by humans on the website *ColourLovers.com*. We download the top 100,000 colour codes from this site, ranked from most to least *loves*; the mean number of *loves* per colour code is 13, while each code has at least one *love* and just one human-assigned name (as the site does not permit multiple names for the same RGB code). For each RGB code our automated naming system seeks out the most apt readymade name it can find. To ensure a good perceptual match between each code and its new name, a threshold distance of 14 is chosen for use with the *Delta E CIE76* distance function, which measures Euclidean distance in the CIELAB space. Thus, the CIELAB code of any colour stereotype (such as *pearl-white*) will only match the CIELAB equivalent of an analogous RGB code (such as #F7FBEF) if their Euclidean distance in CIELAB space is 14 or less. We choose a maximum of 14 empirically, so as to impose tight control on colour matching while allowing every colour code to be assigned at least one readymade.

We automatically identify the most apt readymade for each of the 100,000 downloaded colour codes, using the preferential approach to n-gram selection outlined in the previous section. Of the 100,000 assigned names, 2587 are selected as paintshop-style names using the afore-mentioned *natural-animal-food-drink-culture* filter. It is this subset of readymade names that we focus on here for purposes of empirical evaluation. The mean number of *loves* for each of the named colours on *ColourLovers.com* is 2.188. For each of the 2,587 machine-generated names, we determine the name assigned to the corresponding RGB colour by users of *ColourLovers.com*. This allows us to construct a set of 2,587 triples, each comprising an RGB code, a human-assigned name and a name invented (via a repurposed readymade) by a machine.

We used these triples to pose comparison questions to human judges recruited via the crowd-sourcing platform *CrowdFlower.com*. For each triple, a visual sample of the colour and a pair of names, one human-generated *and* one machine-generated, were put before the judges, who were asked to take a moment to imagine the colour being used. The ordering of both names was randomly selected on a case-by-case basis, so that the human-generated name was listed first in ~50% of cases, and the machine-generated name was listed first in the other ~50% of cases. In all cases, judges were *not* told of the origin of either name. Each judge was paid a small sum to answer 4 questions:

1. *Which name is more descriptive of the colour shown?*
2. *Which name do you prefer for this colour?*
3. *Which name seems the most creative for this colour?*
4. *Why did you answer these questions they way you did?*

The fourth question is a source of qualitative responses that may, in future work, offer useful insights into the factors that shape the appreciation of names. Judges were timed on their responses, and those that spent less than 10 seconds presenting their answers for any colour were classified as *scammers* and discarded. We required that each question be answered by 5 non-scamming judges to be trusted for evaluation, and thus, we obtained 12,608 trusted judgments in all that contributed to the evaluation, and 5,040 untrusted judgments that were instead ignored.

A total of $220 was allocated to the experiment, which was terminated after these funds were exhausted and 940 judges had been paid to contribute to the task. At this point, 1578 out of 2587 colours had received five trusted judgments for each of their questions, and so it is on the collected judgments for these 1578 colours that we base our evaluation. Tallying the individual judgments per question, we see that 70.4% of individual judgments for *most descriptive name* (Q1) favored the machine; that 70.2% of individual judgments for *most preferred name* (Q2) favored the machine; and that 69.1% of individual judgments for *most creative name* (Q3) favoured the machine. Similarly, when we tally the majority judgment for each question under each colour – the choice picked by three or more judges – we see that for just 354 (23%) of the 1578 colours, a majority of judges deemed the human-assigned name for a given colour to be more descriptive than that assigned by the machine. The results for the next two questions, Q2: *which name do you prefer?* and Q3:*which name is most creative?*, are very much in line with those of the first question. Only for 355 colours does a majority of the five human judges for a given colour prefer the human-assigned name over that assigned by the machine, and only for 357 colours does a majority of judges consider the human-assigned name to be more creative than the machine-assigned name. This consistent breakdown of approx. 3-to-1 in favour of the machine suggests that machine-assigned readymade names can be more than competitive with human names.

However, the surprising consistency of these results also suggests that the human judges are really only offering *one* opinion for all three of the binary questions that they are asked. It seems that judges, who are asked to ponder the possible users of a colour before answering the questions that follow, apparently favour a given name for a colour and *then* follow through with much the same answer for all three questions. Indeed, when we calculate the rate of agreement across all questions, we find that judges choose the same name for at least two of the three questions in 93% of cases, and choose the same name for

all three of the questions (that is, most descriptive, most preferred and most creative) in 91% of cases. These agreement statistics suggest that most human judges see these questions as paraphrases of each other. Though it can aid our understanding of the mechanics of linguistic creativity to try and tease apart the related notions of descriptive adequacy, personal preference and creative appreciation, these three notions now appear to be too tightly interwound to effectively separate them, at least within the same experimental task.

## Let our bloody colours wave!

A Twitterbot named *@HueHueBot* has been constructed (by the second author) to showcase the perceptually-anchored creativity of this readymade-based approach to colour-name invention. An example tweet of this bot, with attached colour sample, is shown in Fig. 1.


Figure 1. A tweet with both RGB hexcode and apt name.

*@HueHueBot* exploits colour stereotypes and Google n-grams in the manner described in previous sections. But this inventory of colour stereotypes and their RGB codes can be reused by other Twitterbots that exhibit their own colour aesthetics and linguistic framing preferences. To this end, we gave the stereotype lexicon and a large stock of relevant n-grams to students as resources to be used for a course project on computational linguistic creativity. Students were asked to build colour-naming Twitterbots which might invent and name their own colours, or name the colour codes generated by *@everycolorbot*. The bots that ensued demonstrate a variety of possible approaches to naming and to the linguistic framing of those names.

@ColorCritics frames its outputs as though it as an art critic that specializes in colour, and thus, in addition to offering to name colours generated by *@everycolorbot*, it critiques the palette choices of this bot. @ColorCritics expresses a preference for unigram names, of which examples include *TandooriTikka*, *PukePuke* and *FireSky*. @WorldIsColored mimics the bravura personality of Stan Lee, a famous creator of comic book superheroes, and thus expresses a preference for colour names that use alliteration (a much-loved ploy of Lee's). Its alliterative colour names, such as BlueberryBlush, are framed in the language of superhero comics, such as in this tweet: "*May be coloring my costume as BLUEBERRY BLUSH was not a very good idea! RT .@everycolorbot: 0xdd4fc3*".

*@ColorMixALot* combines 2-gram phrases to generate complex colour names that run to three and four words. Example colour names include *tree frog bile yellow* and *moonlight coral pink*. The Twitterbot *@DrunkCircuit* adopts the persona of a borded worker at an IT company, and so its tweets drip with ennui and bitterness. Examples include the sarcastic riposte to *@everycolorbot* in Fig. 2.

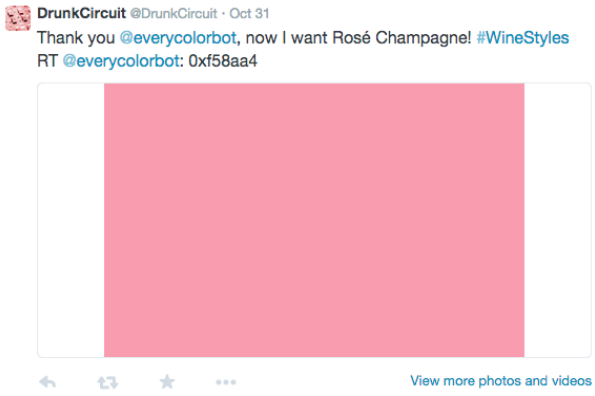



Figure 2. A sarcastic response to another colour bot: "*thank you @everycolorbot, now I want Rosé Champagne #WineStyles @everycolorbot: 0xf58aa4"*

Like *@HueHueBot*, *@DrunkCircuit* locates the category into which a new name fits best (using Wikipedia's hierarchy of topic categories), and then tailors its tweets to exploit this information. Thus, a name that denotes a kind of wine (as in Fig. 2) is affixed with the hashtag *#WineStyles*, while the name *Almond Crust* is used to anchor a tweet that insults the company canteen ("*Looks just like the Almond Crust in the canteen today. Yuck! RT @everycolorbot: 0xd3ba8f*").

*@AwesomeColorBot* also tailors its tweets to suit the category of a name, to produce outputs like that of Fig. 3.


Figure 3. A tweet with a colour, a name, and an attitude.

*@haraweq* is a colour-naming hybrid that combines elements of two popular Twitterbots, *@everycolorbot* and *@metaphorminute*. The latter is a bot by Darius Kazemi that invents random metaphor-like tweets, such as "*an evacuation is a mainframe: evergreen yet slicked*." In this vein, *@haraweq* coins colour similes, such as "*a location like a dusty taxicab RT @everycolorbot: 0xf4ec24*." It uses Wikipedia to determine e.g. that a taxicab is a location, and uses the Google n-grams to find specific combinations such as "*dusty taxicab*", which it interprets as a blend of *taxicab-yellow* and *dust-brown*.

So the most interesting colour bots do more than just invent new colour names; they find a context to motivate a new name, and then frame a tweet as an intelligent – or at least a human-like – response to this context. There is a lesson here for computational linguistic creativity. A new turn of phrase can only be considered creative in a context for which it is non-obvious and apt, and to the extent that it exercises the imagination of the reader. The imagination may take flight on the wings of whimsy, but the most compelling flights into the new and the original remain stubbornly grounded in the realm of familiar experiences.

## Acknowledgements

This research was facilitated by a travel mission funded by the Univerity of Helsinki and by the EC coordination action PROSECCO (*Promoting the Scientific Exploration of Computational Creativity*; *PROSECCO-network.EU*). The authors would like to thank Prof. Hannu Toivonen at the University of Helsinki and all the students of the 2014 UH class on Computational Linguistic Creativity (whose bots *@ColorCritics, @WorldIsColored, @ColorMixALot, @DrunkCircuit, @haraweq, and @AwesomeColorBot* are discussed here; please do check them out on Twitter). We also wish to express our gratitude to Mike Cook of Imperial College, London, who suggested the idea of a colour-naming Twitterbot, and to Hyesook Kim, who painstakingly mapped colour stereotypes to RGB codes.

## References

Thorsten Brants and Alex Franz. (2006). Web 1T 5-gram database, Version 1. *Linguistic Data Consortium*.

Kim Binsted and Graeme Ritchie. (1997). Computational Rules for Generating Punning Riddles. *HUMOR, the International Journal of Humor Research*, 10(1):25-76.

Christian F. Hempelmann. (2008). Computational Humor: Beyond the pun. In Victor Raskin (ed.), *The Primer of Humor Research*. Berlin: Mouton de Gruyter.

Rachel Giora, Ofer Fein, Jonathan Ganzi, Natalie Alkeslassy Levi, Hadas Sabah. (2004). Weapons of Mass Distraction: Optimal Innovation and Pleasure Ratings. *Metaphor and Symbol 19(2):115-141,*

Richard S. Hunter (1948). Photoelectric Color-Difference Meter. *Journal of the Optimal Society of America* 38 (7): 661.

Simon Jennings (2003). *Artist's Color Manual: The Complete Guide to Working With Color*. Chronicle Books.

Kevin L. Keller. (2003). *Strategic brand management: building, measuring and managing brand equity*. Prentice Hall.

Gozde Özbal and Carlo Strapparava. (2012). A computational approach to automatize creative naming. In Proceedings of the 50th annual meeting of the Association of Computational Linguistics (ACL-2012), Jeju Island, Korea.

Stephen Pentak. (2010). Analogous Color Scheme. *Design Basics* (8th edition.). Australia: Cengage.

Gaurav Sharma. (2003). Digital Color Imaging Handbook (1.7.2 edition). CRC Press.

Michael R. Smith, Ryan S. Hintze and Dan Ventura. (2014). Nehovah: A Neologism Creator Nomen Ipsum. *Proceedings of the 5th International Conference on Computational Creativity*, Ljubljana, Slovenia.

Tony Veale. (2006). Tracking the Lexical Zeitgeist with Wikipedia and WordNet. In Proceedings of ECAI'2006, the 17th European Conference on Artificial Intelligence, Trento, Italy.

Tony Veale and Cristine Butnariu. (2006). Exploring Linguistic Creativity via Predictive Lexicology. At the ECAI'2006 Joint International Workshop on Computational Creativity. Italy.

Tony Veale. (2011). Creative Language Retrieval: A Robust Hybrid of Information Retrieval and Linguistic Creativity. In Proceedings of ACL'2011, the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon.

Tony Veale. (2012). *Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity*. London: Bloomsbury Academic.

Tony Veale. (2013). Linguistic Readymades and Creative Reuse. *Transactions of the SDPS: Journal of Integrated Design and Process Science, 17(4):37-51.*

# FIGURE8: A Novel System for Generating and Evaluating Figurative Language

**Sarah Harmon**
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064 USA
smharmon@ucsc.edu

**Abstract**

Similes are easily obtained from web-driven and case-based reasoning approaches. Still, generating thoughtful figurative descriptions with meaningful relation to narrative context and author style has not yet been fully explored. In this paper, the author prepares the foundation for a computational model which can achieve this level of aesthetic complexity. This paper also introduces and evaluates a possible architecture for generating and ranking figurative comparisons on par with humans: the FIGURE8 system.

## Introduction

Figurative language is embedded within and intimately connected to our cultures, behaviors, and models of the world. In fact, humans use figurative language so often that we seldom realize it (Lakoff and Johnson 1980); still, its utility for communication is clear. Using metaphors and similes, one can relate the unfamiliar, or the tenor, in terms of the familiar, or vehicle (Richards 1980). In Figure 1, for example, "moon" is the vehicle for "garden", the tenor. Attributes of the moon, such as its brilliance, are used to describe the beauty of the garden. Prior to the comparison, the garden's appearance is unknown (is it beautiful and luminous, or neglected and overgrown?). The simile helps to resolve this ambiguity and provide the reader with a clearer picture of the scene.

Comparison gives us the ability to delicately express irony and sarcasm ("clear as mud"), exaggeration ("that man was as tall as a giraffe"), and emotion ("my heart was a sinking ship"). With such tools, we can explain how we feel, what kinds of people we are, and what experiences we have had. Further, metaphors give color to dry speech and are understood faster than literal equivalents (Gibbs and Nagaoka 1985); this is likely due to their appeal to common previous experiences and memories.

For the purpose of this paper, we will consider two styles of figurative language: conventional (common analogies used in daily language, such as "I see what you mean") and creative (original comparisons that call attention to themselves as figures of speech, such as "Fear is a slinking cat I find / Beneath the lilacs of my mind" (Tunnell 1977)). Each type can provide value, although previous work on computational generation of figurative language has primarily focused on understanding and reconstructing conventional metaphors and similes.

Clichés (e.g., "fast as lightning") are arguably useful when fast, informal communication is required between a computer and a human, and such phrases can be learned via web query (Veale and Hao 2007a). Generating creative comparisons on par with human authors is a much more difficult challenge. A conventional metaphor is considered "good" if many others have used it before, but uniqueness and aesthetic qualities are critical in generating a strong creative metaphor. For instance, several aesthetic properties, such as syllable counts, phonetics, stressed syllable position, rhyme, and alliteration have been identified as "obvious" criteria for making creative poetic lines sound good, despite the fact that these "do not translate well into precise generative rules" (Gervás, Hervás, and Robinson 2007). While creative generators for figurative language exist, few address this concept of what makes for a high-quality metaphor or simile. I will describe a system, FIGURE8, which contains a novel underlying model for what defines creative and high quality figurative comparisons, and evaluates its own output based on these rules.

## Related Work

Modern research in creativity has generally defined a creative system as one that generates novel, context-appropriate output (Rothenberg and Hausman 1976; Sawyer 2012). Within the context of creative natural language generation, a third criterion has been noted: a creative system must generate context-appropriate knowledge outside of its pre-existing knowledge base (Pérez y Pérez and Sharples 2004).

Several computational systems exist which attempt to meet this benchmark. ASPERA, for instance, combines case-based reasoning with intelligent adaptation of examples from corpora (Gervás 2000). Psychological theories have further informed the art of generating figurative language, resulting in more advanced and thoughtful systems. Notably, Brown (Ortony 1993) and Glucksberg (Glucksberg 2001) have argued that categorization is inherent to metaphor. As a consequence, the concept of property-based concept mapping has inspired metaphor generation approaches, and has been cited as the best method for producing robust, scalable and useful metaphors (Hervás et al. 2007; Veale and Hao 2007a).

One must also consider how to develop an appropriate knowledge base without substantial manual authoring. Previous exemplary work in metaphor generation has emphasized the power of using the web to establish example cases of valid comparisons (Veale and Hao 2007a; 2007b). However, these systems merely generate large amounts of potentially creative descriptions, and cannot distinguish between original and poor quality comparisons (Veale and Hao 2007a). Further, they often ignore context, sentence construction, and aesthetics in the generation process, resulting in less evocative and meaningful language.

FIGURE8 is a system that uses a web-driven approach to form a preliminary knowledge base of nouns and their properties. 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. In this way, FIGURE8 simulates the human-authoring process of revision by generating many vehicle choices and linguistic variations for a single tenor, and choosing the best among them as its favorite. While FIGURE8 does not claim to have a comprehensive set of rules - for example, it does not consider phonetics in its evaluation of description quality - it provides a novel foundation for an intelligent figurative language generation and assessment system.

## Approach

Prior work has established that a strong creative metaphor is not only comprehensible (Tourangeau 1981), novel (Camac and Glucksberg 1984), and context-appropriate (Harwood and Verbrugge 1977; Tversky 1977; Gildea and Glucksberg 1983), but surprising (Tourangeau 1981). The following sections will illustrate how FIGURE8 considers these properties when generating metaphors and similes. A block diagram of the generation process is shown in Figure 3.

### Clarity

A strong metaphor must have an understandable, accurate link between tenor and vehicle. A vehicle is thus only considered acceptable if it has properties in common with the tenor. Further, associating the tenor with the capacities and known manifestations of the vehicle should enhance the clarity of the description. In the FIGURE8 system, these associations are found by mining existing literary corpora (Hart 2014) for instances of the vehicle and using NLTK's parts-of-speech tagging to identify associations (e.g., refer to Figure 2). This procedure enables the system to use words commonly associated with the vehicle to develop a fresh relation to the tenor. For example, if we were to compare a teacher to a horse, FIGURE8 may now be able to reason that the teacher would prance or trot into the room. In this way, a sentence can be generated by only implicitly referring to the vehicle ("The teacher pranced into the room" vs. "The teacher was a wild horse, prancing into the room"). Common verbs, such as forms of "to be", were culled from the generated list of association because - as all nouns have the capability to exist and be - such verbs do not lend clarity to the comparison.

Granted, the word chosen to relate to the tenor may not make sense (especially in the case of verbs), destroying the very clarity it was meant to enhance. FIGURE8 thus performs a web query using Python's urllib module to ensure that others have associated the chosen word with the tenor before. If a previous association has not been made, the metaphor is ranked lower in terms of estimated clarity. This evaluation measure ensures that nonsensical descriptions, such as "The turtle darkened like a blue ocean", are given a lower ranking overall.

### Novelty

Clichés are frowned upon by expert authors; as Salvador Dalí once said, "The first man to compare the cheeks of a young woman to a rose was obviously a poet; the first to repeat it was possibly an idiot" (1968). For computer-generated text, it is thus reasonable to expect that a quality metaphor is a fresh comparison. In the FIGURE8 system, each metaphor is checked against an existing knowledge base of comparisons (Friedman 1996), and all generations are ranked based on their similarity to conventional metaphors in this database.

### Aptness

Ideally, a strong metaphor will fit the context within which it lives. For usage in a narrative context, the FIGURE8 system can be passed a model of a simple world of objects and character models, and incorporate these appropriately into its eventual output along with a prepositional phrase generation module. Additionally, one may ask FIGURE8 to generate ironic comparisons, such as those generated by a sarcastic character when speaking. Irony is achieved by selecting for properties with the exact opposite meanings, in accordance with prior work (Veale and Hao 2007b). The FIGURE8 system also endeavors to match a given context during sentence completion, which will be described in a later section.

### Unpredictability

Metaphors are perceived as cleverer when the vehicle and tenor contain similarities, but the respective domains of these terms are distinct (Tourangeau 1981). A description is thus ranked as more surprising when the words are not very conceptually similar and contain fewer properties in common. With the assumption that they share at least one property in common, the chosen metaphor components are ranked by querying the UMBC Semantic Similarity System (Han et al. 2013). The degree to which the vehicle and tenor share major categories is also considered by using a function similar to WordNet's lexname query. This check is needed because if one or more major categories are shared, the metaphor is considerably less surprising. For

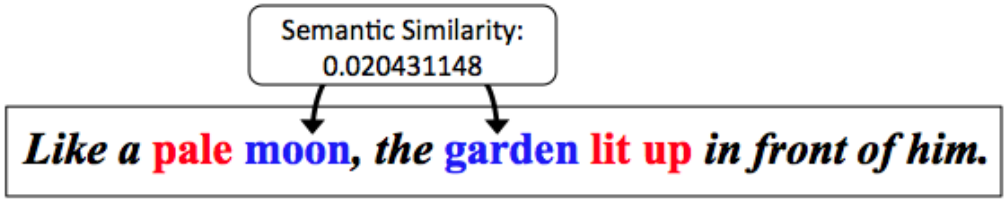



Figure 1: Example of a highly ranked output sentence by FIGURE8. Here, the tenor, vehicle, and associated phrases are *garden*, *moon*, *lit up*, and *pale*. The nouns *garden* and *moon* not only have low semantic similarity, but do not share a major category together. Likening a garden to a moon is also not a cliché comparison, lending to the description's potential novelty.

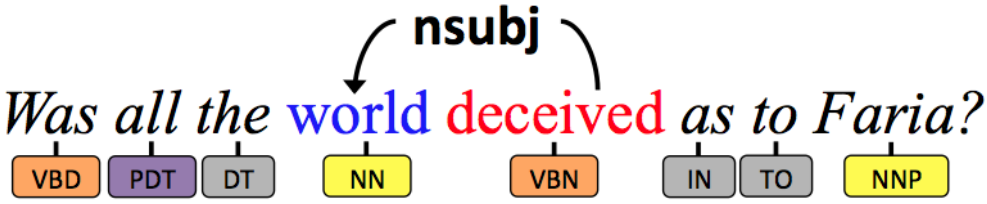



Figure 2: Example of how FIGURE8 discovers and associates a verb with a chosen vehicle, using text from *The Count of Monte Cristo*, and a part-of-speech parsing module similar to the Stanford Parser (Socher et al. 2013). Here, the *nsubj* label refers to a link between a verb ("deceived") and a noun phrase (in this case, the vehicle "world"). The remaining labels in the figure represent the part-of-speech tags.

instance, "the strawberry is a pomegranate" is considered a poor metaphor because strawberry and pomegranate are contained within a major category: fruits. Such a description may be produced by a web-based generator (for instance, the online MIT-licensed Metaphorgy system (Groff-Palermo and Lawson 2013) produces "My strawberry is a Phaeacian cherry"), but will be given a low ranking by FIGURE8.

### Prosody

The prosody of a metaphor can be defined as the rhythmic, tonal, and aesthetic qualities that distinguish one metaphor from another. Descriptions are ranked highly if their prosody is of consistent and high quality. For instance, consider the following similes:

(1) The serpent stretched into the horizon, like a deserted desert.
(2) The snake extended into the horizon, like an abandoned desert.

Although alliteration and assonance can be used beautifully in figurative language, the high similarity of consecutive words in (1) may be distracting. Example (2) depicts the same imagery, but uses words of greater distance in terms of consecutive string similarity.

At present, FIGURE8 conducts string similarity via Python's difflib to evaluate the prosody of its outputs. Using difflib's SequenceMatcher, one can determine a value indicating the degree of similarity between two input strings in a range from 0 (no similarity) to 1 (identical strings). FIGURE8 is thus able to quantify the string similarity for consecutive words, and ranks descriptions lower if there are many consecutive string similarity values above 0.7, which was deemed an appropriate threshold by the author. Consecutive words are also checked for alliteration and assonance, which are considered positive qualities by FIGURE8.

### Sentence Completion

Automated metaphor identification in text has been thoroughly explored (Neuman et al. 2013; Steen et al. 2010) and, as such, FIGURE8 has been provided with a case library of appropriate sentence constructions for metaphor and simile. By following the procedure of imaginative recall (Turner 1992), FIGURE8 first attempts to fit the provided context of the situation to an exact, pre-existing solution. If no solution exists, FIGURE8 searches its memory, solves the problem for a similar case, and adapts that solution to the provided context. As an illustration: if FIGURE8 notes that other authors have used the phrase "to the barn", it should recognize the barn as a noun denoting a man-made object via WordNet. Similarly, a "chair" is a man-made object, and thus, FIGURE8 may decide to replace "barn" with "chair" when told that a chair exists in the current narrative context. This adaptive process enables FIGURE8 to match its constructions to any provided context and complete statements creatively.

## Evaluation

Little research, if any, has worked towards developing a model of what makes a high quality computer-generated metaphor. Although there is no standard method to evaluate computationally-generated figurative descriptions, one reasonable way to judge would seem to be agreement with hu-

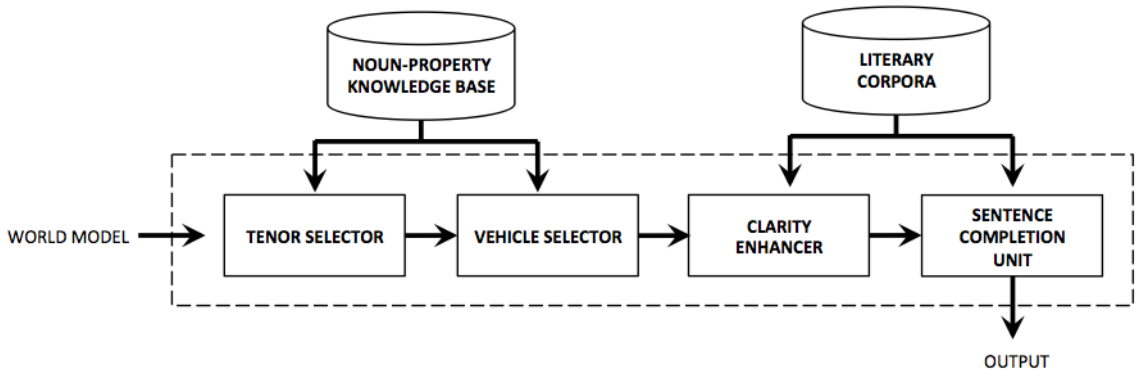



Figure 3: Block diagram of the FIGURE8 generation system. If no world model is given, a tenor is selected at random from the noun-property database. A vehicle is then selected with at least one property in common with the tenor. The Clarity Enhancer module requests verbs and adjectives associated with the vehicle from mined literary corpora. Finally, the sentence is completed by performing imaginative recall with known valid sentence constructions for metaphor identified from literary corpora.

| Generated Description | FIGURE8 Clarity | Human Clarity | FIGURE8 Likability | Human Likability |
|---|---|---|---|---|
| (A) *It was the pearl, fermenting like a wild apple.* | 3 | 2 | 3 | 2 |
| (B) *Like scenic music, the pearl danced in front of him.* | 1 | 1 | 1 | 1 |
| (C) *It was their pearl, sprawling like a wretched corpse.* | 4 | 4/5 | 4 | 3/4 |
| (D) *It was her pearl, crumpling like a drowned corpse.* | 5 | 4/5 | 5 | 5 |
| (E) *It was my pearl, bubbling like a treacherous swamp.* | 2 | 3 | 2 | 3/4 |

Table 1: Comparison 1 of FIGURE8 and human rankings for clarity and overall quality. In this set, FIGURE8 was asked to generate and rank figurative descriptions given "pearl" as the tenor. Human clarity and likability rankings were found to be highly correlated ($\rho = 0.684$). Spearman analyses also indicated positive correlations between human and FIGURE8 rankings (clarity: $\rho = 0.872$; quality: $\rho = 0.821$).

man ratings. This can be assessed by requesting humans to rank descriptions generated by the FIGURE8 algorithm, and determining if the majority are in agreement with the computer's (FIGURE8's) ranking. A pilot study indicated that providing each description with additional context would make the ranking process too time-consuming for participants. Thus, functions to enhance aptness were not included when generating outputs to be evaluated in the full-scale study.

## Method

One hundred participants (73 female, 27 male) were recruited via Amazon's Mechanical Turk. Each participant viewed a series of five sentences at a time, and were asked to rank the similes by how understandable they were (*clarity*), and by how much they, as individuals, enjoyed the comparison (*likability*). Each set of five sentences contained the same tenor, and were originally generated and ranked by FIGURE8. The sets were not hand-selected by the author. That is, the first eleven sets FIGURE8 generated and ranked were used in the study.

## Results

Human preferences were determined by following the majority criterion. As seen in Figures 4 and 5, human clarity ratings were often positively correlated with overall quality ratings, and this correlation was confirmed with Spearman analyses. Overall, FIGURE8's top result for clarity and overall quality generally agreed with the human rankings for each of the eleven sets. FIGURE8 exactly matched the first ranking 46% of the time for clarity and likability. Further, it matched either the first or second ranking 82% and 100% of the time for the clarity and likability categories, respectively. Examples of how FIGURE8 matched human ratings are shown in Tables 1, 2, and 3.

## Discussion and Future Work

In this paper, the author has introduced the FIGURE8 system as a novel tool for generating and evaluating creative figurative descriptions. FIGURE8's assessments are grounded in psychological models of metaphor comprehension, and have thus far been found to adequately match human rankings when agreed upon.

Participants in the evaluation portion were not told that the descriptions were generated by a computer. Only two

| Generated Description | FIGURE8 Clarity | Human Clarity | FIGURE8 Likability | Human Likability |
|---|---|---|---|---|
| (A) *She saw that snow rising like a leafy sun.* | 4 | 4/5 | 3 | 4/5 |
| (B) *I saw that snow flying like a voracious bird.* | 3 | 2 | 1 | 2 |
| (C) *The snow continued like a heavy rain.* | 1 | 1 | 5 | 1 |
| (D) *I saw that snow shedding like a slender moon.* | 5 | 4/5 | 4 | 3 |
| (E) *The snow falls like a dead cat.* | 2 | 3 | 2 | 4/5 |

Table 2: Comparison 2 of FIGURE8 and human rankings for sentences of tenor "snow". Human clarity and likability rankings were found to be positively correlated ($\rho$ = 0.763). Spearman correlation analysis suggested that FIGURE8 clarity rankings were positively associated with human clarity rankings ($\rho$ = 0.872), but no significant association was found between likability rankings in this case ($\rho$ = -0.359).

| Generated Description | FIGURE8 Clarity | Human Clarity (1st) | Human Clarity (1st or 2nd) | FIGURE8 Likability | Human Likability (1st) | Human Likability (1st or 2nd) |
|---|---|---|---|---|---|---|
| (A) *There was a queen, glowing like a sombre forest.* | 4 | 4/5 | 4 | 3 | 4 | 4 |
| (B) *It was their queen, flying like a white bird.* | 3 | 2/3 | 3 | 2 | 3 | 2 |
| (C) *She saw that queen strewing like a yellow flower.* | 5 | 4/5 | 5 | 5 | 5 | 5 |
| (D) *The queen blocked them, like a rugged mountain.* | 2 | 2/3 | 2 | 4 | 2 | 3 |
| (E) *The queen stands like a strong castle.* | 1 | 1 | 1 | 1 | 1 | 1 |

Table 3: A third comparison of FIGURE8 and human rankings for sentences of tenor "queen". In addition to showing first choice rankings, this table displays human rankings when considering first and second choices. That is, "the queen stands like a strong castle" was ranked as either first or second for the majority of respondents. In both cases, human clarity and likability rankings were found to be positively correlated ($\rho > 0.9$). Spearman analyses also suggested for both cases that FIGURE8 and human rankings for clarity and likability were positively correlated with high significance ($\rho > 0.7$).

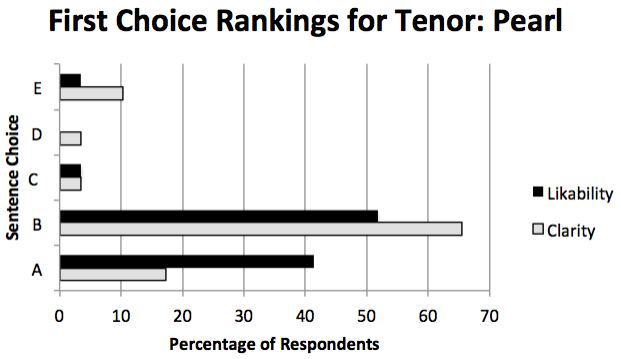



Figure 4: First choice rankings for the generated set of sentences using *pearl* as the tenor. Although some disparities existed, the majority of respondents generally agreed upon which sentence was the most understandable.

comments were made about checking sentences for validity prior to including them in the study, and one regarding how painful it was to rank "bad poetry". Most participants, however, enjoyed the task and provided positive feedback about their experience ("cool hit","super fun","I love this"). It is conceivable that task enjoyment affected user responses, but controlling for explicit indication of task enjoyment yielded no significant difference in the results. Controlling for gender also did not reveal significantly different outcomes.

Interestingly, for roughly half (50-60% per set) of the participants, how much they liked the figurative description was directly correlated with how well they understood it. The most highly ranked phrases for clarity were also often ranked first for likability, and the Spearman coefficient was used to confirm these positive associations. This was a surprising finding, because more variation and subjectivity was expected for these ratings. Discrepancies between human and FIGURE8 likability rankings, such as in Table 2, could potentially be explained by a human tendency to prefer metaphors containing words of positive sentiment value. However, more analysis is required to confirm this idea, and further study is needed to evaluate how qualities of language are weighted across general and expert populations. Judging from participant comments, it is also possible that some people may like metaphors primarily based on qualities other than clarity (such as prosody, sentiment, or whimsy). If these groups could be automatically identified, perhaps future computer-produced descriptions could adapt to generate more personalized descriptions for the optimum enjoyment of the reader.

While FIGURE8 is able to rank its figurative descriptions over various measures of quality, how well its output com-

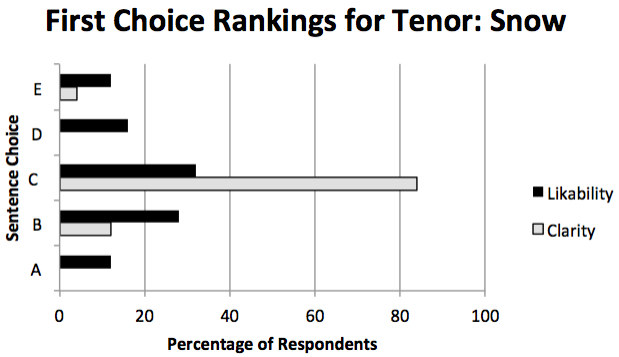



Figure 5: Clarity rankings for the generated set of sentences using *snow* as the tenor. Participants rated what FIGURE8 considered the most unsurprising metaphor as the most clear, but there was no highly significant consensus regarding the most likable description.

pares with human-authored descriptions was not assessed. The fact that most participants in the evaluation did not question the source of the texts is a promising sign that the system presented here generates human-like output. Regardless, its present constructions can be automatically assigned rankings on par with human evaluations. It is assumed that as the quality of FIGURE8's generations increases, it will be able to extract the best output from the results of its "brainstorming". Future research should build upon this foundation and work towards evaluating computer-generated descriptions in terms of aptness, prosody, and unpredictability. When machines are fully able to grasp the subtleties and aesthetics of figurative language, we as humans will be able to relate to them as never before.

## References


Camac, M. K., and Glucksberg, S. 1984. Metaphors do not use associations between concepts, they are used to create them. *Journal of Psycholinguistic Research* 13(6):443–455.

Friedman, S. M. 1996. Cliche finder. Retrieved 2 Mar 2015 from http://www.westegg.com/cliche/.

Gervás, P.; Hervás, R.; and Robinson, J. R. 2007. Difficulties and challenges in automatic poem generation: Five years of research at UCM. *e-poetry 2007*.

Gervás, P. 2000. An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-based Systems* 14:200–201.

Gibbs, R. W., and Nagaoka, A. 1985. Getting the hang of American slang: Studies on understanding and remembering slang metaphors. *Language and Speech* 28(2):177–194.

Gildea, P., and Glucksberg, S. 1983. On understanding metaphor: The role of context. *Journal of Verbal Learning and Verbal Behavior* 22:577–590.

Glucksberg, S., ed. 2001. *Understanding figurative language: From metaphors to idioms*. Oxford: Oxford: University Press.

Groff-Palermo, S., and Lawson, J. 2013. Metaphorgy: Metaphor generator. Retrieved 21 Dec 2014 from http://www.metaphor.gy/.

Han, L.; Kashyap, A. L.; Finin, T.; Mayfield, J.; and Weese, J. 2013. UMBC_EBIQUITY-CORE: Semantic textual similarity systems. In *Proc. 2nd Joint Conf. on Lexical and Computational Semantics, Association for Computational Linguistics*.

Hart, M. 2014. Free ebooks - Project Gutenberg. Gutenberg.org. http://www.gutenberg.org/.

Harwood, D. L., and Verbrugge, R. R. 1977. Metaphor and the asymmetry of similarity. Paper presented at the annual meeting of the American Psychological Association, San Francisco.

Hervás, R.; Costa, R. P.; Costa, H.; Gervás, P.; and Pereira, F. C. 2007. Enrichment of automatically generated texts using metaphor. *MICAI 2007, LNAI 4827* 944–954.

Lakoff, G., and Johnson, M., eds. 1980. *Metaphors We Live By*. Chicago, IL: University Of Chicago Press.

Neuman, Y.; Assaf, D.; Cohen, Y.; Last, M.; Argamon, S.; Howard, N.; and Frieder, O. 2013. Metaphor identification in large texts corpora. *PLoS ONE* 8(4): e62343:doi:10.1371/journal.pone.0062343.

Ortony, A., ed. 1993. *Metaphor and Thought*. Cambridge University Press.

Pérez y Pérez, R., and Sharples, M. 2004. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems* 17(1):15–29.

Richards, I. A., ed. 1980. *The Philosophy of Rhetoric*. Oxford: Oxford University Press.

Rothenberg, A., and Hausman, C. R., eds. 1976. *The Creative Question*. Durham NC: Duke University Press.

Sawyer, R. K. 2012. *Explaining Creativity: The Science of Human Innovation*. Oxford University Press.

Socher, R.; Bauer, J.; Manning, C. D.; and Ng., A. Y. 2013. Parsing with compositional vector grammars. In *Proceedings of ACL 2013*.

Steen, G. J.; Dorst, A. G.; Herrmann, J. B.; Kaal, A.; Krennmayr, T.; and Pasma, T. 2010. *A method for linguistic metaphor identification*. From MIP to MIPVU. Amsterdam: John Benjamins.

Tourangeau, R. 1981. Aptness in metaphor. *Cognitive Psychology* 13(1):27–55.

Tunnell, S. 1977. *The Quotable Women, 1800-1975*. Corwin Books.

Turner, S. 1992. Minstrel: a computer model of creativity and storytelling. Technical Report CSD-920057, Ph.D. Thesis, Computer Science Department, University of California, Los Angeles, CA.

Tversky, A. 1977. Features of similarity. *Psychological Review* 84:327–352.

Veale, T., and Hao, Y. 2007a. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. In *Proceedings of the Twenty-*

*Second AAAI Conference on Artificial Intelligence (AAAI-07)*, 1471–1476. Vancouver, British Columbia: AAAI Press.

Veale, T., and Hao, Y. 2007b. Learning to understand figurative language: From similes to metaphors to irony. In *Proceedings of Cog Sci*, 683–688.

# Game of Tropes:

## Exploring the *Placebo Effect* in Computational Creativity

**Tony Veale**
School of Computer Science and Informatics
University College Dublin, Belfield D4, Ireland.
Tony.Veale@UCD.ie

**Abstract**

Twitter has proven itself a rich and varied source of language data for linguistic analysis. For Twitter is more than a popular new channel for social interaction in language; in many ways it constitutes a whole new genre of text, as users adapt to its new limitations (140 character messages) *and* to its novel conventions such as retweeting and hash-tagging. But Twitter presents an opportunity of another kind to computationally-minded researchers of language, a *generative* opportunity to study how algorithmic systems might exploit linguistic tropes to compose novel, concise and re-tweetable texts of their own. This paper evaluates one such system, a *Twitterbot* named *@MetaphorMagnet* that packages its own metaphors and ironic observations as pithy tweets. Moreover, we use *@MetaphorMagnet*, and the idea of Twitterbots more generally, to explore the relationship of linguistic containers to their contents, to understand the extent to which human readers fill these containers with their own meanings, to see meaning in the outputs of generative systems where none was ever intended. We evaluate this *placebo* effect by asking human raters to judge the comprehensibility, novelty and aptness of texts tweeted by simple *and* sophisticated Twitterbots.

## Tropes: *Containers of Meaning*

A mismatch between a container and its contents can often tell us much more than the content itself, as when a person places the ashes of a deceased relative in a coffee can, or sends a brutal death threat in a *Hallmark* greeting card. The communicative effectiveness of mismatched containers is just one more reason to be skeptical of the *Conduit* metaphor (Reddy, 1979) – which views linguistic constructs as containers of propositional content to be faithfully shuttled between speaker and hearer – as a realistic model of human communication. Language involves more than the faithful transmission of logical propositions between information-hungry agents, and more effective communication – of attitude, expectation and creative intent – can often be achieved by abusing our linguistic containers of meaning than by treating them with the sincerity that the *Conduit* metaphor assumes. Consider the case of verbal irony, in which a speaker deliberately chooses containers that are pragmatically ill-suited to the conveyance of their contents. For instance, the advertising container "*If you only see one [X] this year, make it this one*" assumes that [X] denotes a category of event – such as "*romantic comedy*" or "*movie about superheroes*" – with a surfeit of available members for a listener to choose from. When [X] is bound to the phrase "*comedy about Anne Frank*" or "*musical about Nazis*", the container proves too hollow for its content, and the reader is signaled to the presence of playful irony. Though such a film may well be one-of-a-kind, the ill-fitting container suggests there are good reasons for this singularity that do not speak to X's quality as an artistic event. Yet if carefully chosen, an apparently inappropriate container can communicate a great deal about a speaker's relationship to the content conveyed within, and as much again about the speaker's relationship to their audience.

As more practical limitations are placed on the form of linguistic containers, the more incentive one has to exploit or abuse containers for creative ends. Consider the use of Twitter as a communicative medium: writers are limited to micro-texts of no more than 140 characters to convey both their meaning and their attitude to this meaning. So each micro-text, or *tweet*, becomes more than a container of propositional content: each is a brick in a larger edifice that comprises the writer's online personae and textual aesthetic. Many Twitter users employ irony and metaphor to build this aesthetic and thus build up a loyal audience of followers for their world view. Yet Twitter challenges many of our assumptions about irony and metaphor. Such devices must be carefully modulated if an audience is to perceive a speaker's meaning in the playful (mis)match of a linguistic container to its contents. Failure to do so can have serious repercussions when one is communicating to thousands of followers at once, with tweets that demand concision and leave little room for nuance. It is thus not unusual for even creative tweets to come packaged with an explicit tag such as *#irony*, *#sarcasm* or *#metaphor*.

Metaphor and irony are much-analysed phenomena in social media, but this paper takes a generative approach, to consider the *production* rather than the *analysis* of creative linguistic phenomena in the context of a fully-

autonomous computational agent – a *Twitterbot* – that crafts its own metaphorical and ironical tweets from its own knowledge-base of common-sense facts and beliefs. How might such a system exhibit a sense of irony that human users will find worthy of attention, and how might this system craft interesting metaphoric insights from a knowledge-base of everyday facts that are as banal as they are uncontentious? We shall explore the variety of linguistic containers at the disposal of this agent – a real computational system named *@MetaphorMagnet* – to better understand how such containers can be playfully exploited to convey ironic, witty or thought-provoking views on the world. With *@MetaphorMagnet* we aim to show that interesting messages are not crafted from interesting contents, or at least not necessarily so. Rather, effective tweets emerge from an appropriate if non-obvious combination of familiar linguistic containers with unsurprising factual fillers. In support of this view, we shall present an empirical analysis of the assessment of *@MetaphorMagnet*'s uncurated outputs by human judges.

Just as one can often guess the contents of a physical container by its shape, one can often guess the meaning of a linguistic container by its form. We become habituated to familiar containers, and just as we might imagine our own uses for a physical container, we often pour our own meanings into suggestive textual forms. For in language, meaning follows form, and readers will generously infer the presence of meaning in texts that are well-formed and seemingly the product of an intelligent entity, even if this entity is *not* intelligent and any meaning is *not* intentional. Remarkably, Twitter shows that we willingly extend this generosity of interpretation to the outputs of bots that we *know* to be unthinking users of wholly aleatoric methods. Twitterbots exploit this *placebo effect* – wherein a well-formed linguistic container is presumed to convey a well-founded semantic content – by serving up linguistic forms that readers tacitly fill with their own meanings. We aim to empirically demonstrate here that readers do more than willingly suspend their disbelief, and that a well-packaged linguistic form can seduce readers into seeing what is not there: a comprehensible meaning, or at least an intent to be meaningful. We do this by evaluating two metaphor-generating bots side-by-side: a rational, knowledge-based Twitterbot named *@MetaphorMagnet* vs. an aleatoric and largely knowledge-free bot named *@MetaphorMinute*.

## Digital Surrealists: *La Règle Du Jeu*

Most Twitterbots are simple, rule-based systems that use stochastic methods to explore a loosely-defined space of texual forms. Such bots are high-concept, low-complexity text-production mechanisms that transplant the aleatoric techniques of surrealist writers – from André Breton to William Burroughs and Brion Gysin – into the realms of digital content, social networking and online publishing. Each embodies a language game with its own generative rules, or what Breton called "*la règle du jeu.*" Yet Breton, Burroughs and Gysin viewed the use of aleatorical rules as merely the first stage of a two-stage creation process: at this first stage, random recombinant methods are used to confect candidate texts in ways that, though unguided by meaning, are also free of the baleful influence of cliché; at the second stage, these candidates are carefully filtered by a human, to select those that are novel and interesting. Most bots implement the first stage and ignore the second, pushing the task of critiquing and filtering candidate texts onto the humans who read and selectively re-tweet them.

Nonetheless, some bots achieve surprising effects with the simplest language tools. Consider *@Pentametron*, a bot that generates accidental poetry by re-tweeting pairs of random tweets of ten syllables apiece (for an iambic pentameter reading) if each ends on a rhyming syllable. When the meaning of each tweet in a couplet coheres with the other, as in "*Pathetic people are everywhere" |"Your web-site sucks, @RyanAir*", the sum of tweets produces an emergent meaning that is richer and more resonant than that of either tweet alone. Trending social events such as the *Oscars* or the *Super Bowl* are especially conducive to just this kind of synchronicity, as in this fortuitous pairing: "*So far the @SuperBowl commercials blow.*" | "*Not even gonna watch the halftime show.*"

In contrast, a bot named *@MetaphorMinute* wears its aleatoric methods on its sleeve, for its tweets – such as "*a haiku is a tonsil: peachblow yet snail-paced*" – are not so much random metaphors as random metaphor-shaped texts. Using a strategy that stresses quantity over quality, this bot instantiates that standard linguistic container for metaphors – the copula frame *"X is a Y*" – with mostly random word choices every two minutes. Interestingly, its tweets are as likely to provoke a sense of mystification and ersatz profundity as they are total incomprehension. Yet bots such as *@Pentametron* and *@MetaphorMinute* do not generate their texts from the semantic-level up; rather, they manipulate texts at the word-level, and thus lack any sense of the meaning of a tweet, or any rationale for why one tweet might be better – which is to say, more interesting, more apt or more re-tweetable – than others.

The Full-FACE poetry generator of Colton *et al*. (2012) also uses a template-guided version of the cut-up method to mash together semantically-coherent text fragments in a way that – much like *@Pentametron* – obeys certain over-arching constraints on metre and rhyme. These text fragments come from a variety of online sources, ranging from short tweets to long news articles. News stories are a rich source of readymade phrases that convey resonant images, and these can be clipped from a news text using standard NLP techniques, while tweets that use affect-rich language can also be extracted automatically via standard *sentiment analysis* lexica and tools. Thus, a large stock of resonant similes, such as "*blue as a blueberry*" or "*hot as a sauna*" can be extracted from the Web using a search engine (Veale, 2014), since the simile frame "*as X as Y*" is specific enough to query for, and promiscuous enough to match, a rich diversity of typical X:Y associations. These associations can then be recast in a variety of poetic forms to make their clichéd offerings seem fresh again, as in "*Blueberry-blue overalls*" or "*sauna-hot jungle*."

Indeed, the very act of juxtaposing clichés can itself be a creative act, as evidenced both by the success of the cut-up method in general and that of specific cut-ups in particular. Consider William Empson's withering analysis of the persnickety, cliché-hating George Orwell, whom Empson called "*the eagle eye with the flat feet*" (quoted in Ricks [1995:356], who admires Empson's "*audacious compacting of clichés"*). The Full-FACE system is just one of many CC systems that use an autonomous variant of Burroughs and Gysin's cut-up method to integrate tight constraints on form with loose constraints on meaning.

Breton famously stated that "*Je ne veux pas changer la règle du jeu, je veux changer de jeu*." Twitterbots do not change or transcend their own rules, but different bots do represent different language games with their own rules. So to change the game, a CC developer can simply build a new bot, to exploit a different set of tropes and linguistic containers. It is rare for any one Twitterbot to incorporate a diverse set of tropes and production mechanisms; each typically follows Breton's experimentalist approach to art in its random sampling of a specific space of possibilities. Each bot thus forms its own art installation, to showcase a single generative idea. *@MetaphorMagnet*, the bot at the heart of this paper, represents a departure from this norm, insofar as it exploits a wide range of tropes and rendering strategies, it employs diverse sources of knowledge, and it applies a variety of reasoning styles to generate surprising conclusions from what is otherwise a stock of banal facts. But does this added sophistication – bought at the cost of increased system complexity and knowledge-engineering effort – result in tweets that are seen as more meaningful, novel, apt or retweetable by human users? It is this point that exercises us most in the coming sections.

## The Placebo Effect : *Trope-A-Dope*

We humans obtain more mileage than we care to admit from templates, tropes and other "bot" tricks for linguistic creativity. Consider what Matthew McGlone and Jessica Tofighbakhsh (1999) call the *Keats heuristic*, an insight into creative language use that owes as much to Nietzsche ("*we sometimes consider an idea truer simply because it has a metrical form and presents itself with a divine skip and jump*") as to the poet John Keats ("*Beauty is truth, truth beauty*"). McGlone and Tofighbakhsh (2000) show that when presented with uncommon maxims or proverbs with internal rhyme (e.g. "*woes unite foes*"), subjects tend to view these as more insightful about the world than the equivalent paraphrases with no internal rhyme at all (e.g. "*troubles unite enemies*"). While the Keats heuristic is not exactly a license to pun, it is an incentive to rhyme, and to give as much weight (or more still) to superficial aspects of poetry generation as to deep semantics and pragmatics. Indeed, the heuristic is tacitly central to the operation of virtually every computational creativity (CC) approach to poetry generation (e.g. Milic, 1970; Chamberlain & Etter, 1983; Gervás, 2000; Manurung *et al*. 2012; Veale, 2013). If human poets ask questions first and rhyme later, CC systems typically rhyme first and ask questions later, if at all. For if the human jury in the O.J. Simpson trial could be turned against bald facts with the Keatsian "*If the glove don't fit you must acquit*", readers of computer-generated poetry can be persuaded to see deliberate meaning and resonance in any output that has a "*divine skip and jump*."

There is something undeniably special about poetry, whether it is the gentle poetry of William Shakespeare's "*Shall I compare thee to a summer's day*" or the rough poetry of Johnnie Cochrane's "*If the glove don't fit you must acquit*". Milic (1970), an early CC pioneer, argues that while poetry "*is more difficult to write than prose*" it offers other freedoms to writers due to the willingness of readers to "*interpret a poem, no matter how obscure, until he has achieved a satisfactory understanding*." What then of the enigmatic tweets of bots like *@MetaphorMinute*, whose obscurity is a function of random word choice and whose surface forms are not designed to make any sense at all? Milic argues that computer poetry serves a useful role other than its obviously generative one, by alerting us to "*the curious behavior of familiar words in unfamiliar combinations*." Behaviour that makes perfect sense when dealing with the writings of a gifted human poet, such as our tendency to "*interpret an utterance by making what concessions are necessary on the assumption that a writer has something in mind of which the utterance is the sign*", is, argues Milic, "*inappropriate when the speaker is a computer*." Yet Twitterbots benefit from such concessions and assumptions whether or not followers know them to be bots. This *placebo effect* is especially pronounced in the coining of would-be metaphors, leading Milic to note "*how readily we accept metaphor as an alternative to calling a sentence nonsensical*." *@MetaphorMinute* and other aleatoric bots wring maximal value from this insight by devising texts that they themselves cannot distinguish from nonsense. So this begs an important question: are the meanings imposed on a random text by a creative human of comparable value to those conveyed by a bot with its own model of the world and its own insights to tweet?

## Building Metaphors : *Theory and Practice*

What might it mean for a bot to have "*something in mind of which [its] utterance is the sign*"? When it comes to metaphor generation, we might expect that our bot would generate its figurative tweets from a conceptual model of the world as it sees it, in a way that accords with a sound theory of *how* and *why* humans actually use metaphor. For the latter, AI offers us a range of models to choose from.

Computational approaches to metaphor divide into four broad classes: the categorial, the corrective. the analogical and the schematic. Categorial approaches view metaphor as a means to reconceptualize one idea by placing it into a taxonomic category strongly associated with another (see Hutton, 1982; Way, 1991; Glucksberg, 1998). Corrective approaches view metaphor as an inherently anomalous deviation from literal language, and strive to *recover* the corresponding literal meaning of any figurative statement that violates its lexico-semantic norms (see Wilks, 1978;

Fass, 1991). The analogical approaches aim to capture the relational parallels that allow our representation of an idea in one domain, the *source*, to be systematically projected onto our mental representation of an idea in another, the *target* (see Gentner *et al*., 1989; Veale and Keane, 1997). Finally, schematic approaches aim to explain how related linguistic metaphors arise as surface manifestions of deep seated cognitive structures called *Conceptual Metaphors* (Lakoff & Johnson, 1980; Carbonell, 1981; Martin, 1990; Veale & Keane, 1992). Each approach has its own merits, but none offers a complete computational solution. Bots that aim for a general competence in metaphor must thus implement a selective hybrid of multiple approaches. Yet each approach also requires its own source of knowledge. Categorial approaches require a comprehensive taxonomy of flexible categories that can embrace atypical members on demand. Corrective approaches are built on a substrate of literal case-frames onto which deviant usages can be correctively projected. Analogical approaches assume an inventory of graph-theoretic representations of concepts, from which a structure-mapping engine can eke out its sub-graph isomorphisms. Schematic approaches rely on a stock of Conceptual Metaphors (CMs) – such as *Life is a Journey* or *Theories are Buildings* – to unearth the deep structures beneath the surface of diverse linguistic forms.

Though hybrid approaches demand multiple sources of knowledge, there exist public Web services that integrate this knowledge with the appropriate means of using it for metaphor. The *Thesaurus Rex* Web service of Veale & Li (2013) provides a highly divergent system of fine-grained categorizations that allows a 3rd-party client system to e.g. determine that *War* and *Divorce* have each been viewed as kinds of *destructive thing*, *traumatic event* and *severe conflict* in the texts of the Web. The *Metaphor Eyes* Web service of Veale & Li (2011) is a rich source of relational norms – also harvested at scale from Web texts – such as that businesses earn profits and pay taxes, or that religions ban alcohol and believe in reincarnation. The *Metaphor Magnet* service of Veale (2014) offers a rich source of the stereotypical properties and behaviors of familiar ideas, and provides the means to retrieve salient CMs from the Google n-grams (Brants & Franz, 2006) which can then be further elaborated to create novel linguistic metaphors.

*@MetaphorMagnet* relies on each of these public Web services to generate the conceptual conceits that underpin its figurative tweets. For instance, it uses *Thesaurus Rex* to provide the categorization insights that it then packages as *odd-one-out* lists or as *faux*-dictionary definitions. It uses the *Metaphor Eyes* service to provide the relational structures it needs to perform structure mapping and thus concoct original analogies and dis-analogies. And it uses the *Metaphor Magnet* service to access the stereotypical properties and behaviors of ideas, and to juxtapose these properties via resonant contrasts and norm contraventions. Once the conceptual chassis of a metaphor is constructed in this way, it is then packaged in an apt linguistic form.

## Building Strings: *Trope-On-A-Rope*

CMs such as *Life Is A Journey* and *Politics Is A Game* are more than productive deep-structures for the generation of whole families of linguistic metaphors; they also provide the conceptual mappings that shape our habitual thinking about such familiar ideas as *Life*, *Love, Politics* and *War*. Politicians and philosophers exploit conceptual metaphors to frame an issue and shape our expectations; when a CM fails to match our own experience, we reject it and switch to a more apt metaphor. So a metaphor-generating bot can thus create a thought-provoking opposition by pitting one CM against another that advocates a conflicting view of the world. The following tweet from *@MetaphorMagnet* uses this approach to contrast two views on *#Democracy*:

*To some voters, democracy is an important cornerstone.*
*To others, it is a worthless failure.*
#Democracy=#Cornerstone #Democracy=#Failure

The CM *Democracy Is A Cornerstone* (of society) is often used to frame political discussions, and can be seen as an specialization of the CM *Society Is A Building*, itself an elaboration of the CM *Organization Is Physical Structure* (see Grady, 1997). Yet the importance of cornerstones to the buildings they anchor finds a sharp contrast in the assertion that *Democracy Is A Failure*. Each of these affective claims is so commonly asserted that they can be found in the Google n-grams, a large database of short fragments of frequent Web texts. The 4-gram "*democracy is a cornerstone*" has a frequency of 91 in the Google n-grams, while the 4-gram "*democracy is a failure*" has a frequency of 165. These n-grams, which suggest potential CMs for *@MetaphorMagnet*, are elaborated with added detail via the *Metaphor Magnet* Web service, which tells the bot that the stereotypical *cornerstone* is *important* and the stereotypical *failure* is *worthless*. The following tweet makes similar use of CMs found in the Google n-grams, but renders the conflict in a different linguistic container:

*Remember when tolerance was promoted by crusading liberals? Now, tolerance is violence that only fearful appeasers can avoid.*

The bot is guided here by the suggestive Google 3-gram "*Tolerance for Violence*" (frequency=1353), but it does not directly contrast the ideas #Tolerance and #Violence. Instead, it finds a potential analogy in this juxtaposition, between the promoters of #Tolerance (which it renders as *crusading liberals*) and the opponents of #Violence (which it renders as *fearful appeasers*). The choice of stereotypical properties (*crusading* and *fearful*) is driven by the bot's need to create a resonant semantic opposition. The bot omits the hashtags #Tolerance=#Violence from this tweet due to the confines of Twitter's 140-character limit. But it can also choose to render a complex conceit across two successive tweets, as in the following pair:

*Remember when research was conducted by prestigious philosophers?* #Research=#Fruit #Philosopher=#Insect

*Now, research is a fruit eaten only by lowly insects.* #Research=#Fruit #Philosopher=#Insect

*@MetaphorMagnet* uses a number of packaging strategies to turn a figurative comparison into an ironic observation, ranging from the use of an explicit *#irony* hashtag (which is commonplace on Twitter) to the use of "scare" quotes to focus on the part of a tweet most deserving of disbelief. The following tweet showcases both of these strategies:

#Irony: *When some chefs prepare* "fresh" *salads the way apothecaries prepare noxious poisons.* #Chef=#Apothecary #Salad=#Poison

Irony offers a concise means of contrasting two points of view: that which is expected and the disappointing reality. By comparing the preparation of salads – the "*healthy*" option on most menus – to the preparion of poisons, this analogy undermines the expectation of healthfulness and suggests that some salads are noxious and chemical-filled. The real world is filled with situations in which naturally antagonistic properties are found in surprising proximity. These situations, if expressed in the right linguistic form, can be elevated to the level of situational irony. Consider, for instance, the following *@MetaphorMagnet* tweet:

#Irony: *When the timers that are found in enjoyable games activate gruesome bombs.* #Enjoyable=#Gruesome

It is important to stress that *@MetaphorMagnet* does not simply fill linguistic templates with related words. Rather, the above tweet is constructed at the knowledge-level, by a bot that intentionally seeks out stereotypical norms that are related (e.g. by a pivotal idea *timer*) yet which can be placed into antagonistic juxtapositions around this pivot. In effect, the goal of the linguistic rendering is to package a knowledge-level conceit – typically a conflict of ideas and properties – in a tweet-sized narrative. For example, the following tweet is rendered as a narrative of change:

*To join and travel in a pack: This can turn pretty girls into ugly coyotes.* #Girl=#Coyote

Twitter offers unique social affordances that allow a bot to elevate almost any contrast of ideas into a dramatic narrative. Rather than talk of generic liberals or appeasers, a bot can give these straw men real names, or at least invent fake names that look like the real thing and which, as Twitter handles, seem wittily apropos to the views that are espoused. In this way, by imagining its central conceit as a topic of a vigorous debate by real people, a bot can turn an abstract metaphor into a concrete situation with its own colorful participants. Consider the social debate that is made personal in this tweet from *@MetaphorMagnet*:

.@war_poet *says history is a straight line*
.@war_prisoner *says it is a coiled chain*
#History=#Line #History=#Chain

The handles @war_poet and @war_prisoner are invented by *@MetaphorMagnet* to suit, and amplify, the figurative views that they are advanced in the tweet, by using a mix of relational knowledge (from the *Metaphor Eyes* service) and language data (via the Google n-grams). Since poets write poems about the wars that punctuate history, and poems contain lines, the 2-gram "*war poet*" is recognized as an apt handle for an imaginary Twitter user who might advance a view of *history as a line*. In this case the handle @war_poet really does name a real Twitter user, but this only adds to the sense that Twitterbot confections are a new kind of interactive theatre and performance art (see Dewey, 2014). Note that the more profound aspects of this contrast are not appreciated by *@MetaphorMagnet* itself, or at least not yet. For example, the bot does not yet appreciate what it means for history to be a straight line, and while it knows enough to invent the intriguing handle @war_prisoner, neither does it appreciate what it might mean to be a prisoner of history, enslaved in a repeating cycle of war. The placebo effect is not a binary effect: it benefits by degrees, and can benefit knowledge-rich bots just as much as knowledge-free bots. Our bots will always evoke in we humans more than they themselves can ever appreciate, yet this may itself be a key part of CC's allure.

## Bot Vs. Bot : *The Metaphor Challenge*

*@MetaphorMagnet* differs from *@MetaphorMinute* in a number of key ways. For one, its mechanics are informed by Lakoff and Johnson's *Conceptual Metaphor Theory* and a range of computational approaches. For another, it draws on considerable semantic and linguistic resources, from a large knowledge-base of conceptual relations and stereotypical beliefs to the linguistic diversity of the Google n-grams. All of *@MetaphorMagnet*'s tweets – all its hits and all its misses – are open to public scrutiny on Twitter. But to empirically evaluate the success of the bot as a knowledge-based, theory-driven producer of novel, meaningful and retweet-worthy metaphors, we turn to the crowdsourcing platform *CrowdFlower*, where we conduct a comparative evaluation of *@MetaphorMagnet* and its closest knowledge-*free* counterpart, *@MetaphorMinute*. The latter, designed by noted bot-maker Darius Kazemi, uses a wholly aleatoric approach to metaphor generation yet has over 500 followers on Twitter that do not mind its *one-every-two-minutes* scattergun approach to generation. *@MetaphorMinute* crafts metaphors by filling a template with nouns and adjectives that are chosen more-or-less at random, to produce inscrutable tweets such as "*a cubit is a headboard: stational yet tongue-obsessed*."

We chose 60 tweets at random from the past outputs of each Twitterbot. CrowdFlower annotators, who were each paid a small sum per judgment, were not informed of the origin of any tweet, but simply told that each was selected from Twitter because of its metaphorical content. We did not want annotators to actively suspend their disbelief by knowingly dealing with bot outputs. Annotators were paid to rate the content of each tweet along three dimensions,

*Comprehensibility*, *Novelty* and likely *Retweetability*, and to rate all three dimensions on the same scale: *Very Low* to *Medium Low* to *Medium High* to *Very High*. Ten annotations were solicited for each dimension of each tweet, though the responses of likely scammers (non-engaged annotators) were later removed from the dataset. Tables 1 through 3 present the distributions of mean ratings per tweet, for each dimension and each Twitterbot.

| ***Comprehensibility*** | ***@Metaphor Magnet*** | ***@Metaphor Minute*** |
|---|---|---|
| ***Very Low*** | 11.6% | **23.9%** |
| ***Med. Low*** | 13.2% | **22.2%** |
| ***Med High*** | **23.7%** | 22.4% |
| ***Very High*** | **51.5%** | 31.6% |

Table 1. *Relative Comprehensibility of each bot*

So more than half of *@MetaphorMagnet*'s tweets were ranked as having very high comprehensibility, while less than one third of *@MetaphorMinute*'s tweets are so ranked. More surprising, perhaps. is the result that annotators found more than half of *@MetaphorMinute*'s wholly random metaphors to have medium-high to very-high comprehensibility. This Twitterbot's use of abstruse terminology, such as *stational* and *peachblow*, may be a factor here, as might the bot's use of the familiar copula container *X is Y* for its metaphors, which may well seduce annotators into believing that an apparent metaphor really does have a comprehensible meaning, if only one were to expend enough mental energy to actually discern it.

| ***Tweet Novelty*** | ***@Metaphor Magnet*** | ***@Metaphor Minute*** |
|---|---|---|
| ***Very Low*** | **11.9%** | 9.5% |
| ***Med. Low*** | **17.3%** | 12.4% |
| ***Med High*** | **21%** | 14.9% |
| ***Very High*** | 49.8% | **63.2%** |

Table 2. *Comparative Novelty of each bot's tweets*

The dimension *Novelty* yields results that are equally surprising. While half of *@MetaphorMagnet*'s metaphors are rated as having very-high novelty in Table 2, almost two-thirds of *@MetaphorMinute*'s tweets are just as highly rated. However, we should not be overly surprised that *@MetaphorMinute*'s bizarre juxtapositions of rare or unusual words, as yielded by its unconstrained use of aleatoric techniques, are seen as more unusual than those word juxtapositions arising from *@MetaphorMagnet*'s controlled use of attested Web n-grams and stereotypical knowledge. As shown by Giora *et al*. (2004), novelty is neither a source of pleasure in itself nor is it a reliable benchmark of creativity. Rather, pleasurability derives from the recognition of *useful* novelty, that is, novelty that can be understood and appreciated relative to the familiar.

| ***Re-Tweetability*** | ***@Metaphor Magnet*** | ***@Metaphor Minute*** |
|---|---|---|
| ***Very Low*** | 15.5% | **41%** |
| ***Med. Low*** | **41.9%** | 34.1% |
| ***Med High*** | **27.4%** | 15% |
| ***Very High*** | **15.3%** | 9.9% |

Table 3. *Relative Retweetability of each bot's tweets*

On Twitter, useful exploitation is frequently a matter of social reach. A tweet is novel and useful to the extent that it attracts the attention of Twitter users and is deemed worthy of re-tweeting to others in one's social circle. Our third dimension, *Re-Tweetability*, reflects the likelihood that an annotator would ever consider re-tweeting a given metaphorical tweet to others. Though we ask annotators to speculate here – neither bot has enough followers to perform a robust statistical analysis of actual retweet rates – the results largely conform to our expectations. The results of Table 3 show retweetability to be a matter of novelty *and* comprehensibility, and not just novelty alone. Though annotators are not generous with their *Very-High* ratings for either bot, *@MetaphorMagnet*'s tweets are judged to be considerably more re-tweetable than the largely random offerings of *@MetaphorMinute*.

Comprehensibility and comprehension are two different things: while a Computational Creativity (CC) version of the placebo effect may well foster a belief that a given tweet has a coherent meaning, it cannot actually provide this meaning. Meaning is the product of interpretation, and interpretation is often hard. Milic (1970) notes that in a context that licences a poetic interpretation, such as one in which a reader is told that a particular text is a metaphor, readers are more likely to accept that the text – as inscrutable as it may be – has a metaphorical meaning rather than dismiss it as nonsense. Recall that over 75% of *@MetaphorMagnet*'s tweets and over 50% of *@MetaphorMinute*'s tweets are judged as having *medium-high* to *very-high* comprehensibility. We thus need to look deeper, to determine whether raters can actually back up these judgments with actual meanings.

In a second CrowdFlower experiment, we make raters work harder, to reconstruct a partial tweet by adding the missing information that will make it whole and apt again. That is, we employ a *cloze* test format for this experiment, by removing from each tweet the pair of key qualities that anchor the tweet and make its comparison of ideas seem meaningful and apt. For *@MetaphorMagnet*, for example, we remove the properties *detailed* and *vague* in this tweet:

*To some freedom fighters, freedom is a* detailed *recipe. To others, it is a* vague *dream*. #Freedom=#Recipe #Freedom=#Dream

For *@MetaphorMinute*, we excise the pair of qualities *hippy* and *revisional* from the following tweet:

*a flatfoot is a houseboat:* hippy *and* revisional

For each tweet from each bot, we blank out a pair of original qualities as above; this pairing is the answer that is sought from human judges. We also choose 4 distractor pairs for each original pair, by selecting pairs from other tweets from the same bot. As in our first experiment, we chose 60 tweets at random from the past outputs of each bot, and 10 ratings were solicited for each. Annotators were presented with a tweet in which the key properties were blanked out (as above), and given five randomly ordered pairs of possible fillers to choose from. To make the results of the experiment comparable to those of the 1st experiment (Tables 1,2,3), we obtain the mean aptness of each tweet, so that e.g. if 7 out of 10 raters correctly choose the original pairing, then that tweet is deemed to have an aptness of 0.7. We then place these aptness scores into bands, where the *Very Low* band = 0 to 0.25, *Medium Low* = 0.26 to 0.5, *Medium High* = 0.51 to 0.75, and *Very High* = 0.76 to 1. By calculating the distribution of tweets to each band, we can determine e.g. the percentage of tweets from each bot that are put into the *Very High* band.

Our hypothesis is rather straightforward: if tweets are linguistic containers that are carefully crafted to convey a particular meaning, then it should be easier to select the missing pair of qualities that make this meaning whole; if, on the other hand, the tweet is all there is, and its content is chosen mostly at random, then raters will choose the right pairing with no more success than random selection.

The results reported in Table 4 bear out our hypothesis.

| ***Metaphor Aptness*** | ***@Metaphor Magnet*** | ***@Metaphor Minute*** |
|---|---|---|
| ***Very Low*** | 0% | **84%** |
| ***Med. Low*** | **22%** | 16% |
| ***Med High*** | **58%** | 0% |
| ***Very High*** | **20%** | 0% |

Table 4. *Relative Aptness of each bot's metaphors*

The placebo effect in CC can lead us to appreciate a bot's tweets as meaningful but cannot tell us what this meaning should be. Though the results above may seem a foregone conclusion, as *@MetaphorMagnet*'s tweets are designed to communicate a fully recoverable meaning while those of *@MetaphorMinute* are not, this is surely what it means to engage in real communication: to design an utterance so that an intended meaning is re-created, in whole or in part, in the mind of an intelligent, receptive audience.

## Fake It 'Til You Make It

The *Placebo Effect* benefits *all* Computational Creativity systems, from superficial users of surealistic techniques to sophisticated knowledge-based AI systems. That this is so should come as no surprise, for we humans also benefit from the effects of an active and receptive mind when dealing with other people. Just as a prior belief in the efficacy of a medical intervention can lead us to perceive (and experience) a post-hoc benefit from an otherwise empty treatment, a prior belief in the meaningfulness of a verbal intervention can lead us to perceive (and enjoy) a creative meaning where none was ever intended. When a CC system uses superficial techniques to convey a sense of understanding and profundity with otherwise shallow linguistic forms, as in Weizenbaum's (1965) infamous ELIZA system, the label "*ELIZA Effect"* proves to be an apt one (Hofstadter, 1995). However, we humans are also subject to an *ELIZA effect* of our own, insofar as we often do others the courtesy of assuming their utterances to be freighted with real meaning and creative intent, and will often work hard to uncover that meaning for them.

At one time or another, we have all relied on catch-phrases, clichés, slogans, idioms, canned jokes and other half-empty linguistic containers to suggest to others that we have deeper meanings in mind, or have something more profound to offer, than we actually do. In a famous polemical essay from 1946, George Orwell excoriates speakers of English for their reliance on jargon, foreign words and empty phraseology as a substitute for thoughts of real substance, while Geoff Pullum (2003) upbraids modern speakers for a grating over-reliance on "*multi-use, customizable, instantly recognizable, time-worn, quoted or misquoted phrases or sentences that can be used in an entirely open array of different jokey variants by lazy journalists and writers.*" These "*phrases for lazy writers in kit form*" are not that different from the template-based language games played by superficial Twitterbots, and though we humans fill our templates – such as "*X is the new black*", "*In X no one can hear you scream*" or "*if the Eskimos have N words for snow then Xs surely have as many for Y*" – with lexical fillers that are contextually apt, we employ our templates to be just as provocative, and to imply or to suggest more than we actually mean.

To see machines work with humans in the construction of real figurative meanings, readers are directed to a variant of *@MetaphorMagnet* – a related bot named *@MetaphorMirror* – that tweets its own novel metaphors in response to breaking news events. This bot's metaphors are not offered as informative summaries of the news, but as figurative lenses through which followers can view the news and adopt a novel perspective on human affairs.

## Acknowledgements

This research was supported by the EC project *WHIM:*

*The What-If Machine*. See *http://www.whim-project.eu/*

## References


Thorsten Brants and Alex Franz. (2006). Web 1T 5-gram database, Version 1. *Linguistic Data Consortium*.

Jaime G. Carbonell. (1981). Metaphor: An inescapable phenomenon in natural language comprehension. *Report 2404*. Carnegie Mellon Computer Science Dept.

William Chamberlain and Thomas Etter. (1983). *The Police-man's Beard is Half-Constructed: Computer Prose and Poetry*. Warner Books.

Simon Colton, Jacob Goodwin and Tony Veale. (2012). Full-FACE Poetry Generation. In *Proc. of the 3rd International Conference on Computational Creativity*, Dublin, Ireland.

Caitlin Dewey. (2014). What happens when @everyword ends? Intersect, *Washington Post*, May 23rd edition.

Dan Fass. (1991). Met*: a method for discriminating metonymy and metaphor by computer. *Computational Linguistics,* 17(1):49-90.

Dedre Gentner, Brian Falkenhainer and Janice Skorstad. (1989). Metaphor: The Good, The Bad and the Ugly. In *Theoretical Issues in NLP*, Yorick Wilks (Ed.) Hillsdale, NJ: Lawrence Erlbaum Associates.

Pablo Gervás. (2000). Wasp: Evaluation of different strategies for automatic generation of Spanish verse. In *Proc. of the AISB-2000 Symposium on Creative & Cultural Aspects of AI*, 93-100.

Rachel Giora, Ofer Fein, Jonathan Ganzi, Natalie Alkeslassy Levi and Hadas Sabah. (2004).Weapons of Mass Distraction: Optimal Innovation and Pleasure Ratings. *Metaphor and Symbol **19**(2):115-141*.

Sam Glucksberg. (1998). Understanding metaphors. *Current Directions in Psychological Science,* 7:39-43.

Joseph Grady. (1997). Foundations of Meaning: Primary Metaphors and Primary Scenes. University of California.

Douglas Hofstadter. (1995). The Ineradicable Eliza Effect and Its Dangers. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought* (Preface 4), Basic Books: New York.

James Hutton (*translator)* (1982). *Aristotle's Poetics*. New York, NY: Norton.

George Lakoff and Mark Johnson. (1980). *Metaphors We Live By*. Chicago, Illinois: Chicago University Press.

James H. Martin. (1990). *A Computational Model of Metaphor Interpretation*. Academic Press.

Ruli Manurung, Graeme Ritchie and Henry Thompson. (2012). Using genetic algorithms to create meaningful poetic text. JETAI 24(1):43–64.

Matthew S. McGlone and Jessica Tofighbakhsh. (1999). The Keats heuristic: Rhyme as reason in aphorism interpretation, *Poetics* **26**(4):235-44.

Matthew S. McGlone and Jessica Tofighbakhsh. (2000). Birds of a feather flock conjointly (?): rhyme as reason in. aphorisms. *Psychological Science* **11** (5): 424–428.

Louis T. Milic. (1971). The possible usefulness of computer poetry. *The Computer in Literary and Linguistic Research*, R.A. Wisbey (Ed.), Cambridge, MA.

Geoffrey Pullum. (2003). Phrases For Lazy Writers in Kit Form. *Language Log post*, October 27, 2003.

George Orwell. (1946). Politics and the English language. *Horizon,* 13(76), April issue.

Michael J. Reddy. (1979). *The conduit metaphor: A case of frame conflict in our language about language.* In A. Ortony (Ed.), *Metaphor and Thought, 284–310.* Cambridge University Press.

Christopher B. Ricks, (1980). Clichés. In: L. Michaels and C. Ricks (Eds), *The State of the Language.* University of California Press, Berkeley.

Tony Veale and Mark T. Keane. (1992). Conceptual Scaffolding: A spatially founded meaning representation for metaphor comprehension. Computational Intelligence 8(3):494-519.

Tony Veale and Mark T. Keane. (1997). The Competence of Sub-Optimal Structure Mapping on 'Hard' Analogies. In *Proceedings of IJCAI'97, the 15th International Joint Conference on Artificial Intelligence.* Nagoya, Japan. Morgan Kaufmann.

Tony Veale and Guofu Li. (2011). Creative Introspection and Knowledge Acquisition. In Proc. of AAAI-2011, *The 25th Conference of the Association for the Advancement of Artificial Intelligence.* San Francisco: AAAI Press.

Tony Veale and Guofu Li. (2013). Creating Similarity: Lateral Thinking for Vertical Similarity Judgments. *In Proceedings of ACL 2013, the 51st Annual Meeting of the Assoc. for Computational Linguistics, Sofia, Bulgaria,*

Tony Veale. (2013). Less Rhyme, More Reason: Knowledge-based Poetry Generation with Feeling, Insight and Wit. *In Proc. of ICCC 2013, the 4th Int. Conference on Computational Creativity. Sydney, Australia.*

Tony Veale. (2014). Running With Scissors: Cut-Ups, Boundary Friction and Creative Reuse. In Proceedings of *ICCBR-2014, the 22nd International Conference on Case-Based Reasoning*.

Eileen Cornell Way. (1991). *Knowledge Representation and Metaphor: Studies in Cognitive systems*. Kluwer.

Joseph Weizenbaum. (1966). ELIZA – A Computer Program For the Study of Natural Language Communication Between Man And Machine. *Communications of the ACM* **9** (1): 36–45.

Yorick Wilks. (1978). Making Preferences More Active. *Artificial Intelligence* 11(3):197-223.

# OMG UR Funny!
# Computer-Aided Humor with an Application to Chat

**Miaomiao Wen**
Carnegie Mellon University

**Nancy Baym, Omer Tamuz, Jaime Teevan, Susan Dumais, Adam Kalai**
Microsoft Research

**Abstract**

In this paper we explore Computer-Aided Humor (CAH), where a computer and a human collaborate to be humorous. CAH systems support people's natural desire to be funny by helping them express their own idiosyncratic sense of humor. Artificial intelligence research has tried for years to create systems that are funny, but found the problem to be extremely hard. We show that by combining the strengths of a computer and a human, CAH can foster humor better than either alone. We present CAHOOTS, an online chat system that suggests humorous images to its users to include in the conversation. We compare CAHOOTS to a regular chat system, and to a system that automatically inserts funny images using an artificial humor-bot. Users report that CAHOOTS made their conversations more enjoyable and funny, and helped them to express their personal senses of humor. Computer-Aided Humor offers an example of how systems can algorithmically augment human intelligence to create rich, creative experiences.

## Introduction

Can a computer be funny? This question has intrigued the pioneers of computer science, including Turing (1950) and Minsky (1984). Thus far the answer seems to be, "No." While some computer *errors* are notoriously funny, the problem of creating Computer-Generated Humor (CGH) systems that *intentionally* make people laugh continues to challenge the limits of artificial intelligence.

State-of-the-art CGH systems are generally textual. CHG systems have tried to do everything from generating word-play puns (Valitutti 2009) (e.g., "What do you get when you cross a fragrance with an actor? A smell Gibson") and identifying contexts in which it would be funny to say, "That's what she said," (Kiddon and Yuriy 2011) to generating I-like-my-*this*-like-my-*that* jokes (Petrovic and David 2013) (e.g., "I like my coffee like I like my war, cold") and combining pairs of headlines into tweets such as, "NFL: Green Bay Packers vs. Bitcoin – live!"[1] However, none of these systems has demonstrated significant success.

Despite the challenge that computers face to automatically generate humor, humor is pervasive when people use computers. People use computers to share jokes, create funny videos, and generate amusing memes. Humor and laughter have many benefits. Online, it fosters interpersonal rapport and attraction (Morkes et al. 1999), and supports solidarity, individualization and popularity (Baym 1995). Spontaneous humor production is strongly related to creativity, as both involve making non-obvious connections between seemingly unrelated things (Kudrowitz 2010).

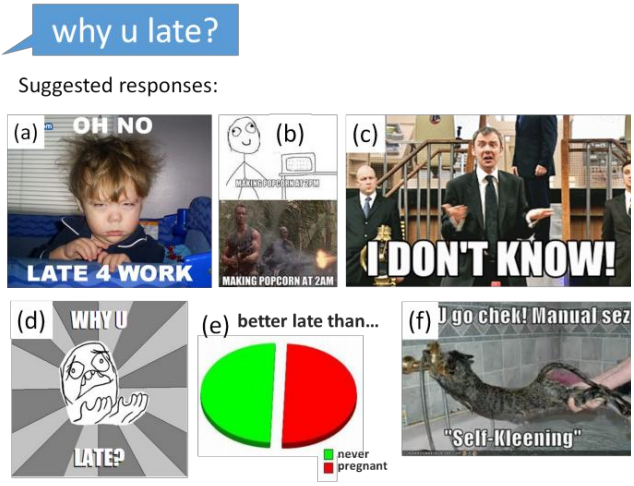



**Figure 1. Images suggested by CAHOOTS in response to chat line, "why u late?" (a), (b), and (e) are from image search query "funny late", (f) is from query "funny why", (c) is a canned reaction to questions, and (d) is a meme generated on-the-fly.**

Computers and humans have different strengths, and therefore their opportunity to contribute to humor differs. Computers, for example, are good at searching large data sets for potentially relevant items, making statistical associations, and combining and modifying text and images. Humans, on the other hand, excel at the complex social and linguistic (or visual) processing on which humor relies. Rather than pursuing humor solely through a CGH strategy, we propose providing computational support for humorous interactions between people using what we call *Computer-Aided Humor* (CAH). We show that by allowing the computer and human to work together, CAH systems can help people be funny and express their own sense of humor.

We explore the properties of this form of interaction and prove its feasibility and value through CAHOOTS (Computer-Aided Hoots), an online chat system that helps people be funny (Figure 1). CAHOOTS supports ordinary text chat, but also offers users suggestions of possibly funny

[1] http://www.twitter.com/TwoHeadlines

images to include based on the previous text and images in the conversation. Users can select choices they find on-topic or humorous and can add funny comments about their choices, or choose not to include any of the suggestions. The system was designed iteratively using paid crowd workers from Amazon Mechanical Turk and interviews with people who regularly use images in messaging.

We compare CAHOOTS to CGH using a chat-bot that automatically inserts funny images, and to ordinary chat with no computer humor. The bot uses the same images that CAHOOTS would have offered as suggestions, but forcibly inserts suggestions into the conversation. Compared to these baselines, CAHOOTS chats were rated more fun, and participants felt more involved, closer to one another, and better able to express their sense of humor. CAHOOTS chats were also rated as more fun than ordinary chat. Our findings provide insights into how computers can facilitate humor.

## Related Work

In human-human interaction, humor serves several social functions. It helps in regulating conversations, building trust between partners and facilitating self-disclosure (Wanzer et al. 1996). Non-offensive humor fosters rapport and attraction between people in computer-mediated communication (Morkes et al. 1999). It has been found that five percent of chats during work are intended to be primarily humorous (Handel and James 2002), and wall posts in Facebook are often used for sharing humorous content (Schwanda et al. 2012). Despite the popularity and benefits of humorous interaction, there is little research on how to support humor during computer-mediated communication. Instead, most related work focuses on computationally generating humor.

### Computational Humor

Computational humor deals with automatic generation and recognition of humor. Prior work has mostly focused on recognizing or generating one specific kind of humor, e.g. one-liners (Strapparava et al. 2011). While humorous images are among the most prominent types of Internet-based humor (Shifman 2007), little work addresses computational visual humor.

Prior work on CGH systems focus on amusing individuals (Dybala 2008; Valitutti et al. 2009). They find humor can make user interfaces friendlier (Binsted 1995; Nijholt et al. 2003). Morkes et al. (1998) study how humor enhances task-oriented dialogues in computer-mediated communication. HumoristBot (Augello et al. 2008) can both generate humorous sentences and recognize humoristic expressions. Sjobergh and Araki (2009) designed a humorous Japanese chat-bot. However, to the best of our knowledge, no prior research has studied collaboratively being funny using humans and computers.

### Creativity Support Tools

CAH is a type of creativity support tool aimed specifically at humor generation within online interaction. Shneiderman (2007) distinguishes creativity support tools from productivity support tools through three criteria: clarity of task domain and requirements, clarity of success measures, and nature of the user base.

Creativity support tools take many forms. Nakakoji (2006) organizes the range of creativity support tools with three metaphors: running shoes, dumbbells, and skis. Running shoes improve the abilities of users to execute a creative task they are already capable of. Dumbbells support users learning about a domain to become capable without the tool itself. Skis provide users with new experiences of creative tasks that were previously impossible. For users who already utilize image-based humor in their chats, CAHOOTS functions as running shoes. For the remaining users, CAHOOTS serves as skis.

## System Design

Our system, CAHOOTS, was developed over the course of many iterations. At the core of the system lie a number of different algorithmic *strategies* for suggesting images. Some of these are based on previous work, some are the product of ideas brainstormed in discussions with comedians and students who utilize images in messaging, and others emerged from observations of actual system use. Our system combines these suggestions using a simple reinforcement learning algorithm for ranking, based on R-Max (Brafman and Tennenholtz 2003), that learns weights on strategies and individual images from the images chosen in earlier conversations. This enabled us to combine a number of strategies.

### User Interface

CAHOOTS is embedded in a web-based chat platform where two users can log in and chat with each other. Users can type a message as they would in a traditional online chat application, or choose one of our suggested humorous images. Suggested images are displayed below the text input box, and clicking on a suggestion inserts it into the conversation. Both text and chosen images are displayed in chat bubbles. See Figure 2 for an example. After one user types text or selects an image, the other user is provided with suggested image responses.

### The Iterative Design Process

We initially focused on text-based humor suggestions based on canned jokes and prior work (Valitutti et al. 2009). These suffered from lack of context, as most human jokes are produced within humorous frames and rarely communicate meanings outside it (Dynel 2009). User feedback was negative, e.g., “The jokes might be funny for a three year old” and “The suggestions are very silly.”

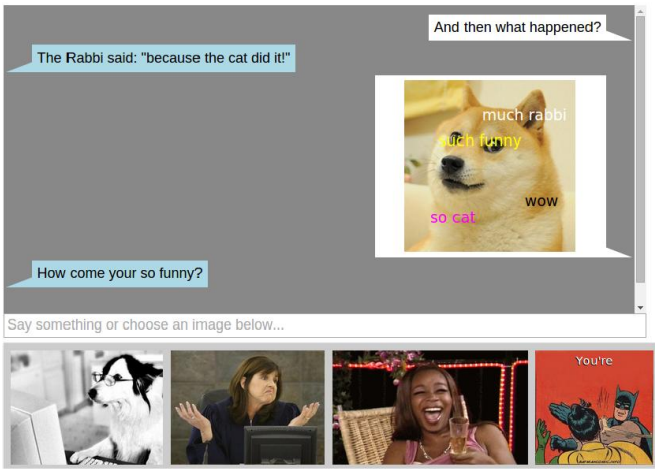



**Figure 2. The CAHOOTS user interface in a chat, with user's messages (right in white) and partner's (left in blue). All text is user-entered while images are suggested by the computer. The system usually offers six suggestions.**

Based on the success of adding a meme image into suggestions, we shifted our focus to suggesting funny images. In hindsight, image suggestions offer advantages over text suggestions in CAHOOTS for multiple reasons: images are often more open to interpretation than text; images are slower for users to provide on their own than entering text by keyboard; and images provide much more context on their own, i.e., an image can encapsulate an entire joke in a small space.

### Image Suggestion Strategies

In this section, we describe our most successful strategies for generating funny image suggestions based on context.

#### *Emotional Reaction Images and gifs*

Many chat clients provide emoticon libraries. Several theories of computer-mediated communication suggest that emoticons have capabilities in supporting nonverbal communications (Walther and Kyle 2001). Emoticons are often used to display or support humor (Tossell et al 2012). In popular image sharing sites such as Tumblr[2], users respond to other people's posts with emotional reaction images or gifs. In CAHOOTS, we suggest reaction images/gifs based on the emotion extracted from the last sentence.

Previous work on sentiment analysis estimates the emotion of an addresser from her/his utterance (Forbes-Riley and Litman 2004). Recent work tries to predict the emotion of the addressee (Hasegawa et al. 2013). Following this work, we first use a lexicon-based sentiment analysis to predict the emotion of the addresser. We adopt the widely used NRC Emotion Lexicon[3]. We collect reaction images and


**Figure 3. In response to text with positive sentiment, we suggest a positive emotional reaction image.**

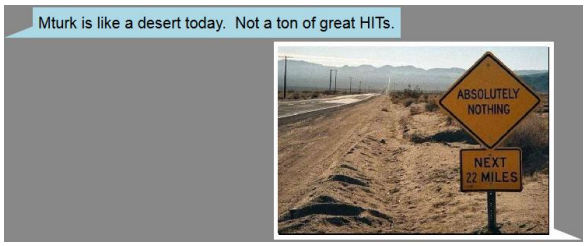



**Figure 4. In response to the utterance, the user chooses a suggestion generated by Bing image search with the query "*funny desert*".**

their corresponding emotion categories from reacticons.com. We collect reaction gifs and their corresponding emotion categories from reactinggifs.com. Then we suggest reaction images and gifs based on one of five detected sentiments: anger, disgust, joy, sadness, or surprise. An example of an emotional reaction is shown in Figure 3.

#### *Image Retrieval*

We utilize image retrieval from Bing image[4] search (Bing image) and I Can Has Cheezburger[5] (Cheezburger) to find funny images on topic. Since Bing search provides a keyword-based search API, we performed searches of the form "funny *keyword(s)*," where we chose *keyword(s)* based on the last three utterances as we found many of the most relevant keywords were not present in the last utterance alone. We considered both individual keywords and combinations of words. For individual words, we used the term frequency-inverse document frequency (tf-idf) weighting, a numerical statistic reflecting how important a word is to a document in a corpus, to select which keywords to use in the query. To define tf-idf, let $f(t, s_{-i})$ be 1 if term $t$ occurred in the $i^{\text{th}}$ previous utterance. Let $U$ be the set of all prior utterances and write $t \in u$ if term $t$ was used in utterance $u \in U$. Then weighted tf and tf-idf are defined as follows:

$$\text{wtf} = .7f(t, s_{-1}) + .2f(t, s_{-2}) + .1f(t, s_{-3})$$

[2] http://www.tumblr.com

[3] http://saifmohammad.com/WebPages/lexicons.html

[4] http://www.bing.com/images

[5] http://icanhas.cheezburger.com

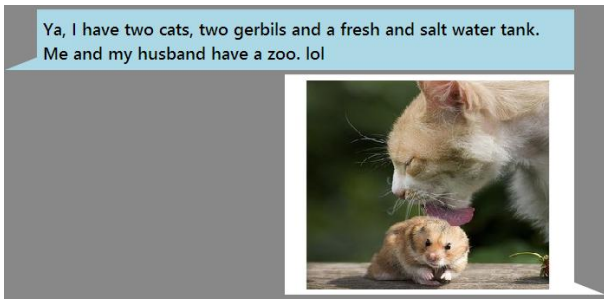



**Figure 5. An example of an utterance that generated a keyword combination *cat gerbil*, and a resulting image retrieved for the search *funny cat gerbil*.**

$$\text{idf} = \log \frac{N}{|u \in U : t \in u|}$$
$$\text{tf-idf} = \text{wtf} * \text{idf}.$$

Here $N = 43{,}370$ is the total number of utterances collected during prototyping. The weights are designed to prioritize words in more recent utterances. An example of a single keyword for Bing is shown in Figure 4.

*Combinations* of keywords were also valuable. Humor theorists argue humor is fundamentally based on unexpected juxtaposition. The images retrieved with a keyword combination may be funnier or more related to the current conversation than images retrieved with a single keyword. However, many word pairs were found to produce poor image retrieval results. Consequently, we compiled a list of common keywords, such as cat and dog, which had sufficient online humorous content that they often produced funny results in combination with other words. If a user mentioned a common funny keyword, we randomly pick an adjective or a noun to form a keyword combination from the last three utterances. An example of a query for a combination of keywords is shown in Figure 5.

### *Memes*

Meme images are popular forms of Internet humor. Coleman (2012) defines online memes as, "viral images, videos, and catchphrases under constant modification by users". A "successful" meme is generally perceived as humorous or entertaining to audiences.

Inspired by internet users who generate their own memes pictures through meme generation website and then use them in conversations in social media sites like Reddit or Imgur, our meme generation strategy writes the last utterance on the top and bottom of a popular meme template. A meme template is an image of a meme character without the captions. The template is chosen using a machine-learning trained classifier to pick the most suitable meme template image based on the last utterance, as in Figure 1(d), with half of the text on the top and half on the bottom. To train our classifier to that match text messages to meme template, we collected training instances


**Figure 6. A "Doge" meme example.**

from the Meme Generator website [6]. This website has tremendous numbers of user-generated memes consisting of text on templates. In order to construct a dataset for training machine learning models, we collected the most popular one hundred meme templates and user generated meme instances from that site. To filter out the memes that the users find personally humorous, we only keep those memes with fifty or more "upvotes" (N = 7,419). We use LibLinear (Fan et al. 2008), a machine learning toolkit, to build a one-vs-the-rest SVM multi-class classifier (Keerthi et al. 2008) based upon Bag-of-words features. Even though this is multi-class classification with one hundred classes, the classifier trained in this simple way achieved 53% accuracy (compared with a majority-class baseline of 9%).

The fact that the meme's text often matched exactly what the user had just typed often surprised a user and led them to ask, "are you a bot?" Also note that we have other strategies for generating different types of image memes, which modify the text, such as the Doge meme illustrated in Figure 6.

### *Canned Responses*

For certain common situations, we offer pre-selected types of funny images. For example, many users are suspicious that they are actually matched with a computer instead of a real person (which is partly accurate). As mentioned, we see users asking their partner if he/she is a bot. As a canned response, we suggest the results of keyword-search for "funny dog computer," "funny animal computer," or "funny CAPTCHA".

### *Responding to Images with Images*

We observed users often responding to images with similar images. For example, a picture of a dog would more likely be chosen as a response to a picture of a dog. Hence, the respond-in-kind strategy responds to an image chosen from a search for "funny *keyword*" with a second image from the same search, for any keyword.

Another strategy, called the rule-of-three, will be triggered after a user selects a respond-in-kind. The rule-of-three will perform an image search for "many *keyword*" or "not

[6] http://memegenerator.net

*keyword"*. An example is shown in Figure 7. The rule-of-three is motivated by the classic comic triple, a common joke structure in humor (Quijano 2012). Comedians use the first two points to establish a pattern, and exploit the way people's minds perceive expected patterns to throw the audience off track (and make them laugh) with the third element. In our system, when the last two images are both Bing image retrieved with the same keyword, e.g. funny dog images, we will suggest a Bing funny image with "many"+ keyword (e.g. "many dog") or "no" + keyword (e.g. "no dog") image as the third element.

In response to images, "LOL", "amused" or "not-amused" reaction images and gifs were suggested to help users express their appreciation of humor.

#### Ranking Suggestions using Reinforcement Learning

The problem of choosing images to select fits neatly into the paradigm of Reinforcement Learning (RL). Our RL algorithm, inspired by R-Max (Brafman and Tennenholtz 2003), maintains counts at three levels of specificity, for number of times a suggestion was offered and number of times it was accepted. The most general level of counts is for each of our overall strategies. Second, for specific keywords, such as "dog," we count how many times, in general, users are offered and choose an image for a query such as "funny dog." Finally, for some strategies, we have a third level of specific counts, such as a pair for each of the fifty images we receive from Bing's API. We use the "optimistic" R-Max approach of initializing count pairs as if each had been suggested and chosen five out of five times. The score of a suggestion is made based on a back-off model, e.g., for a Bing query "funny desert": if we have already suggested a particular image multiple times, we will use the count data for that particular image, otherwise if we have sufficient data for that particular word we will use the count data for that word, and otherwise we will appeal to the count data we have for the general Bing query strategy.

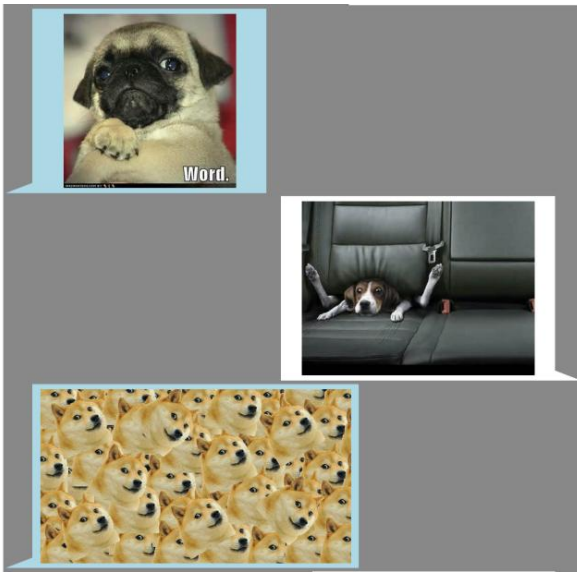



**Figure 7. The "rule of three" strategy suggests putting an image of many dogs after two dog images.**

## Experiments

To test the feasibility of CAH we performed a controlled study. Before the experiment began, we froze the parameters in the system and stopped reinforcement learning and adaptation.

#### Methodology

Participants were paid Mechanical Turk workers in the United States. Each pair of Turkers chatted for 10 minutes using: 1) CAHOOTS, our CGH system, 2) plain chat (no image suggestions), or 3) a CGH system with computer-generated images, all using the same interface In the CGH system, whenever one user sends out a message, our system automatically inserted the single top-ranking funny image suggestion into the chat, with "computer:" inserted above the message, as is common in systems such as WhatsApp. Assignment to system was based on random assignment.

We also varied the number of suggestions in CAHOOTS. We write $CAH_n$ to denote CAHOOTS with $n$ suggestions. We use CAHOOTS and $CAH_6$ interchangeably (6 was the default number determined in pilot studies). The systems experimented with were CGH, plain chat, $CAH_1$, $CAH_2$, $CAH_3$, $CAH_6$, $CAH_{10}$.

A total of 738 participants (408 male) used one of systems, with at least 100 participants using each variant. Pairs of participants were instructed how to use the system and asked to converse for at least 10 minutes. After the chat, participants were asked to fill out a survey to evaluate the conversation and the system. We asked participants to what extent they agree with four statements (based on Jiang et al. 2011), on a 7 point Likert scale. The four statements were:

- *The conversation was fun.*
- *I was able to express my sense of humor in this conversation.*
- *I felt pretty close to my partner during the conversation.*
- *I was involved in the conversation.*

#### Experiments

Averaged over the chats where our system made suggestions ($CAH_{1,2,3,6,10}$) participants selected an image in 31% of the turns. In contrast, a field study found emoticons to be used in 4% of text messages (Tossell et al. 2012).

##### *System Variant*

Figure 8 summarizes participants' responses for the four Likert questions. Results are shown for chat, CGH, and two variants of CAHOOTS. P-values were computed using a one-sided Mann-Whitney U test.

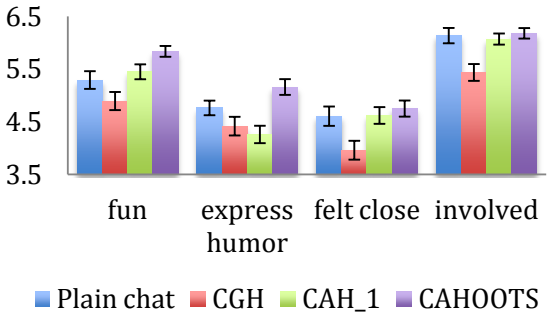



**Figure 8. Mean Likert ratings with Standard Error. 7 is strongly agree, 1 is strongly disagree, and the statements were 1. The conversation was fun. 2. I was able to express my sense of humor in this conversation. 3. I felt pretty close to my partner during the conversation. 4. I was involved in the conversation.**

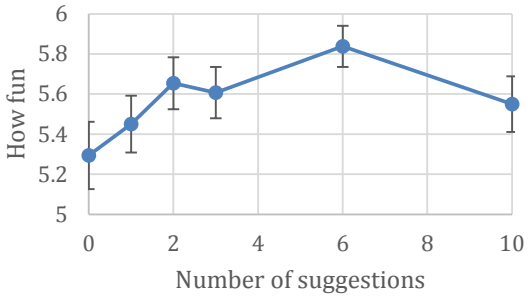



**Figure 9. Mean and SE for "the conversation was fun" as we vary the number of suggestions, with 0 being plain chat.**

*CAHOOTS vs. CGH*

Participants rated CAHOOTS conversations better on average than CGH with p-values less than 0.05 for all four questions -- more fun, able to express sense of humor, closer to partner, and more involved in conversation

It is also interesting to compare $CAH_1$ to CGH as this reflects the difference between one image automatically into the conversation and one image offered as a suggestion. Here $CAH_1$ got higher response for fun, involvement, and closeness than CGH again with $p < .05$. Curiously, participants using $CAH_1$ felt somewhat less able to express their senses of humor.

*CAHOOTS vs. plain chat*

CAHOOTS was also rated more fun than plain chat ($p < .05$), and CAHOOTS participants also reported being able to express their own sense of humor better than plain chat participants ($p < .05$). For the other two questions CAHOOTS was not statistically significantly better than plain chat.

Note that while it may seem trivial to improve on plain chat by merely offering suggestions, our earlier prototypes (especially with text but even some with image suggestions) were not better than plain chat.

*Number of Suggestions*

Figure 9 shows responses to the fun question for different numbers of suggestions in CAHOOTS. In general, more suggestions makes the conversation more fun, though ten suggestions seemed to be too many. This may be because of the cognitive load required to examine ten suggestions or simply that with many suggestions scrolling is more likely to be required to see all image suggestions.

*Effective Image Generation Strategies*

As described earlier we used several different strategies for generating images. Table 1 shows how often each type was shown and how often it was selected. The rule-of-three (inspired by our meetings with comedians) was suggested less often than some other techniques, but the rate at which it was selected was higher. Reaction images/gifs were the next most frequently selected image strategy.

| | # suggestions | % chosen |
|---|---|---|
| Bing Images | 44,710 | 10% |
| Reaction Images and gifs | 4,375 | 19% |
| Meme | 709 | 13% |
| Rule-of-three | 698 | 24% |
| Cheezburger | 537 | 7% |

**Table 1 Selection rate of the top five strategies.**

*Limitations*

Since we evaluate our system with paid workers, we have only tested the system between anonymous strangers whose only commonality is that they are US-based Mechanical Turk workers. We also asked workers to indicate with whom they would most like to use CAHOOTS: a family member, a close friend, an acquaintance, a colleague, or a stranger. Workers consistently answer that CAHOOTS would be best when chatting with a close friend who "can understand their humor."

Also, we cannot compare CAHOOTS to every kind of CGH. For example, it is possible that users would prefer a CGH system that interjects images only once in a few turns or only when it is sufficiently confident.

**Qualitative Insights**

We analyzed the content of the text and image messages as well as worker feedback from both prototyping and experimentation phases. Note participants often remarked to one another, quite candidly, about what they liked or problems with our system, which helped us improve.

Anecdotally, feedback was quite positive, e.g., "It should be used for online speed dating!" and "When will this app be available for phones and whatnot? I want to use it!" Also, note that when we offered a small number of suggestions, feedback called for more suggestions. In contrast, feedback for CGH was quite negative, such as "The pictures got kind


**Figure 10. Two workers start to talk about Bill Murray after using a reaction gif featuring Bill Murray.**

of distracting while I was trying to talk to him/her." We now qualitatively summarize the interactions and feedback.

*Humorous Images Bring New Topics to the Conversation*

Without CAHOOTS image suggestions, most of the chats focused on working in Mechanical Turk, which they seemed to find interesting to talk about. With suggestions, however, workers chose an image that suited their interests and naturally started a conversation around that image. Common topics included their own pets after seeing funny animal images, and their own children and family, after seeing funny baby images. As one worker commented: "great for chatting with a stranger, starts the conversation." An example is shown in Figure 10, where two workers start to talk about Bill Murray after using a reaction gif featuring Bill Murray.

*Image Humor is Robust*

We found CAHOOTS robust in multiple ways. First, participants had different backgrounds which made them understand images differently. For example, one participant might complain that our memes were outdated, while the other participant's feedback would indicate that they didn't even recognize that the images were memes in the first place. Nonetheless, the latter could still find the images amusing even if they didn't share the same background.

Second, we found CAHOOTS robust to problems that normal search engines face. For example, a normal search engine might suffer from ambiguity and therefore perform *word-sense disambiguation*, whereas humor is often heightened by ambiguity and double-entendres. While we didn't explicitly program in *word-sense ambiguity*, it often occurs naturally.

*Yes, and…*

A common rule in improvisational comedy, called the *yes and* rule, is that shows tend to be funnier when actors accept one another's suggestions and try to build them into something even funnier, rather than changing the direction even if they think they have a better idea (Moshavi 2001).

Many CAHOOTS's strategies lead to yes-and behaviors. An example is shown in Figure 11. On the top, the computer suggestions directly addresses the human's remark to makes the conversation funnier.

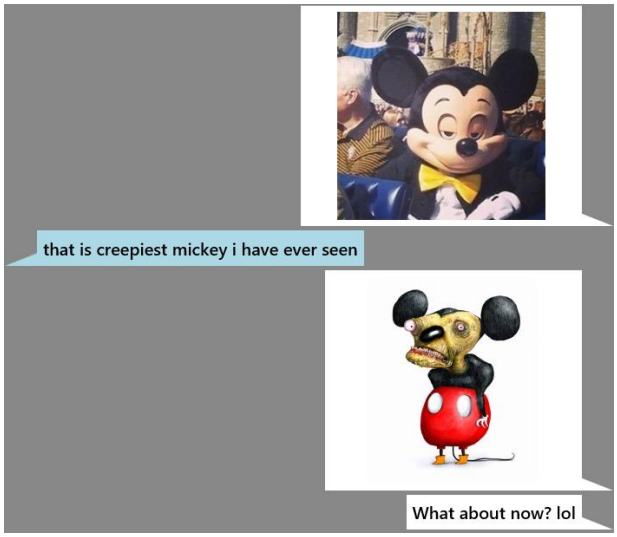



**Figure 11. An example of man-machine riffing.**

*Users Tend to Respond with Similar Images*

Humor support, or the reaction to humor, is an important aspect of interpersonal interaction (Hay 2001). With CAHOOTS, we find that users tended to respond to a funny image with a similar image to contribute more humor, show their understanding and appreciation of humor. When one user replied to her partner's image message with an image, 35% of the time the other user chose an image generated by the same strategy. Compared with two random images in a conversation, the chance that they are generated by the same strategy is 22%.

## Conclusion

In this paper we introduce the concept of Computer-Aided Humor, and describe CAHOOTS—a chat system that builds on the relative strengths of people and computers to generate humor by suggesting images. Compared to plain chat and a fully-automated CGH system, people using found it more fun, enabled them to express their sense of humor and more involvement.

The interaction between human and computer and their ability to riff off one another creates interesting synergies and fun conversations. What CAHOOTS demonstrates is that the current artificial intelligence limitations associated with computational humor may be sidestepped by an interface that naturally involves humans. A possible application of CAH would be an add-on to existing chat clients or Facebook/Twitter comment box that helps individuals incorporate funny images in computer-mediated communication.

## References

Augello, A., Saccone, G., Gaglio, S., and Pilato, G. 2008. Humorist bot: Bringing computational humour in a chat-bot system. In *Complex, Intelligent and Software Intensive*

*Systems,* 703–708.

Baym, N. 1995. The performance of humor in computer-mediated communication. *Journal of Computer-Mediated Communication*, 1(2).

Binsted, K. 1995. Using humour to make natural language interfaces more friendly. In *Proceedings of the IJCAI Workshop AI and Entertainment*.

Brafman, R., and Tennenholtz, M. 2003. R-max: a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231.

Coleman, G. 2012. Phreaks, hackers, and trolls: The politics of transgression and spectacle. *The social media reader*, 99–119.

Dybala, P., Ptaszynski, M., Higuchi, S., Rzepka, R., and Araki, K. 2008. Humor prevails! - implementing a joke generator into a conversational system. In *Australasian Joint Conference on AI*, 214–225.

Dynel, M. 2009. Beyond a joke: Types of conversational humour. *Language and Linguistics Compass*, 3(5):1284–1299.

Fan, R.-E. Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Forbes-Riley, K. and Litman, D. 2004. Predicting emotion in spoken dialogue from multiple knowledge sources. In *HLT-NAACL*, 201– 208.

Hasegawa, T., Kaji, N., Yoshinaga, N., and Toyoda, M. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *Proceedings of* the *ACL, Vol. 1,* 964–972.

Jiang, L., Bazarova, N., and Hancock, J. 2011. The disclosure–intimacy link in computer-mediated communication: An attributional extension of the hyperpersonal model. *Human Communication Research*, 37(1):58–77.

Keerthi, S., Sundararajan, S., Chang, K.-W., Hsieh, C.-J. and Lin, C.-J. 2008. A sequential dual method for large scale multi- class linear SVMs. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 408–416.

Kiddon, C. and Brun, Y. 2011. That's what she said: double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 89–94.

Morkes, J. Kernal, H., and Nass, C. 1998. Humor in task-oriented computer-mediated communication and human-computer interaction. In *Human Factors in Computing Systems*, 215–216.

Morkes, J., Kernal, H., and Nass. C. 1999. Effects of humor in task-oriented human-computer interaction and computer-mediated communication: A direct test of SRCT theory. *Human- Computer Interaction*, 14(4):395–435.

Moshavi, D. 2001. Yes and...: introducing improvisational theatre techniques to the management classroom. *Journal of Management Education*, 25(4):437–449.

Nijholt, A., Stock, O., Dix, A., and Morkes, J. 2003. Humor modeling in the interface. In *Human Factors in Computing Systems*, 1050–1051.

Petrovic, S. and Matthews, D. 2013. Unsupervised joke generation from big data. In *ACL (2)*, 228–232.

Quijano, J. 2012. Make your own comedy. In *Capstone*.

Ritchie, G. 2001. Current directions in computational humour. *Artificial Intelligence Review*, 16(2):119–135.

Sosik, V., Zhao, S., and Cosley. D. 2012. See friendship, sort of: How conversation and digital traces might support reflection on friendships. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, 1145–1154.

Shifman, L. 2007. Humor in the age of digital re-production: Continuity and change in internet-based comic texts. *International Journal of Communication*, 1(1):23.

Sjobergh, J., and Araki, K. 2009. A very modular humor enabled chat-bot for Japanese. In *Pacling* 2009, 135–140.

Strapparava, C., Stock, O., and Mihalcea, R. 2011. Computational humour. In *Emotion-oriented systems,* 609–634.

Tossell, C., Kortum, P., Shepard, C., Barg-Walkow, L., Rahmati, A., and Zhong, L. 2012. A longitudinal study of emoticon use in text messaging from smartphones. *Computers in Human Behavior*, 28(2):659–663.

Valitutti, A., Stock, O., and Strapparava, C. 2009. Graphlaugh: a tool for the interactive generation of humorous puns. In *Affective Computing and Intelligent Interaction and Workshops, 2009.* 1–2.

Valitutti, A., Toivonen, H.., Doucet, A., and Toivanen, J. 2013. "Let everything turn well in your wife": Generation of adult humor using lexical constraints. In *ACL (2)*, 243–248.

Walther, J. and D'Addario, K. 2001. The impacts of emoticons on message interpretation in computer-mediated communication. *Social science computer review*, 19(3):324–347.

Wanzer, M., Booth-Butterfield, M., and Booth-Butterfield. S. 1996. Are funny people popular? an examination of humor orientation, loneliness, and social attraction. *Communication Quarterly*, 44(1):42–52.

# A Semantic Map for Evaluating Creativity

**Frank van der Velde[1,2], Roger A. Wolf[3], Martin Schmettow[1] and Deniece S. Nazareth[1]**
[1]Cognitive Psychology and Ergonomics (CPE-BMS), CTIT, University of Twente,
Drienerlolaan 5, 7522 NB Enschede, The Netherlands
[2]IOP Leiden University, The Netherlands
[3]Saxion University of Applied Sciences,
Handelskade 75, 7417 DH Deventer, The Netherlands
f.vandervelde@utwente.nl

**Abstract**

We present a semantic map of words related with creativity. The aim is to empirically derive terms which can be used to rate processes or products of computational creativity. The words in the map are based on association studies performed by human subjects and augmented with words derived from the literature (based on human raters). The words are used in a card sorting study to investigate the way they are categorized by human subjects. The results are arranged in a heat map of word relations based on a hierarchical cluster analysis. The cluster analysis and a principal component analysis provide a set of five to six clusters of items related to each other, and as clusters related to creativity. These clusters could form a basis for scales used to rate aspects of computational creativity.

## Introduction

In his *Principles of Psychology*, published in 1890, William James introduced his definition of 'attention' as follows: "Everyone knows what attention is". Yet, debates on the distinctive features of attention continue up to the present day.

Perhaps a similar situation could be found with the notion of 'creativity'. In some way, 'everyone knows what creativity is'. But it is non-trivial to find methods by which creativity can be evaluated. Yet, when we investigate creativity, either in humans or as achievements of computational systems, we need some way to evaluate creativity. For example, we need a measure of creativity to distinguish between brain states in the neuroscientific investigation of creativity (Fink and Benedek, 2014). We also need it to assess the products of computational systems as creative or not. Indeed, the question of how computational creativity can be evaluated has been described as one of the 'Grand Challenges' of computational creativity research (Cardoso, Veale and Wiggins, 2009).

Definitions of creativity have been presented in the literature. For example, "creativity is commonly defined as the ability to produce work that is both novel (original, unique) and useful" (Fink and Benedek, 2014, p. 111). Similar characteristics are novelty and useful or value (Amabile, 1996; Hennessey and Amabile, 2010), typicality, novelty and quality (Ritchie, 2007), novelty, value, and unexpectedness or surprise (Grace and Maher, 2014), and skill, imagination and appreciation (Colton, 2008).

Each of these qualifications may capture aspects of creativity. But when they are used as criteria for the evaluation of creativity by human raters, as in the evaluations of processes or products of computational creativity, we need to validate their relation with the notion of creativity. In this context it is important to realize that an assessment (rating) performed by humans is an empirical investigation (behavioral experiment), whether or not the raters are experts or arbitrary people, and the rating scales used are instruments of measurement, which need to be validated. For this, it is not sufficient to argue that the rating scales are based on some kind of definition (no matter how sound the definition may appear to be).

Recently, Jordanous (2012a, 2012b, 2014) investigated the question of how creativity of computational creativity systems is and should be evaluated. Based on an analysis of the evaluation of creativity in the scientific literature related to computational creativity, she found that evaluation ratings (if performed at all) were based on criteria set up by the researchers themselves (or by other researchers in the literature).

To achieve a more empirical basis for rating computational creativity (i.e., not just derived from the subjective acceptance by researchers), Jordanous (2012a,b) used a statistical analysis by comparing word frequencies in scientific articles related to the study of computational creativity with word frequencies in scientific articles related to other topics. An analysis of this kind is based on the assumption that the meaning of words is related to the context in which the words are used. In particular, the meaning of a word (or aspects of it) can be determined by finding other words that co-occur with it statistically more often than can be expected on the basis on chance (Landauer and Dumais, 1997).

Based on her analysis, Jordanous (2012a,b) derived a set of 694 terms that occurred more frequently in the scientific literature related to computational creativity comparted to

other, non-related, scientific articles. On the basis of these words, she derived 14 dimensions on which creativity could be evaluated.

Here, we investigate the empirical basis for rating (computational) creativity based on empirical (behavioural) studies with human subjects. After all, ratings of creativity are conducted by human subjects, so we could also probe human subjects for the basis of these rating scales. Our aim is to arrive at a 'semantic map' of terms related to the notion creativity, which can be used to derive and compare rating scales for creativity.

To arrive at this semantic map, we conducted a study in which human subjects were asked to provide terms associated with creativity. Next, the terms associated with creativity were used in a 'reverse' association study, to see whether terms like 'creativity' are in turn associated with these terms. Then, a selected set of words based on both association studies was used in a card sorting study with human subjects. The words used in our card sorting study were augmented with a selected subset of the 694 words related to creativity based on the analysis of Jordanous (2012a,b). A card sorting study provides information about how a set of words are categorized by human subjects. Using the words based on our association studies, this in turn provides a prototype for a semantic map related to creativity.

The remainder of this article is structured as follows. First, we outline how a set of words was derived as the basis for the semantic map. Then, we present and discuss the card sort study used to derive the semantic map. Next, the prototype of the semantic map based on the card sorting study is presented and discussed. Finally, we present the conclusions and briefly discuss future work.

## Word associations with creativity

As introduced above, we conducted two word association studies. Word associations are used as a technique in experimental psychology, for example to obtain controlled stimulus material (Nelson, McEvoy & Schreiber, 2004).

In an association study, a target word is given and subjects are asked to produce words associated with the target. In a free association study a subject can give an unlimited number of associated words. In a restricted or discrete association study, the number of association words is restricted beforehand (in case of a discrete study, only one associated word can be given). A problem with a free association study is the occurrence of a chain of associations, in which (new) associated words are given not because they are associated with the target word but instead are associated with a previously given associated word. We therefore used a restricted and a discrete association study.

The aim of our first association study was to derive a set of terms associated with the word 'creativity'. For this, we conducted a restricted association study. In this study, 36 subjects between the age of 18 and 52 (29 Dutch and 7 German) were asked to give at most three terms associated with the word 'creativity' (either in Dutch or German). From this list three human raters selected a list of words on which they all agreed as words associated with creativity. This resulted in a set of 58 words.

We augmented this list by a selection of words based on the set of words derived by Jordanous (2012a,b). She analyzed two corpora of texts: one consisting of scientific articles related to the study of creativity and one consisting of scientific articles not related to the study of creativity. A statistical analysis revealed a set of 694 terms that occurred statistically more frequently in the scientific articles related to the study of creativity. In our study, this set was reviewed by three human raters. They each selected words from this set that in their view were associated with creativity. The words on which all three raters agreed were included in the set of words associated with creativity. This procedure resulted in an initial list of 32 words based on the list provided by Jordanous (2012a,b).

The list of 58 words obtained in our first association study included 10 words from the list of Jordanous (2012a,b) selected by the three human raters (see above). The list of 58 words included another eight words from the list of Jordanous (2012a,b) which were not selected by the three human raters.

In this way, we obtained a list of 80 words to be used in our second association study. In this list of 80 words, 22 words derived exclusively from the list of Jordanous (2012a,b), in the manner outlined above; 40 words were derived exclusively from the list provided by human subjects in our first association study; 18 words co-occurred in the list of Jordanous and in the human subject list obtained in our first association study.

In our second association study we used the list of 80 words obtained in our first association study, augmented with the words selected from Jordanous (2012a,b), to conduct a 'backward' (or reverse) discrete association study. That is, for each of these 80 words human subjects were asked to provide one term associated with that word. The list of words was presented in a randomized order to prevent priming effects. A subject sat in front of a screen and a keyboard in an isolated cubicle. One word at a time appeared on the screen. The subject then used a keyboard to type the answer. After that, a new word appeared. The subjects consisted of 50 students between age 19 and 27. None of them participated in the first part of the study. There were 29 Dutch and 21 German participants from whom 24 were men and 26 women. There were 25 technical students, 22 social studies students and 3 art students.

The first aim of our second association study was to obtain 'reversed' associations to the words associated with creativity (the list of 80 words outlined above). In particular, to see whether words like 'to create', 'creative' or 'creativity' are in turn associated with the words associated with the word 'creativity'. A second aim of this study was to see whether words in the list of 80 words are associated with each other.

A subset of the list of 80 words gave a 'creativity' word ("creativity", "creative" or "to create") as a (reversed) association in our second association study. In this subset, 55% of the words came from the human list derived in our

first association study, 28% from the list provided by Jordanous (2012a,b) and 17% from both lists. However, the whole list of 'reversed' associated words obtained in our second association study was used as one of the lists on which the words for the card sorting study were based, in the manner outlined below.

## Card sorting study

The list of words obtained in our first association study (augmented with words from the list of Jordanous) and the list of words obtained in our second association study were used to select the words for the card sorting study.

| | Words used in card sorting | Source: H (Human); J (Jordanous); B (Both) |
|---|---|---|
| 1. | Different | **H** |
| 2. | Artistic | **B** |
| 3. | (To) Knit | **H** |
| 4. | Extraordinary | **B** |
| 5. | (To) Think | **B** |
| 6. | Imagination | **B** |
| 7. | Thought | **J** |
| 8. | Poem | **B** |
| 9. | Mind | **H** |
| 10. | Feeling | **H** |
| 11. | Craftsmanship | **H** |
| 12. | Idea | **B** |
| 13. | Hunch | **B** |
| 14. | Innovation | **J** |
| 15. | Inspiration | **B** |
| 16. | Intelligence | **J** |
| 17. | Knowledge | **J** |
| 18. | Colours | **H** |
| 19. | (To) Craft | **H** |
| 20. | Art | **H** |
| 21. | (To) Make | **H** |
| 22. | Difficult | **H** |
| 23. | Music | **B** |
| 24. | Novelty | **B** |
| 25. | Unconventional | **B** |
| 26. | (To) Design | **H** |
| 27. | Original | **B** |
| 28. | Passion | **H** |
| 29. | Planning | **H** |
| 30. | Process | **J** |
| 31. | Painter | **B** |
| 32. | (To) Play | **B** |
| 33. | Spontaneity | **B** |
| 34. | Talent | **J** |
| 35. | (To) Draw | **H** |
| 36. | Invent | **J** |
| 37. | Unique | **H** |
| 38. | Skill | **J** |
| 39. | Renewing | **H** |
| 40. | Realise | **H** |
| 41. | Resourceful | **H** |
| 42. | Happy | **H** |

Figure 1. List of words used in the card sorting study

The selection was based on three conditions:

Firstly, a word had to appear in both lists of words. Thus, a word is considered to be strongly associated with creativity if that word is both directly and indirectly (reversely) associated with creativity. Direct association entails that the word is associated with creativity (more specifically, the word belongs to the word list of our first association study, augmented with words from Jordanous, 2012a,b). Indirect association entails that the word is associated with a word that is in turn associated with creativity (more specifically, the word belongs to the list of words obtained in our second association study).

Secondly, a word had to appear more than once as an answer in our second association study (to avoid the use of idiosyncratic words in the card sorting study).

Thirdly, the word could not be the word "creative" or any derivative of that base word, because the aim of this card sorting study was to investigate the internal semantic structure of the words strongly associated with creativity without interference from the base word "creativity" itself.

In all, 42 words were selected for the card sorting study. In the study 40 Dutch participants took part. They did not participate in any of the previous studies. Figure 1 presents the words used in the card sorting study and the source (lists) on which they are based. That is, the source consists of the list derived from our association studies (H, 19 words); the list of Jordanous (2012a,b) (J, 8 words); or both lists (B, 15 words).

Card sorting can be used to evaluate how people organize a set of items (Harloff and Coxon, 2006). Figure 2 illustrates a card sorting study with the following set of words: *keyboard*, *printer*, *mouse*, *cat*, *dog*.

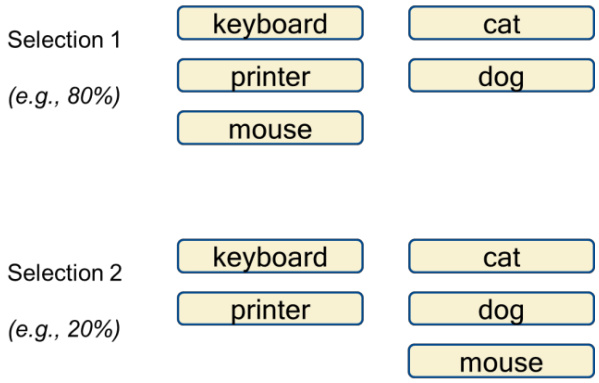



Figure 2: Example of a card sorting study

In a card sorting study, these words are printed on cards and subjects are asked to group these cards into categories[1]. If, in their view, a word cannot be placed in a category, it forms a category on its own. All words have to be selected in this way. The set of words in figure 2 could, for example, be grouped as {*keyboard*, *printer*, *mouse*} and {*cat*, *dog*} (selection 1) or as {*keyboard*, *printer*} and {*mouse*, *cat*, *dog*} (selection 2). The number of times (percentage) a particular categorization is chosen determines the (relative) strength of that categorization.

[1] One can also use an online version of a card sorting study. For an example, see https://conceptcodify.com/studies/jfvi9n5751vue9bn/via/demo_use_only_not_recording/

The results of the card sorting study with our set of 42 word associated with creativity were analyzed with a Hierarchical Cluster Analysis (HCA), using the statistical programming environment R (Salmoni, 2012). The HCA technique (Coxon, 1999) selects the two highest associated words (i.e., that most often occur together in a card sorted group) and replaces them with a single item. The associations of this item with the other words are the average of those of the two words forming the item. Continuing in this way, a hierarchical cluster can be obtained of the results of the card sorting study.

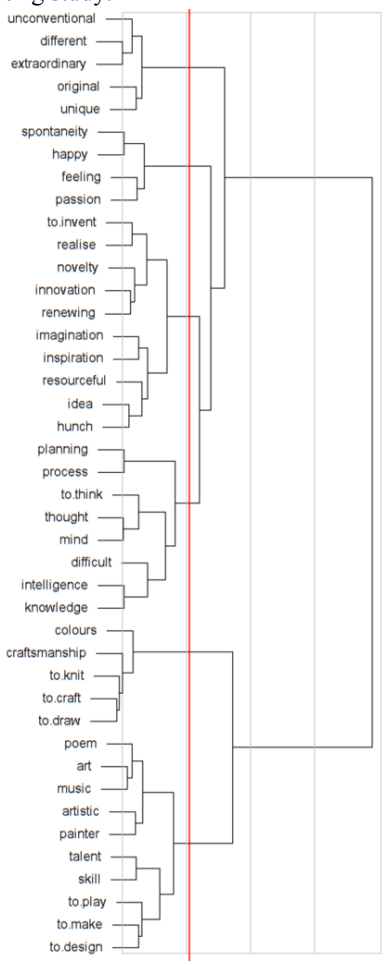



Figure 3: Hierarchical clustering of the 42 words used in the card sorting study of terms associated with creativity

The results of the HCA on the card sorting data are presented in Figure 3. The hierarchical cluster structure provided by the HCA starts with clusters of one or two words at the left and ends with two overall clusters at the right. The horizontal distances in figure 3 provide a measure of (relative) distance between clusters and subclusters. Short distances between subclusters (as between the first layer of clusters at the left of the hierarchy) suggest that they essentially form a larger subcluster. Visual inspection of the HCA suggests that a set of subclusters to the left of the red line might provide information about a meaningful classification of the words related to creativity, because the distances within these subclusters are relatively short compared to the distances between the subclusters.

Figure 4 presents a set of basic clusters of terms associated with creativity, based on the HCA presented in figure 3. They are selected (as indicated by the red line), by using the same distance from the basis as a selection measure. A basis for the selection is the observation that item-distances between clusters are substantially larger than item-distances within clusters.

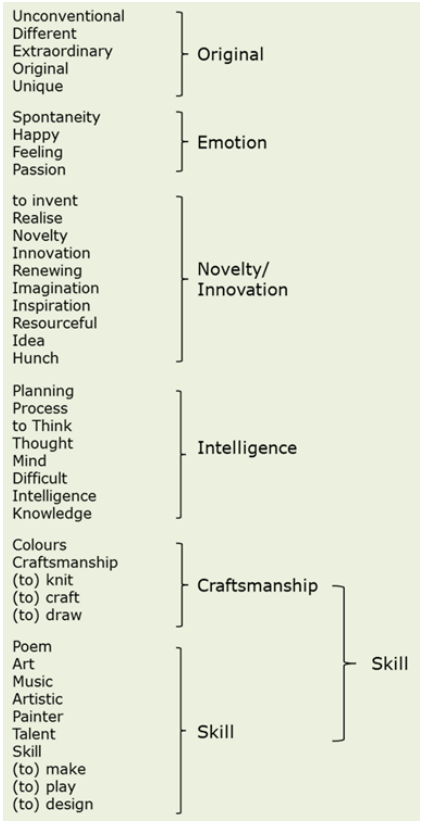



Figure 4: Tentative clusters related to creativity

Figure 4 presents six clusters and tentative cluster names. Perhaps the last two clusters could be combined into one, given that the item-distances between these clusters and the other clusters are the largest distances of the hierarchy in figure 3. This would provide the following five main clusters of items associated with the concept creativity:

- Original (originality)
- Emotion (emotional value)
- Novelty / innovation (innovative)
- Intelligence
- Skill (ability)

Before discussing these clusters we present and discuss a further analysis of the data based on the 'heat map' presentation of the results from the card sorting study.

## Heat map of card sorting results

The results of the card sorting study can also be represented in a heat map, in which the color indicates the strength of the association between two terms.

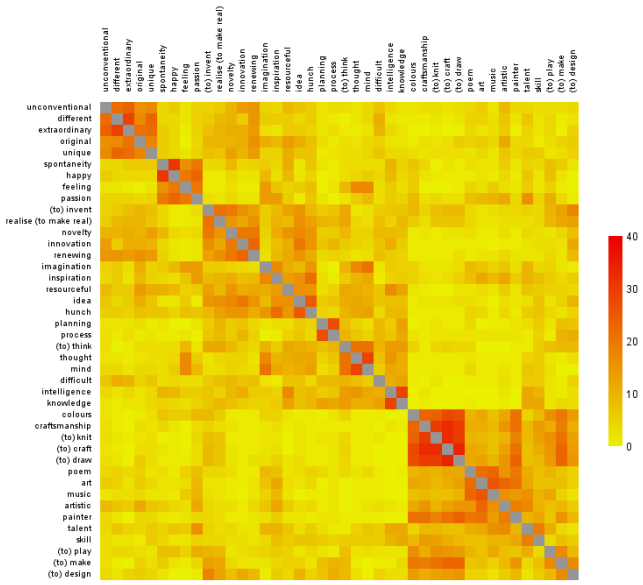



Figure 5: Heat map presentation of the card sorting results

Figure 5 presents the heat map based on the results of the card sorting study. The rows and columns in the heat map represent the words used in the card sorting study (figure 1). The words in the heat map are arranged in the order of the HCA analysis presented in figure 3. In this way, the heat map forms a matrix. The color in each matrix cell represents the number of times the row and column word corresponding to the cell belonged to the same group in the card sorting study. Given that 40 subjects participated in the study, this number can vary between 0 and 40. The heat map presents this number in terms of a color, varying from light yellow (0) to deep red (40). In the data, the lowest number was 0 and the highest number was 34. The heat map is symmetric because the words used in the card sorting study are represented as rows and as columns. For this reason, the diagonal in the heat map does not represent data from the card sorting study.

It is clear that the squares that form groups of words are related to the clusters in figure 3 (which results from the fact that the words in the heat map are arranged in the order of the HCA analysis presented in figure 3). For example, in the top left corner there is a 5x5 square that is much more red (darker) than the yellow around it. This 5x5 square belongs to a group of five words: *unconventional*, *different*, *extraordinary*, *original* and *unique*. If we wanted to label this group with one name, it could be 'original', as indicated by the cluster name in figure 4. Original is often referred to in the literature as a characteristic of creativity (e.g., Hennessey and Amabile, 2010). Also, in the right corner at the bottom we see a large group that is relatively distinct from the rest. This is the group that we labeled as 'skill' in figure 4. This group comprises a smaller 'skill' group and a 'craftsmanship' group in figure 4 (the 'craftsmanship' group stands out within the larger 'skill' group in the heat map). 'Skill' has also been related to creativity in the literature (e.g., Colton, 2008).

Yet, although the HCA structure in figure 3 and the heat map in figure 5 are based on the same data, they reveal different aspects of the semantic map based on the card sorting study of terms associated with creativity.

The HCA structure shows a metric within and between the clusters of terms related to creativity. The metric is given by the (vertical) distance that needs to be travelled in going from one word to another. So, for example, the distance between *unconventional* and *innovation* is shorter than that between *unconventional* and *skill*. This metric is not directly revealed in the heat map.

But the heat map shows that a word that belongs to a group can also be associated to words outside that group. For example, *unconventional* belongs to the 5 by 5 group referred to above, but it also has some association strength with *renewing*. These outside associations are not directly revealed by the HCA structure, due to the forced choice procedure on which the structure is based. In this way, the HCA analysis seems to miss the more global structure that is present in the results (and thus in the heat map). To analyze this more global structure, we analyzed the data in the heat map using a Principal Component Analysis (PCA).

## PCA analysis of the card sorting results

A Principal Component Analysis (PCA) of a set of data reveals the orientations (axes) along which most of the variance in the data is found (Jolliffe, 1986; Jackson, 1991). These are referred to as the Principal Components (PCs). Starting with a covariance or correlation matrix of the data, a PCA analyses the matrix in terms of its eigenvalues and eigenvectors. The highest eigenvalue corresponds to the PC along which most of the variance in the data is found. The second eigenvalue then reveals the PC along which most of the remaining variance is found. This process continues until all of the variance in the data is accounted for. Because the eigenvectors are orthogonal, a PCA shows independent sources of variance in the data.

A PCA starts with a covariance or correlation matrix of the data. For this we used the data underlying the heat map expressed in decimal fractions (based on the maximum possible score of 40). For the diagonal values we used the score 1.0 based on the assumption that a word is maximally related to itself.

One of the advantages of a PCA is that it allows a reduction of the dimensions underlying the data, by taking into account only the PCs with the highest eigenvalues.

Figure 6 presents a graph of the eigenvalues of the heat map data in descending order. This is also known as a scree graph or scree plot (Jolliffe, 1986). A rule is to use only the eigenvalues presented by the scree plot in the section before the plot levels off. In this case that would result in representing the data based on PCs corresponding to the

five highest eigenvalues (all > 2).

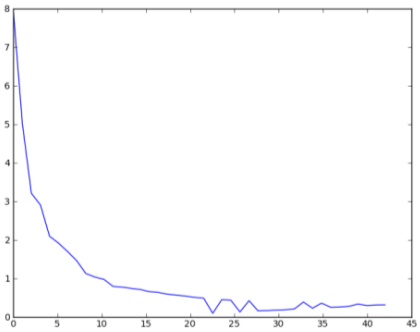



Figure 6: Scree plot of the eigenvalues in the PCA analysis of the heat map

A PCA gives the PCs of the highest variance in the data, but it does not provide an interpretation of a PC (Jackson, 1991). Looking at the heat map, however, it is clear that a substantial variance in the data results from the difference between high (red) and low (yellow) values. For a word, this difference corresponds to belonging to a subcluster (such as represented in figure 4) or not. It would seem that the first eigenvalue captures this source of variance. However, every word has both high and low values in the heat map, so this source of variance does not reveal much about the ways words belong to difference groups. Furthermore, when the values in the analyzed matrix are all positive, the coefficients of the first PC (eigenvector) are all of the same sign (Jackson, 1991).

Therefore, in figure 7 we present the words in the heat map in terms of the PCs given by the eigenvalues of the PCA of the heat map, starting with the second highest eigenvalue. The PCs are all uncorrelated, but the coefficients of a PC (eigenvector) can be correlated. These correlations are in particular affected by the signs of the coefficients (Jackson, 1991). Therefore, we group words by the signs of their coefficients for a PC. The groupings are presented in figure 7, in terms of the second to the fifth PC with the highest eigenvalues, in descending order. In figure 7, the signs of the coefficients of PC 3 to 5 are represented by the letters **P** and **N**, to indicate that different groups could have the same sign on that PC.

Figure 7 shows that the second PC (eigenvalue) separates the words in the heat map into two groups. We arranged the words in figure 7 in the manner as they appear based on 5 eigenvalues. This results in a word order (partly) different from the one found in figures 3, 4 and 5. However, it is clear that the two groups selected by the second eigenvalue in figure 7 correspond to the two largest clusters in figure 3. Thus, the first separation in the heat map (capturing most of the variance after the first eigenvalue) is between the large 'skill' cluster in figure 4 and the other words (also illustrated with the difference between the large red-like square in the bottom right corner of the heat map and the other words).

In figure 4 we selected five groups of words based on the HCA, with the 'craftsmanship' and 'skill' groups as one. In figure 7, the first four PCs also give five groups if we take the 'craftsmanship' and 'skill' groups as one. A comparison between both groupings reveals that they are quite compatible, although a few noticeable differences appear. The 'original' group in figure 4 is maintained in figure 7, with the addition of the word *renewing*, which at face value seems to be related with these words. The 'emotion' group in figure 4 is maintained as well, with the addition of *imagination* and *inspiration* (which split off with 5 PCs). So, 'emotion' may not be the correct label for this group.

| Eigenvalue 2 | Eigenvalue 3 | | Eigenvalue 4 | | Eigenvalue 5 | |
|---|---|---|---|---|---|---|
| Innovation | Innovation | P | Innovation | P | Innovation | P |
| Idea | Idea | | Idea | | Idea | |
| Planning | Planning | | Planning | | Planning | |
| Process | Process | | Process | | Process | |
| Difficult | Difficult | | Difficult | | Difficult | |
| to invent | to invent | | to invent | | to invent | N |
| Realise | Realise | | Realise | | Realise | |
| Novelty | Novelty | | Novelty | | Novelty | |
| Original | Original | | Original | N | Original | P |
| Unconventional | Unconventional | | Unconventional | | Unconventional | N |
| Different | Different | | Different | | Different | |
| Extraordinary | Extraordinary | | Extraordinary | | Extraordinary | |
| Unique | Unique | | Unique | | Unique | |
| Renewing | Renewing | | Renewing | | Renewing | |
| Hunch | Hunch | N | Hunch | P | Hunch | P |
| to Think | to Think | | to Think | | to Think | |
| Thought | Thought | | Thought | | Thought | |
| Mind | Mind | | Mind | | Mind | |
| Knowledge | Knowledge | | Knowledge | | Knowledge | |
| Resourceful | Resourceful | | Resourceful | | Resourceful | N |
| Intelligence | Intelligence | | Intelligence | | Intelligence | |
| Imagination | Imagination | | Imagination | N | Imagination | P |
| Inspiration | Inspiration | | Inspiration | | Inspiration | |
| Spontaneity | Spontaneity | | Spontaneity | | Spontaneity | N |
| Happy | Happy | | Happy | | Happy | |
| Feeling | Feeling | | Feeling | | Feeling | |
| Passion | Passion | | Passion | | Passion | |
| Craftsmanship | Craftsmanship | P | Craftsmanship | P | Craftsmanship | N |
| (to) knit | (to) knit | | (to) knit | | (to) knit | |
| (to) craft | (to) craft | | (to) craft | | (to) craft | |
| (to) draw | (to) draw | | (to) draw | | (to) draw | |
| (to) play | (to) play | | (to) play | | (to) play | |
| (to) design | (to) design | | (to) design | | (to) design | |
| Skill | Skill | N | Skill | P | Skill | P |
| Colours | Colours | | Colours | | Colours | N |
| Poem | Poem | | Poem | N | Poem | P |
| Art | Art | | Art | | Art | |
| Music | Music | | Music | | Music | |
| Artistic | Artistic | | Artistic | | Artistic | |
| Painter | Painter | | Painter | | Painter | |
| Talent | Talent | | Talent | | Talent | |
| (to) make | (to) make | | (to) make | | (to) make | N |

Figure 7: Word clusters based on the first 5 eigenvalues in the PCA of the heat map

The more substantial changes are with the 'novelty' and 'intelligence' groups in figure 4. Five words from the 'intelligence' group in figure 4 are maintained in figure 7 together with *hunch* and *resourceful* from the 'novelty' group in figure 7. Five words from the 'novelty' group in figure 4 are maintained in figure 7 together with *planning*, *process*, and *difficult* from the 'intelligence' group in figure 7.

However, despite these changes there seems to be a substantial overlap in the cluster structure obtained with HCA and PCA. The difference results from the fact that the PCA takes the overall structure of the heat map into account. The clusters as presented in figure 4 and figure 7 could be seen as a semantic map of words related to each other and, as clusters, related to creativity. This map could be used as a basis for the evaluation of creativity.

## Semantic map as a basis for evaluation

The literature provides several characteristic of creativity that could be used to evaluate processes or products of computational creativity. As outlined in the introduction these include terms like novel (novelty), original, unique, useful, value, typicality, quality, unexpectedness, surprise, skill, imagination or appreciation.

Many of these are found in the semantic map (figure 4, 7) as well. These include *novel* (*novelty*), *original*, *unique*, *skill*, and *imagination*. Other words are related to words in the semantic map. For example, unexpectedness or surprise are related to *unconventional* and *extraordinary*. The fact that words used in the literature are also found in the semantic map based on empirical investigations underscores their relation with creativity and justifies their use in assessing creativity.

However, some words reported in the literature are notably absent in the semantic map. One of those is the word 'useful'. Although often referred to as a characteristic of creativity (Amabile, 1996; Hennessey and Amabile, 2010; Fink and Benedek, 2014), it is not found in the semantic map. This raises the question of whether humans would qualify useful as related to creativity, and thus as a dimension on which creativity could or should be evaluated.

Because 'useful' did not emerge in our word association studies, we could not investigate its relation with the other terms in the card sorting study. But in a follow up study we will include 'useful' as an item to study its relation to other words related to creativity and to 'creativity' itself in a card sorting study. The outcome will enhance our insight in the way useful and creativity are related as seen by human subjects (instead of by assumption or definition).

One reason of why useful was not included may have resulted from the fact that we asked for terms associated with creativity without any further instruction or direction. It might be that when more specific instructions are given, for example to relate terms to creativity in a particular task or domain, terms like useful might appear.

Hence, another venue of research is to investigate semantic maps related to creativity within specific domains (e.g., music, poetry, architecture), to see if differences between these maps are found. If so, that would argue for more specific forms of evaluation to be used for these domains.

Yet another venue of research is to investigate whether semantic maps (whether or not related to specific domains) also differ between languages. In our association studies (but not the card sorting task itself) we used both Dutch and German native speakers. We could not find significant differences between the two. But this could be related to the similarity between both languages.

The main clusters as presented in figures 4 and 7 could be used to develop rating scales for evaluating the creativity of artificial systems and humans. All of the terms in a cluster could be used as dimensions on which creativity is rated, each one as an example of the main cluster to which it belongs. An analysis of the ratings in terms of the cluster structure could then be related to the clusters found in the semantic map. That is, if the clusters in the semantic map reflect the notions that humans have about creativity, they would also determine the way they evaluate creativity. In that case, evaluations using terms within a cluster would be related to each other and between cluster evaluations would reflect the between cluster structure in the map.

This procedure could also be used for the more domain specific semantic maps, if they are found. In that case, these maps could be used for the evaluation of domain specific forms of creativity and the results of the evaluation could be compared with the structure of the maps.

When more semantic maps are investigated a more complete structure of the semantic relations with creativity will emerge. By comparing this with evaluations of creative processes and products (both computational and human) we will develop a more complete picture of how semantic relations with creativity influence the evaluation of creativity.

The empirically derived semantic maps related to creativity could also be used to develop and evaluate experimental paradigms for investigating the neural basis of creativity. This might begin to unravel the diverse and sometimes apparently conflicting results obtained in the neuroscientific research of creativity (Arden et al., 2010; Dietrich and Kanso, 2010; Sawyer, 2011; Fink and Benedek, 2014).

## Effective use of semantic map in evaluation

To use the concepts in the semantic map as tools for evaluation we need to develop and test rating scales based on these concepts. Here, a number of considerations play a role and should be addressed.

The first one is the number of rating scales that can be used effectively. Using all concepts in the map would result in a large set of scales that could be ineffective. We can study this by using the rating scales based on these concepts in pilot evaluations and compare the scales using factor analysis. In this way we can investigate again whether concepts from the same cluster are used in the same way in an evaluation. If so, these rating scales could then be used as alternatives between evaluations. Or they could be used as alternatives within an evaluation (between or within subjects).

The second one concerns the subjects that would perform an evaluation. One option is to use experts in a given domain. Another option is to use the users of a domain in an evaluation. Here, given that the subjects in our studies were students, one can think of creative domains like visual art in gaming (and movies), dance music (and other forms of 'pop' music) and the use of new media like YouTube. Students certainly are involved here as users, and users to a large extent determine success in these domains and thus the way in which these domains develop. It is too simple to argue that only experts determine how forms of creativity develop. Users play a substantial role in that too (as they have also done in the past).

Given a set of rating scales, we can also compare evaluations by experts with that of users. An interesting topic

of research here is whether experts in a domain would have a different conceptual structure related to creativity compared to users or whether they would have a similar conceptual structure (as in the semantic map) but would use it differently in an evaluation. This could consist of a different factorization of the rating scales with evaluations performed by experts compared to users.

## Conclusions

An empirical basis for the evaluation of creativity is needed because evaluations, as conducted by human raters, are empirical investigations. Hence, the assumptions underlying these investigations, such as the rating scales used, need to be validated. We presented a semantic map of terms related to creativity based on human association and card sorting studies. The semantic map as presented here can be further developed by investigating domain specific aspects of terms related to creativity and the use of other terms often reported as related to creativity in the literature.

To derive the semantic map in the card sorting study, we augmented the words based on our human association studies with words reported in the literature that were based on a statistical analysis. Interestingly, there is an overlap in the set of words formed by the two methods, but there are also some differences. Further investigations could reveal how these methods are related and if they are both needed (as complements) to arrive at more objective procedures for the evaluation of computational (and human) creativity.

## Acknowledgements

We thank Saskia Hartmann and Janina Roppelt for their assistance. The work presented here was funded 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

Amabile, T. M. (1996). Creativity and innovation in organizations (Vol. 5). Boston: Harvard Business School.

Arden, R., Chavez, R. S., Grazioplene, R. and Junga, R. E. (2010). Neuroimaging creativity: a psychometric review. Behavioural Brain Research, 214:143156.

Coxon, A. P. M. (1999). Sorting data: Collection and analysis. London, United Kingdom: Sage Publications.

Cardoso A, Veale T, Wiggins GA. (2009). Converging on the divergent: the history (and future) of the international joint workshops in computational creativity. AI Mag, 30(3):15–22.

Colton S. (2008). Creativity versus the perception of creativity in computational systems. In: Proceedings of AAAI symposium on creative systems, p. 14–20.

Dietrich, A. and Kanso, R. (2010). A review of EEG, ERP, and neuroimaging studies of creativity and insight. Psychological Bulletin, 136:822–848.

Fink, A. & Benedek, M. (2014). EEG alpha power and creative ideation. Neuroscience and Biobehavioral Reviews, 44(100): 111–123.

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 ICCC2014. http://computationalcreativity.net/iccc2014/proceedings/

Harloff, J., & Coxon, A. P. M. (2006). How to Sort. A Short Guide on Sorting Investigations. http://www.methodofsorting.com/HowToSort1-1_english.pdf (GNU Documentation License).

Hennessey, B.A. & Amabile, T.M. (2010). Creativity. Annual Review of Psychology, 61, 569-598.

Jackson, J. E. (1991). A user's guide to principal components. New York: Wiley.

James, W. (1890). The Principles of Psychology. New York: Henry Holt, Vol. 1, pp. 403–404.

Jolliffe, I. T. (1986). Principal component analysis. New York: Springer.

Jordanous, A. (2012a). Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application. Ph.D. Dissertation, University of Sussex, Brighton, UK.

Jordanous, A. (2012b). 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. (2014). Stepping Back to Progress Forwards: Setting Standards for Meta-Evaluation of Computational Creativity. In S. Colton, D. Ventura, N. Lavrač and M. Cook (Eds.) Proceedings of the Fifth International Conference on Computational Creativity ICCC-2014, June 10-13, 2014, Ljubljana, Slovenia (pp. 129-136).

Landauer, T. K. and Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. Psychological Review, 104, 211-240.

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.

Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. Minds and Machines, 17(1), 67-99.

Salmoni, A. (2012). Open card sort analysis 101. http://www.uxbooth.com/articles/open-card-sort-analysis-101/

Sawyer, K. (2011). The cognitive neuroscience of creativity: A critical review. Creativity Research Journal, 23(2):137–154.

# Human Competence in Creativity Evaluation

**Carolyn Lamb, Daniel G. Brown, Charles L.A. Clarke**
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
c2lamb@uwaterloo.ca, dan.brown@uwaterloo.ca, claclark@plg.uwaterloo.ca

**Abstract**

We investigate the performance of non-expert judges in using leading computational poetry evaluation metrics to evaluate poetry written by humans. We find that regardless of the model used, non-expert judges are very poor at using metrics to evaluate creativity, even displaying the reverse of the desired rating pattern, preferring novice poetry to professional poetry. We discuss likely reasons for this finding and the implications for the evaluation of computational creativity. Researchers using human judges should be aware that using a metric or structured evaluation does not negate the need for judge expertise.

## Introduction

An increasingly important debate in Computational Creativity is the development of standardised evaluation methods. There are many reasons why it is desirable for computers to recognize and evaluate creativity, including the assistance of humans in creative acts, understanding of the creative human mind, and the AI application of teaching computers to behave creatively themselves. However, it is not clear how exactly one would go about distinguishing more creative from less creative output. Two important camps in this debate are those who use a metric with specific criteria (e.g. (Pease, Winterstein, and Colton 2001; Ritchie 2007; Colton 2008a; Colton, Pease, and Charnley 2011) and those who prefer a consensual assessment based on the agreement of expert judges, without specific criteria (Amabile 1983).

While the Consensual Assessment Technique has been rigorously tested (see e.g. (Kaufman, Baer, and Cole 2009)), specific metrics used in the field of Computational Creativity have not. We therefore undertook an empirical test of four such metrics from the existing literature. These metrics evaluate a product's creativity based on (for example) its novelty, value, skill and other qualities, or on some calculation involving these qualities.

We collected poems generated by humans at various levels of skill. We then recruited a large number of humans to evaluate the poems on the criteria used in our selected metrics. Our results were very counter-intuitive. On nearly every criterion, our judges significantly preferred amateur, unskilled poems to the work of professional poets—the reverse of what one would expect.

Poetry is a rarefied field, and we suspected that the reversed results were caused by untrained raters having difficulty understanding the professional poems. Such poetry might not be accessible to an untrained reader. We ran the experiment again with poems written for children. This second experiment did not produce reversed results, but any power of the criteria to differentiate between good and bad poetry was reduced to noise.

Our experiments show that non-expert judges do not apply creativity metrics appropriately to poetry. Of course, the Consensual Assessment Technique already mandates the use of expert judges for this reason. Non-experts in a consensual assessment have poor inter-rater reliability and poor agreement with the judgments of experts (Kaufman, Baer, and Cole 2009). However, our research shows that this problem also applies to judgments made with specific criteria. Using such criteria is not an escape from the issue of judge selection. Moreover, beyond simply losing reliability, the use of non-expert judges can produce the exact opposite of the intended result.

Many evaluations in computational creativity today are still done by the researchers themselves (Colton, Goodwin, and Veale 2012; Norton, Heath, and Ventura 2010; Riedl and Young 2006; Smith, Hintze, and Ventura 2014) or by a group of human volunteers whose expertise in creativity is not discussed (Burns 2015; Gervás 2002; Karampiperis, Koukourikos, and Koliopoulou 2014; Llano et al. 2014; Monteith, Martinez, and Ventura 2010; Norton, Heath, and Ventura 2013; Román and y Pérez 2014). For robust evaluation, it may turn out that neither of these approaches is sufficient.

## Background and Related Work

The past 25 years of computational creativity research owe much to Boden's (Boden 1990) work on the meaning of creativity. Boden focuses on creativity as the exploration and transformation of conceptual space. While Boden's book does not give a definition which can be broken down into formulaic parts, she does repeatedly mention the need for creative systems to produce works which are both novel and valuable. Subsequent researchers have built on her work to propose numerical metrics.

Ritchie, the first such researcher, proposes that human creativity is evaluated according to the criteria of Novelty

("To what extent is the produced item dissimilar to existing examples of that genre?") and Quality ("To what extent is the produced item a high-quality example of that genre?") (Ritchie 2001). For computational creativity, he proposes replacing Novelty with Typicality—as a computer program must first be able to generate plausible examples of a type of creative product before attempting to make ones dissimilar from what has gone before. Ritchie then suggests various tentative criteria, such as "high quality items should make up a significant proportion of the results", for evaluating a system based on its Typicality and Quality over several runs. The presence of these composite criteria implies that using Typicality and Quality measurements directly for creativity evaluation, without further analysis, may be overly simplistic. Nevertheless, one can easily imagine common-sense constraints on the base measurements. For example, while the Quality measurement could be used in various ways, one would certainly not expect creative poems to have a lower average Quality than uncreative ones.

Ritchie's model has been used to evaluate creative systems in practice (e.g. (Gervás 2002; Tearse, Mawhorter, and Wardrip-Fruin 2011)). Other researchers performing similar work focus on Novelty rather than Typicality, a choice more in line with Boden's work. For example, Pease *et al.* (Pease, Winterstein, and Colton 2001) suggest a variety of ways to formally measure both Novelty and Value (a synonym of Quality).

Some difficulties in the Boden-based models, particularly Ritchie's, have been illuminated through experience. Many of Ritchie's composite criteria are based on comparisons with an inspiring set of existing work. In the absence of a quantitative measure for similarity between creative products, such criteria are difficult to evaluate (Gervás 2002). Ventura's RASTER thought experiment (Ventura 2008) also claims to illustrate flaws in Ritchie's model: a highly uncreative system, generating works completely at random, can technically be said to meet the criteria. However, the RASTER thought experiment uses images from a Web search to guide output, without considering those images an inspiring set. It also fails to consider typicality and quality independently, which renders many criteria inapplicable. Ventura suggests that the inapplicability of these criteria, in and of itself, is a reason to treat a system with suspicion.

Another metric, Colton's Creative Tripod (Colton 2008a), judges creative work by whether it appears to be skillful, appreciative, and imaginative. Colton's tripod has frequently been used to evaluate creative systems (Smith, Hintze, and Ventura 2014; Chan and Ventura 2008; Monteith, Martinez, and Ventura 2010; Young, Bown, and others 2010) or to guide their development (Norton, Heath, and Ventura 2010; Colton 2008b). A weakness of the tripod is that specific definitions for the three criteria are not provided. It has been pointed out (Bown 2014) that this provides too much opportunity for authors to make impressionistic statements about why their system meets the criteria, without rigorous, falsifiable inquiry into whether its performance in these areas is sufficient. Even the intentionally uncreative RASTER (Ventura 2008) is argued to meet Colton's criteria in this manner.

Colton *et al.* have added many words to the tripod since its construction, including Learning, Intentionality, Accountability, Innovation, Subjectivity, and Reflection (Colton et al. 2014). However, since the majority of recent work implementing the tripod uses only the original three words, we focus our research on these original three.

Another proposal by Colton *et al.* is the IDEA model (Colton, Pease, and Charnley 2011), in which an ideal audience rates a creative product according to Wellbeing (how much they likes the product) and Cognitive Effort (how prepared they are to spend effort thinking about and interpreting it). Like the criteria of Ritchie's model, Wellbeing and Cognitive Effort can be combined to measure different aspects of a product's reception. For example, if the variance in Wellbeing is high, a product would get a high score on "Divisiveness".

Many other standardized metrics for evaluating a creative system have been proposed. Jordanous's SPECS model (Jordanous 2012) incorporates many criteria based on cultural beliefs about the meaning of creativity, including criteria similar to Novelty and Value. Burns's EVE' model defines creativity as a combination of Surprise and Meaning, and has been applied to humorous poetic advertisements (Burns 2015), humorous haiku (Burns 2012) and, in thought experiment form, to line drawings (Burns 2006). Other new metrics either proposed or used ad hoc in the past ten years come from varied sources including Piaget's theories of cognitive development (Aguilar and Pérez y Pérez 2014), theories about quality in a specific art form (Das and Gambäck 2014; Rashel and Manurung 2014; Pearce and Wiggins 2007), interestingness (Román and y Pérez 2014; Gervás 2007), and many others (Brown 2009; Lehman and Stanley 2012; Llano et al. 2014; Monteith et al. 2013; Norton, Heath, and Ventura 2013).

Very rarely have any such metrics been validated through direct use on human-generated products. A few researchers have used the metrics to compare computational products to human-generated products. Monteith *et al.* compare human-composed to computer-composed music using an operationalization of Colton's tripod (Monteith, Martinez, and Ventura 2010). The computer music did better at expressing specific emotions (Skill) but the human music sounded more like "real music" (Appreciation). Burns tested his EVE' model on human products (Burns 2015) and found good correspondence between his model and human ratings; Surprise multiplied with Meaning accounted for 70% of the variability in ratings of Creativity.

Binsted *et al.* built a system, JAPE, to generate riddles (Binsted, Pain, and Ritchie 1997), and evaluated it using children's responses to criteria similar to those which would later form Ritchie's model: "Was that a joke?" (Typicality) and "How funny was it?" (Quality). JAPE's jokes were compared to human jokes and to two categories of human-generated non-jokes. Binsted *et al.* found that children rate human-generated jokes as more typical and of higher quality than non-jokes. JAPE's jokes were somewhere in between. Ritchie *et al.* performed further tests on this data and repeated the study with college students (Ritchie et al. 2008). Thir results were broadly the same, but there was low inter-

rater reliability, especially on Quality.

While we focus on four specific metrics in our work, we do not mean to imply that these metrics represent four completely separate schools of thought. Instead, all four are influenced by each other and by prior work such as Boden's. What they all have in common is the idea of decomposing creativity into sub-concepts, then measuring creativity by somehow measuring and combining other criteria. For example, under Ritchie's model, if one can calculate the Typicality and Quality of a creative work, one can then (by some means, perhaps a complex one) calculate the work's level of creativity. This contrasts to the Consensual Assessment Technique, in which judges rate creativity however they see fit. The advantage of a metrical perspective is that it invites standardized quantitative calculation and avoids circularity. We use four metrics from the literature to represent a range of influential perspectives within the paradigm of metrical assessment. Our aim is to add to our understanding of metrical assessment of creativity as a whole.

## Experiment I

### Method

We tested 4 common metrics for creativity evaluation: Ritchie's model, Pease *et al.*'s novelty and value criteria, Colton's creative tripod, and the IDEA model. These metrics are easy to test on human poetry since they focus on the creative product and not on the process. Since none of these metrics have been put into a standardized questionnaire form, we constructed our own five-point Likert scale-based rating system for each. Each participant was only shown the questions for one of the four metrics, not all four. The questions we used are as follows:

**Ritchie's model**

- This resembles other poems I have read. *(Typicality)*
- This is a high quality poem. *(Quality)*
- I don't think this is a very good poem. *(Quality, reverse coded)*
- This is not a poem. *(Typicality, reverse coded)*

**Pease's criteria**

- This is a high quality poem. *(Value)*
- This poem is not like other poems I have seen before. *(Novelty)*
- I don't think this is a very good poem. *(Value, reverse coded)*
- This poem is clichéd. *(Novelty, reverse coded)*

**Colton's Creative Tripod**

- The author of this poem seems to have no trouble writing poetry. *(Skill)*
- The author of this poem is imaginative. *(Imagination)*
- The author of this poem understands how poetry works. *(Appreciation)*
- The author of this poem isn't very good at writing poetry. *(Skill, reverse coded)*
- The author of this poem isn't bringing anything new or different into the poem. *(Imagination, reverse coded)*
- The author of this poem doesn't really know anything about poetry. *(Appreciation, reverse coded)*

**IDEA model**

- I like this poem. *(Wellbeing)*
- I am willing to spend time trying to understand this poem. *(Cognitive Effort)*
- This poem makes me unhappy. *(Wellbeing, reverse coded)*
- This poem is not worth bothering with. *(Cognitive Effort, reverse coded)*

It should be noted that the construction of questions to represent abstract concepts from existing models is a potential source of error. For example, the IDEA model's Wellbeing criterion is based on like or dislike of a poem; it is not clear how an ideal reader would respond if they appreciated a poem but found it very sad. Appreciation in Colton's tripod, despite the lack of strict definitions of Colton's terms, also arguably refers to a creator's ability to evaluate its own work, rther than its ability to understand its field in general. However, researchers such as Norton et al (Norton, Heath, and Ventura 2010) refer to the Appreciation part of the tripod when training computers to apply labels to pre-existing images, implicitly lending support for the latter interpretation. After all, to evaluate one's own art one needs to be able to understand and evaluate art in general. A fully robust set of questions for a standardized questionnaire would require repeated testing and refinement in a variety of contexts; we have not yet reached the point of performing such tests.

### Data

For this experiment we used three hand-collected data sets of contemporary poetry written by humans. Each set contained 30 short poems in English of between 5 and 20 lines; we stuck to contemporary poetry so as to avoid different eras of poetry becoming a confounding factor, and so as to minimize the probability that a study participant had read the poems before. In no case did more than one poem by a single author appear across data sets.

For our purposes, we assumed that poems published in professional venues are more creative than poems written by novices. That is, we assumed that the editors of poetry magazines are experts and that their opinion strongly correlates with the actual creativity of the poetry published. This is, of course, debatable. Editors are sure to have specific cultural tastes and biases, but since all human judgments of creativity are culturally situated we find it an acceptable simplifying assumption.

The **Good** data set was composed of poems from Poetry Magazine between November 2013 and April 2014. Poetry Magazine is a very long-established, professional magazine which can reasonably be considered to contain the work of the most critically acclaimed mainstream literary poets working today. All poems meeting the length and non-duplication requirements and appearing in the magazine during this time window were selected, with the exception of a

| Metric | Criterion | Good | Medium | Bad | $F$ |
|---|---|---|---|---|---|
| Ritchie | Typicality | 0.20 | 0.41 | 1.23 | 10.6** |
| | Quality | 0.23 | 0.67 | 1.40 | 10.2** |
| IDEA | Wellbeing | 0.78 | 1.14 | 1.54 | 13.9** |
| | Cognitive Effort | 0.60 | 0.94 | 1.46 | 14.9** |
| Colton | Imagination | 0.75 | 1.16 | 1.07 | 2.3 |
| | Appreciation | 0.67 | 1.11 | 1.68 | 8.3** |
| | Skill | 0.44 | 0.84 | 1.40 | 7.4* |
| Pease | Novelty | 0.96 | 0.80 | 0.49 | 9.8** |
| | Value | 0.17 | 0.44 | 0.72 | 3.6 |

Table 1: Average ratings and $F$ scores for poem categories according to each metric. Each component is scored between -4 and +4. Significant results ($p < 0.05$) following Bonferroni correction are marked with a *, or ** if highly significant ($p < 0.01$).

few which were discarded due to complex visual formatting and two which were discarded due to experimenter discomfort. The remaining 30 poems comprised the Good data set.

The **Medium** data set was composed of 2 poems each from 15 lesser-known online magazines. Some of these were magazines devoted exclusively to poetry while others were a combination of poetry and prose. Each magazine pays a token amount (between US $5 and $10) per a poem. For each magazine, the most recent 2 poems meeting length and non-duplication requirements were chosen for the data set, with a single exception in which one poem was discarded and the third-most-recent poem chosen as a replacement. This added up to a Medium data set of 30 poems.

The **Bad** data set was composed of poems by unskilled amateur poets. We chose these poems by going to the Newbie Stretching Room at the Poetry Free-For-All, an online poetry critique forum. This section is for newcomers who have not posted poetry on the forum before; both experienced moderators and other newcomers can comment on the poems. We chose poems meeting the length and non-duplication requirements from this section, and discarded any which had received positive feedback from a moderator. Most of the chosen poems received comments from moderators instructing the author to read introductory articles on how to improve; a few had more specific, pointed comments. (Example: "This is dreadfully bad beginner's doggerel that fails for many, many, many reasons.") Selecting the most recently posted poems which fit these requirements resulted in a Bad data set of 30 poems.

Finally, we collected a **Test** data set containing 6 texts which were the same length as the chosen poems, but were obviously not poems. 3 of these were snippets from business news, and 3 from sports news.

These data sets are all available upon request.

## Collection

We recruited study participants on Crowdflower, a crowd-sourced microtasking website. In order to minimize cultural and linguistic difference as a confounding factor, participants were limited to those living in the United States.

Each participant was given six poems at a time, selected from any or all of the data sets, and shown the questions for only one of the four metrics. The participant was then asked to rate each poem based on that metric. Participants could rate poems repeatedly up to a maximum of 36 poems per participant per metric. We collected enough responses to amass 20 responses on each metric for each poem.

Participants were not shown the headings or names for the metric they were given, nor the names of the criteria on which the questionnaire items were based. Our justification for separating the metrics in this manner, and for coding Quality and Value separately even though the questions are identical, is that we were interested in taking each metrical approach as a whole, rather than mixing and matching criteria from all the metrics.

For each criterion, we ran a single-factor ANOVA comparing the Good, Medium, and Bad poems' scores on that criterion. Since there were three two-criterion metrics and one three-criterion metric, we ran nine ANOVAs and then applied a Bonferroni correction for nine hypotheses. The null hypothesis was that, for all metrics, participants' responses to Good, Bad, and Medium poems would be drawn from an identical distribution. The alternative hypothesis was that the distributions would not be identical: that is, that on some criteria, poems from one or more categories would be rated differently than others.

## Results

Results were the opposite of what we expected. For most criteria, participants rated Bad poems significantly (at $p = 0.05$ or better, following Bonferroni correction) more highly than Good ones. The exception was Novelty, in which Good poems were rated more highly than Bad. For Imagination and Value, the differences between categories were not significant. Exact F and p-values for each of these criteria are shown in Table 1.

This was a highly surprising result since it is not attributable to rater incompetence or failure to pay attention. Incompetent crowd workers who failed to pay attention might give the same score to all poems, or give random scores. Our raters, however, had significantly different reactions to the different groups. Adding test questions and bonuses to incite workers to pay more attention did not change the overall response pattern. This indicates that crowd workers can differentiate between these groups—but their preferences are different from what we had imagined.

The results for Medium poems were more ambiguous. We ran a Fisher's Least-Significant Difference Test to under-

| **Metric** | Criterion | C-Bad | C-Good | *t* |
|---|---|---|---|---|
| Ritchie | Typicality | 1.25 | 1.46 | 0.48 |
| | Quality | 0.08 | 0.77 | 0.13 |
| IDEA | Wellbeing | 1.30 | 1.61 | 0.35 |
| | Cognitive Effort | 0.11 | 0.63 | 0.23 |
| Colton | Imagination | 0.65 | 0.80 | 0.74 |
| | Appreciation | 1.12 | 1.50 | 0.42 |
| | Skill | 0.84 | 1.20 | 0.40 |
| Pease | Novelty | 0.32 | 0.24 | 0.74 |
| | Value | 0.11 | 0.34 | 0.62 |

Table 2: Average ratings and *t* scores for children's poem categories according to each metric. At $p < 0.05$, there were no significant differences found after Bonferroni correction

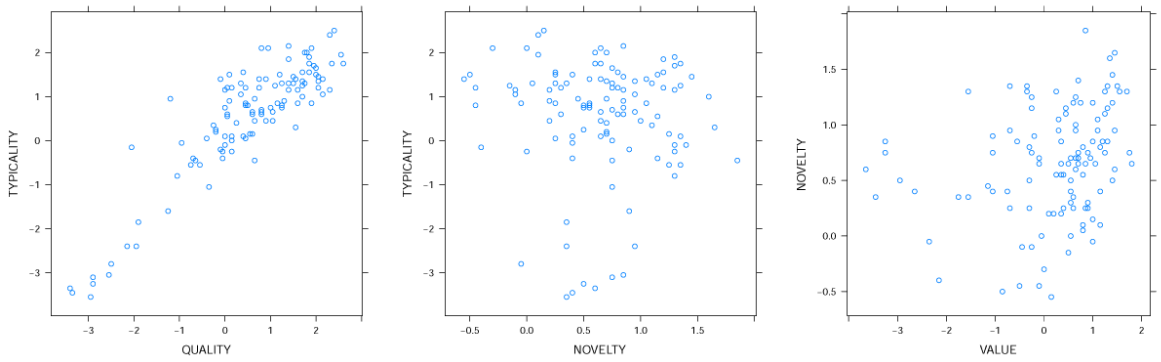



Figure 1: Sample scatterplots showing relationships between Novelty, Typicality, and Quality for poems in all of the data sets from both experiments.

stand the pairwise relationships between the three groups, again applying Bonferroni correction. Although Medium poems generally rated more highly than Good poems, in no case was this statistically significant. The difference between Medium and Bad poems, meanwhile, depends on the criterion. For Typicality, Novelty, and Effort, Medium poems were significantly different from Bad ones. For the other criteria, there was no significant difference between Medium poems and either other group.

## Experiment II

One potential explanation for why participants preferred Bad poems is that the Bad poems were more accessible. Poems from a prestigious literary journal may be difficult to understand due to heavy allusiveness and other poetic conventions. To test the inaccessibility hypothesis, we ran a second experiment focusing on poems written with children as the indended audience.

The **C-Good** data set was composed of children's poems found in the Children's Poetry section of the Poetry Foundation website in November 2014. The same selection constraints were used as with the first data set: poems were between 5 and 20 lines in length and no poet's work was used more than once. We also excluded poems by poets born prior to the 20th century. We collected a total of 10 C-Good poems, by authors such as Kenn Nesbitt and Shel Silverstein.

The **C-Bad** data set was composed of poems posted on the Family Friend Poems forum by amateur poets between September and November 2014, meeting the length and author uniqueness criteria. 10 such poems were selected. As there is no expectation of detailed critique at Family Friend Poems, we did not filter poems by critiques given as we did with the Bad adult poems. In fact, most responses to these poems were brief and complimentary (e.g. "Brilliant. Loved it 10"), even when the poems made large mistakes with meter and rhyme.

These poems were randomized and evaluated in the same way as the poems from Experiment I, on the criteria from the same four metrics. Since there are only two data sets in Experiment II, a *t*-test was performed on every criterion to detect differences in how the children's poems were rated.

### Results

The children's poem results lacked the effect seen in the adult poems. Participants rated C-Good poems more highly than C-Bad poems on most criteria, but these results were not statistically significant. A power analysis determined that this was not solely a result of the smaller size of the second study; hundreds of poems would have been needed for significance. Using children's poems removed raters' preference for bad poems, but did not introduce a preference for good poems above the level of noise.

### Correlations within and between metrics

It is not empirically clear if the different criteria from the different metrics actually elucidate different components of creativity. We investigated this by combining the data from Experiments 1 and 2, then generating scatterplots and correlation coefficients to examine the relationships between different criteria. With the exception of Novelty, all criteria were fairly well-correlated with each other ($0.65 < r < 0.99$), and scatterplots showed approximately linear relationships. Novelty had no significant positive, negative, or non-linear relationship with any other criterion. Example scatterplots are given in Figure 1

The high correlations between different criteria may indicate that these criteria—especially those with extremely high correlations, such as Skill and Appreciation at $r = 0.99$—are not actually separate concepts, or at least, are not adequately separated in the minds of raters when phrased as our questionnaire phrases them. An alternative interpretation, suggested by a reviewer to this paper, is that the high correlation is a good thing: if all criteria measure some aspect of creativity, then one would expect them all to change in similar ways along with an underlying change in creativity.

## Discussion

Our goal was to illustrate differences in effectiveness between different metrics, but we ended up finding something different. When using metrics, rather than simply asking judges how creative something is, the purpose is to be more objective and ensure that the appropriate factors are considered. However, the criteria we tested were not objective enough to produce trustworthy judgments from non-expert raters. Regardless of the criteria, non-expert raters showed a strong bias against Good poems due to these poems' inaccessibility. Even when more accessible poems were used, non-expert raters were unable to clearly distinguish between skillful and unskillful human poems.

### On Novelty, Typicality, Quality and Value

A major difference between Ritchie's (Ritchie 2001) and Pease *et al.*'s (Pease, Winterstein, and Colton 2001) work is the concept of Novelty. While Pease *et al.* define Novelty as a necessary component of creativity, Ritchie prefers to measure its opposite, Typicality. The claim is that, first, a creative computational system must learn to produce acceptable examples of the target output class. For example, a poetry program should not simply produce random words, but should produce something recognizeable as a poem. Only when this hurdle has been crossed can we begin to work towards novel forms of poem.

It is commonly claimed that the novelty and quality of creative works should form a Wundt curve. A completely non-novel work is not interesting. As works begin to diverge meaningfully from other works in their target class, they become more interesting. However, works which are too novel can be off-putting or difficult to accept. At the extreme, a completely novel and chaotic work is indistinguishable from meaningless noise, and is uninteresting for that reason. Therefore, an optimal creative work should involve a moderate amount of novelty. The empirical evidence for such a Wundt curve is not strong (see (Galanter 2012)) but when Ritchie and others treat typicality as a prerequisite to novelty, they implicitly argue for such a curve.

Our research fails to show a Wundt curve or similar relationship between novelty, typicality, and value. Indeed, our research suggests that typicality and novelty are not opposites: the correlation between them is nearly zero ($R = -0.05$). Poems with high Typicality may have high or low Novelty, and vice versa. Typicality is strongly correlated with most of the other criteria tested, with our non-expert raters seeing poems as more valuable, skillful, etc the more typical they are. Even though our data set included very atypical works (non-poems), there did not appear to be a threshold at which poems became "typical enough" for novelty to become relevant.

Meanwhile, Good poems are rated as more novel than Bad. Taken at face value, this would suggest that Novelty might be a better metric than others for measuring creativity. However, the effect for novelty disappears when applied to children's poems. Rather than measuring the creativity of a poem, it is more likely that Novelty for non-expert raters measures inaccessibility: Good poems are rated as more novel than others because they are more difficult to understand. This implies that a participant's rating of a poem as novel may signify discomfort. Without enough domain expertise to see the meaning underlying novelty, non-expert judges prefer poems without it.

### On accessibility and the target audience

If non-expert judges prefer a minimum of novelty, one would expect to see a very different pattern of response from experts. If a poem can be too novel, then this raises the question: too novel *to whom?* Clearly, to the editors of Poetry Magazine, each poem in their magazine made sense and was of high quality. Yet Crowdflower users—presumably ordinary people with little formal education in poetry—saw less quality and sense in these poems than in the work of novice poets.

The poems in Poetry Magazine are so complex that the magazine comes with an explanatory Discussion Guide. Poems allude heavily to other works and imply or illustrate things instead of stating them outright; some raise difficult questions such as "who is creating what, as well as who is inside the work and who is outside" (Poetry Foundation 2014). Without education in poetry, it is no wonder that an ordinary person finds such complexity offputting. Our results suggest that this offputting effect may be so strong that it drowns out any other differences between skilled and unskilled human poetry. To non-expert judges, the confusing complexity of professional poems is worse than any of the clumsiness of an amateur. Yet to an expert in poetry, it would be absurd to say that the amateur poems are therefore of higher quality.

The strength of the effect here—not just negating but reversing expected trends—is surprising. It suggests that there is a great danger in ignoring the question of rater expertise. The use of specific criteria such as Novelty, Value, Skill, Appreciation, or Imagination does not remove the need for this

question. When poems are judged for their quality, who performs that judgment? The researcher? An ordinary reader? An expert? If so, what kind of expert? Future computational creativity studies need to make their answers to these questions explicit, even if they are not already using techniques which demand the use of experts.

In the meantime, without an identifiable target audience, it may be very dangerous to talk about quality, value, or skill in computational creativity as though it is only one thing. The quality of popular appeal and the quality of appeal to experts may be diametrically opposed, and there may be other audiences with still other views of quality. Until such an audience is chosen and the choice justified, the notion of creativity, without the notion of creativity *to whom*, is operationally meaningless.

## Conclusions

Using the conceptual criteria from four popular computational creativity evaluation metrics, we have shown that non-expert humans using these metrics can produce the opposite result from what is intended. Non-expert humans prefer more accessible poetry, even if that poetry is much less skilled according to experts. These results strongly suggest that even when structured metrics are being used, non-expert judges cannot appropriately evaluate the creativity of a human or computer system. Regardless of the metric used, care must be taken in selecting and assessing an appropriate group of judges.

## References


Aguilar, W., and Pérez y Pérez, R. 2014. Criteria for evaluating early creative behavior in computational agents. In *Proceedings of the Fifth International Conference on Computational Creativity*, 284–287.

Amabile, T. 1983. In *The social psychology of creativity*, 37–64. Springer-Verlag New York.

Binsted, K.; Pain, H.; and Ritchie, G. 1997. Children's evaluation of computer-generated punning riddles. *Pragmatics & Cognition* 5(2):305–354.

Boden, M. A. 1990. *The creative mind: Myths and mechanisms*. Psychology Press.

Bown, O. 2014. Empirically grounding the evaluation of creative systems: incorporating interaction design. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Brown, D. 2009. Computational artistic creativity and its evaluation. Number Dagstuhl Seminar 09291, 1–8.

Burns, K. 2006. Atoms of eve: A bayesian basis for esthetic analysis of style in sketching. *AIE EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 20(03):185–199.

Burns, K. 2012. Eve s energy in aesthetic experience: a bayesian basis for haiku humour. *Journal of Mathematics and the Arts* 6(2-3):77–87.

Burns, K. 2015. Computing the creativeness of amusing advertisements: A Bayesian model of Burma-Shave's muse. *AIE EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* 29(01):109–128.

Chan, H., and Ventura, D. 2008. Automatic composition of themed mood pieces. In *Proceedings of the 5th International Joint Workshop on Computational Creativity*, 109–115.

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. 2008a. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.

Colton, S. 2008b. Experiments in constraint-based automated scene generation. *Proceedings of the Fifth International Workshop on Computational Creativity 2008* 127.

Das, A., and Gambäck, B. 2014. Poetic machine: Computational creativity for automatic poetry generation in bengali. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Galanter, P. 2012. Computational aesthetic evaluation: Past and future. In *Computers and Creativity*. Springer. 255–293.

Gervás, P. 2002. Exploring quantitative evaluations of the creativity of automatic poets. In *Proc. of the 2nd Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, the 15th European Conf. on Artificial Intelligence (ECAI 2002)*.

Gervás, P. 2007. On the fly collaborative story-telling: Revising contributions to match a shared partial story line. *International Joint Workshop on Computational Creativity* 13.

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.

Karampiperis, P.; Koukourikos, A.; and Koliopoulou, E. 2014. Towards machines for measuring creativity: The use of computational tools in storytelling activities. In *Proceedings of the 14th International Conference on Advanced Learning Technologies (ICALT)*, 508–512.

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.

Lehman, J., and Stanley, K. O. 2012. Beyond open-endedness: Quantifying impressiveness. In *Artificial Life*, volume 13, 75–82.

Llano, M. T.; Hepworth, R.; Colton, S.; Gow, J.; Charnley, J.; Granroth-Wilding, M.; and Clark, S. 2014. Baseline methods for automated fictional ideation. In *Proceedings of the Fifth International Conference on Computational Creativity*, 211–219.

Monteith, K.; Brown, B.; Ventura, D.; and Martinez, T. 2013. Automatic generation of music for inducing physiological response. In *Annual Meeting of the Cognitive Science Society*, 3098–3103.

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.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *Proceedings of the International Conference on Computational Creativity*, 26–35.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *The Journal of Creative Behavior* 47(2):106–124.

Pearce, M. T., and Wiggins, G. A. 2007. Evaluating cognitive models of musical composition. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 73–80.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In *Workshop on Creative Systems, 4th International Conference on Case Based Reasoning*, 129–137.

Poetry Foundation. 2014. Poetry magazine discussion guide. `http://www.poetryfoundation.org/poetrymagazine/guide/89`. Accessed: 2015-02-03.

Rashel, F., and Manurung, R. 2014. Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry. In *Proceedings of the Fifth International Conference on Computational Creativity*, 82–90.

Riedl, M. O., and Young, R. M. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing* 24(3):303–323.

Ritchie, G.; Munro, R.; Pain, H.; and Binsted, K. 2008. Evaluating humorous properties of texts. In *AISB 2008 Convention Communication, Interaction and Social Intelligence*, volume 1, 17.

Ritchie, G. 2001. Assessing creativity. In *Proc. of AISB01 Symposium*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Román, I. G., and y Pérez, R. P. 2014. Social Mexica: A computer model for social norms in narratives. In *Proceedings of the Fifth International Conference on Computational Creativity*, 192–200.

Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A neologism creator nomen ipsum. In *Proceedings of the Fifth International Conference on Computational Creativity*, 193–181.

Tearse, B.; Mawhorter, M. M. P.; and Wardrip-Fruin, N. 2011. Experimental results from a rational reconstruction of MINSTREL. In *Proceedings of the Second International Conference on Computational Creativity*.

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.

Young, M. W.; Bown, O.; et al. 2010. Clap-along: A negotiation strategy for creative musical interaction with computational systems. In *Proceedings of the International Conference on Computational Creativity 2010*, 215–222.

## Poetry Sources

Bell, J., and Ius, D. Vine Leaves Literary Journal. `http://www.vineleavesliteraryjournal.com/`. [accessed April 2014].

Bobet, L. Ideomancer. `http://www.ideomancer.com/`. [accessed April 2014].

Card, O. S. Strong Verse. `http://www.strongverse.org/`. [accessed April 2014].

Delmater, W. S. Abyss & Apex. `http://www.abyssapexzine.com/`. [accessed April 2014].

el Mohtar, A., and Paxson, C. Goblin Fruit. `http://www.goblinfruit.net`. [accessed April 2014].

ELJ Publications. Amethyst Arsenic. `http://www.amethystarsenic.com/`. [accessed April 2014].

Gage, K., and Filek, M. K. Writing Tomorrow. `http://writingtomorrow.com/`. [accessed April 2014].

Gaskin, E. Astropoetica. `http://www.astropoetica.com/`. [accessed April 2014].

Greene, R. Raleigh Review. `http://www.raleighreview.org/`. [accessed April 2014].

Hart, M. Through the Gate. `http://throughthegate.net/`. [accessed April 2014].

Peg Leg Publishing. GlassFire Magazine. `http://www.peglegpublishing.com/glassfire.htm`. [accessed April 2014].

Poetry Foundation. Poetry Magazine. `http://www.poetryfoundation.org/poetrymagazine/`. [accessed April 2014].

Poetry Free-For-All, T. Newbie Stretching Room. `http://www.everypoet.org/pffa/forumdisplay.php?26-Newbie-Stretching-Room`. [accessed April 2014].

Rademacher, K. Silver Blade. `http://silverblade.silverpen.org/`. [accessed April 2014].

Unknown publisher. Neon - A Literary Magazine. `http://neonmagazine.co.uk/?p=5103`. [accessed April 2014].

Well Done Marketing, Inc. Punchnel's. `http://www.punchnels.com/`. [accessed April 2014].

# Measuring cultural value using social network analysis: a case study on valuing electronic musicians

**Anna Jordanous**
School of Computing
University of Kent, UK
a.k.jordanous@kent.ac.uk

**Daniel Allington**
Department of Arts & Cultural Industries,
University of the West of England, UK
Daniel.Allington@uwe.ac.uk

**Byron Dueck**
Department of Music
Open University, UK
byron.dueck@open.ac.uk

**Abstract**

In evaluating how creative a program or an artefact is, a key factor to consider is the value inherent in that program or artefact. We investigate how to measure subjective, cultural value: value which has been expressed by members of a community towards other members. Specifically we focus on a case study asking: *to what extent can we use social network activity to examine the value that electronic musicians place in each other's work?* Focusing on activity by electronic musicians on the music social network SoundCloud, we combined qualitative and quantitative research to understand and trace significant 'valuing activities' in SoundCloud data. Exploring interaction on the site in this guided way has enabled us to compare, contrast and assess what value is attributed to different members of the electronic music community on SoundCloud. In this paper we report our results and consider how this work offers a methodology for computational analysis of cultural value. We hypothesise that this methodology is extensible to other creative domains; potentially this could lead to a tool for automated cultural value judgement methods on large social network datasets. Hence we move towards computationally generated evaluations of value, a fundamental part of creativity.

**Keywords:** Social network analysis, value metric, evaluation

## Introduction

How can we measure the value of creative entities to a community? (especially unquantified value, expressed through esteem rather than money?) And how could such value judgements be automated across large amounts of data and implemented within computational systems?

Value judgements are a vital part of creativity; the usefulness or value inherent in a creative system and what it does is intricately connected to how creative it is (Ritchie 2007; Jordanous 2012a). In computational research on creativity, we would like our systems to be able to perform evaluation of their own processes. Autonomous judgements of value, integrated within a computational system, are desirable but only occasionally realised in computational creativity.

Value itself can be difficult to identify and measure. In particular, a distinction exists between the more easily identifiable *economic value* of creative works and their producers, compared to their inherent and intangible *cultural value*. Cultural value is attributed through peer interaction and underground expressions of esteem rather than measures such as sales of artefacts or ticket sales. There is a 'relative independence of a status order built from peer esteem from one built purely upon popularity or sales' (Lena and Pachucki 2013, 239). For example, electronic music is a creative domain consisting of many underground subcultures, where economic or popular recognition is often not achieved and quite often not even pursued to any great degree. Value attributions become difficult to recognise due to lack of official recognition or monetary reward for electronic musicians.

So how do you measure or evaluate cultural value? Here we address this question through a case study on electronic musicians. In the Valuing Electronic Music (VEM) project[1] we investigated how electronic musicians show their appreciation and value for other musicians, via qualitative interviews and quantitative research around SoundCloud,[2] a social network for musicians (particularly for electronic musicians). Our aim was to gauge how value is attributed and recognised through interactions between electronic musicians. In particular, we wanted to identify features of inter-artist networking and peer evaluation contributing to value production that are detectable in quantitative analysis of digital interactions. The main aim relevant to computational creativity is to determine what computational analysis could be performed as a proxy for cultural value.

We argue that the approach developed in the VEM project is adaptable to assessment of value in a range of cultural contexts. We offer a method for empirical evaluation of cultural value through analysis of social interactions.

## Value evaluation in computational creativity

Where evaluation of computational creativity systems includes some value judgements, objective metrics have to be carefully selected to ensure value is evaluated in an appropriate and representative manner. In computational work, though, objective metrics have key advantages over subjective data collection, which can be time consuming (especially if collating user feedback) and problematic in terms of identifying representative samples of users. Also, it is difficult to integrate such testing within a computational sys-

[1] See http://valuingelectronicmusic.org

[2] http://www.soundcloud.com

tem's processes and respond to the feedback, particularly if system testing is carried out towards the end of research projects. But there is a need for autonomous value judgements that could be integrated within computational creativity systems; creativity is not just about new work but also the development and refinement of this work (Boden 2004).[3]

The term *value* encompasses many different aspects such as appropriateness, relevance, usefulness, correctness, worthiness and/or quality. A minimum (probably insufficient) definition of creativity could be $novelty + value$ (Jordanous 2012b). Jordanous (2012a) defines value as:

- 'Making a useful contribution that is valued by others and recognised as an influential achievement; perceived as special; "not just something anybody would have done".
- End product is relevant and appropriate to the domain being worked in.' (Jordanous 2012a, 258)

In his discussion of value, Ritchie (2007) makes extensive use of value ratings but leaves open what type of method should be used to generate these ratings. Domain-general heuristics for value judgements are difficult if not impossible to identify; value is relative to the domain and is embodied in different ways. For example, accuracy is vital for mathematical proof generation systems, (Colton 2008) but not for creative musical improvisation (Jordanous and Keller 2012).

One of this paper's authors recently reviewed evaluation of computational creativity systems (Jordanous 2011). She found that 43% of papers containing some content on system evaluation aimed to evaluate the value, quality or appropriateness of the system or system's output. Many types of empirical value measurements were found, as well as value measurements based on user feedback. The value of a creative system entails more than the value of its products; but this perspective was not evident in Jordanous's review. Typically, systems were evaluated based on the value or validity of the artefacts they produce, e.g. statistical tests for validity, calculations of how fit-for-purpose material produced during runtime was, how interesting their products were, or other domain-specific indicators of validity or value.

**Social and cultural value, particularly in music** Blacking noted that the existence of musical geniuses such as Bach and Beethoven is reliant on the presence of a discriminating audience (Blacking 1973). We push this viewpoint further: the relationship between audience and musical performer is both vital for appreciating musical value and the division between audience and musical performer can be blurred. Turino (2008) contrasts 'presentational' musics, based around the quality of works and performances, with 'participatory' musics, where value is within the quality and intensity of social interaction. Turino reminds those in a Western Classical musical mindset of a vital aspect of music: the collective, participatory social aspect of musical experiences, especially when incorporating collective listening, composition, performance and dancing. For example, social interaction and communication are key for creativity in musical improvisation (Jordanous and Keller 2012).

Csikszentmihalyi (1988) proposes a systems model of creativity as a dynamic process of interaction between Domain, Field and Individual/Person:

> '[creative] is the product of ... a set of social institutions, or *field*, that selects from the variations produced by individuals those that are worth preserving; a stable cultural *domain* that will preserve and transmit the selected new ideas or forms to the following generations; and finally the *individual*, who brings about some change in the domain, a change that the field will consider to be creative.' (Csikszentmihalyi 1988, 325).

Csikszentmihalyi's emphasis on interactions between domain, individual and field (Csikszentmihalyi 1988), can be situated within the broader area of *field theory* (Bourdieu 1993) where producers compete for recognition rather than financial gain. Bourdieu posits that all agents involved in music-making form part of the musical communities that attribute value to music-making activities, regardless of level of ability or profile. So 'hidden musicians' (Finnegan 2007) (everyday music-makers who are key to the musical life of communities but understudied by scholars and publics) play a significant part in determining who and what is valuable within musical practices (Dueck 2013). Exploring how hidden and star musicians link together in networks of evaluation and commentary lets us see how all depend upon one another, jointly producing the cultural context in which their music can have value. Although Bourdieu focused on what he termed 'legitimate' culture (i.e. serious literature, art music, etc), his ideas have since been adapted to other cultural forms e.g. Lopes 2000 (jazz), Elafros 2013 (hip-hop).

Social networking and new media websites have provided music makers with new spaces in which to negotiate and produce cultural value for their work, taking on tasks that would once have been the sphere of specialists in marketing, publicity and criticism. These phenomena appear to have had a particular impact on electronic music, which is typically made by lone, but highly networked, individuals and is often circulated non-commercially online. A recent report across UK-based professional musicians found that 64% 'us[e] web-based technologies to produce, promote, and distribute their music' (DHA Communications 2012).[4]

De Nooy argues that social network analysis can legitimately 'be used to gauge the amount of... symbolic capital' (De Nooy 2003, 325). De Nooy's proposed approach to the study of symbolic capital had been successfully implemented as a methodology for studying the production of cultural value by one author of this paper (Allington, under review). Allington used data harvested from online sources to study the production of value within Interactive Fiction (stories that develop in plot through user interaction). Centrality measures were used to assess the level of value associated with specific creators working within that community

[3] Some computational creativity researchers use evaluation in the processes of creative systems (Pérez y Pérez, Aguilar, and Negrete 2010, engagement-reflection), (McCormack 2007, evolutionary computing), (Pease, Guhe, and Smaill 2010, generate-and-test).

[4] This figure may be higher for electronic music, which typically attracts music makers highly familiar with digital technology.

(Allington, under review). Allington's methodology formed the starting point of the present project (complemented with ethnographic research). We scale up from de Nooy's work with tens of producers and Allington's with thousands, to hundreds of thousands of users in the current work.

## Identifying cultural value in electronic music

Looking specifically at electronic music, the Valuing Electronic Music project investigates how we can gauge what cultural value electronic musicians hold. With the above discussions guiding our work, we looked at how peer groups of electronic musicians showed appreciation of each other. Our quantitative work focused on tracing activities for ascribing value to users, through network analysis on large collections of data. This paper reports the project's findings, from the perspective of developing a methodology for empirically identifying and evaluating cultural value (that could in future be incorporated autonomously in a creative system).

Partly inspired by successes using social network analysis to make proxy judgements about value within a network of Interactive Fiction writers (Allington, under review), the research focuses on interactions between creative producers on the music social network SoundCloud, aggregating peer evaluations and tracing the production of value.

**Our approach to cultural value judgements** Our quantitative research centred around collecting and analysing data from SoundCloud's API, about how users interacted with each other on SoundCloud. SoundCloud provides a good data source for technical reasons (a well-developed API provides access to all public data), for social reasons (it is widely used by amateur, semi-professional, and professional electronic musicians for networking and publishing music), and for ethical reasons (the data is clearly marked to site users as public). This sits in contrast to sites such MySpace, which has declined in popularity.

We initially collected data on all demographic information and activities that SoundCloud made public, with the intention of using our qualitative data to understand the relative importance of each activity. Demographic data that users had made publicly available include their location, URLs and avatars relating to their online profile, number of followers, details of record labels they were attached to, etc.[5]The activities that we collected user data for were the publishing of tracks, following and being followed by other users, liking a track, commenting on a track, creating personal playlists of tracks and creating or joining a group.

While the project was primarily a study of online data, this study was contextualised and enriched through study of SoundCloud users in the offline environments in which they primarily perform. In particular, our initial research on SoundCloud suggested there existed more-or-less closely-knit communities of co-located producers of electronic music. This implies that, even in the apparently transnational world of electronic music and online distribution, the social production of value may still be influenced by localised real-time face-to-face interactions. Hence 'offline' qualitative work was conducted alongside our quantitative work, with each mode of research guiding and influencing the other.

We interviewed eight electronic musicians, representing various different types of musicians in different genres from grime to techno. We also attended three electronic music performances and made observations, and interviewed a panel of three musicians at a public event we organised in London in June 2014. Informing our qualitative research, we also actively engaged in the SoundCloud community e.g. 'liking' tracks we enjoyed and following musicians. The interviews helped us to explore the performers' perceptions of value. Using semi-structured interviews allowed us to cover areas of interest such as how the interviewees valued other people's music, while allowing the interviewee to guide the conversation towards areas they felt important. Observation data from gigs (e.g. order of appearance of various performers, prominence of performers' names on promotional materials, audience behaviour, etc) informed the interviews themselves as well as providing much-needed context for our relatively abstract online data.

A common theme emerging from our qualitative research was that rather than searching for *value* (as an entity to measure), we should be focusing on *valuing activities*. Actions by and interactions between musicians were reported by interviewees as a vital way in which they perceived that people appreciated them and their work. Similarly in observations during gigs and in specific questions to live performers, we often noted the importance attached to people's body language and responses to music. In these electronic music communities, the status attached to people also affected to what degree any valuing activities were. In particular, our interviewees typically gave higher credence to interactions with other musicians, compared to those with non-musicians (or those perceived as a non-musician, for example if their reputation as a musician was not known by the interviewee, if they had not mentioned their own musical activities during the interactions or if they had not included pointers to their own work in their SoundCloud profile or other online profiles). This is similar to Bourdieu's emphasis on cultural producers' esteem for one another's work (Bourdieu 1993).

**Data Collection** We wrote code in Python to collect public data automatically from SoundCloud, using the SoundCloud API and Python SDK.[6] It was impractical to study the entire network of users, which comprises tens of millions of accounts, many of them inactive or controlled by bots, and huge amounts of data to collect. We initially adopted a snowball sampling method: starting with a seed individual, collecting data for the seed and the individuals they are connected to, then collecting data for the individuals connected to our seed's connections, and so on). However, we encountered problems with this approach due to SoundCloud network structure and sheer density of data. Many millions of

[5]More details at the SoundCloud API documentation at http://developers.soundcloud.com/docs/api/guide and our github: http://www.github.com/ValuingElectronicMusic/network-analysis.

[6]This code is open-source and available at http://www.github.com/ValuingElectronicMusic/network-analysis - it is built from existing code by Allington for social network analysis, also available from the ValuingElectronicMusic github.

users would frequently be found within just two degrees of separation of a single individual. Undeclared restrictions placed by the SoundCloud API on downloads of information meant that we were prevented from collecting full data on all of those people, with an upper limit of 8199 in place. For example, if a given user had over 100000 followers, one would be unable to discover the identity of more than 8199 of them.

Following discussion with experts at a workshop organised as part of the project, we decided to adopt a different approach. We switched to a two-fold data collection approach of (i) a sample of 150000 randomly selected SoundCloud users and (ii) ego-networks consisting of the networks of users around our interviewees and their followers/followees. In each case, we collected all publicly available data about each user, along with data on all tracks uploaded by these users and those who followed them. Due to the download restrictions of the SoundCloud API we could only download up to 8199 items of data per information request, but in practice this only affected data collection for a very small number of highly popular SoundCloud users. Some minimal data cleaning was needed, mainly for reconciling locations of users where different people used different variations of a location name (e.g. Cairo and Al Qahirah, or NYC and New York), or neighbourhoods within cities rather than cities.

Genres of electronic music are varied and broad, including: house, trance, techno, trap, EDM, ambient, grime, etc. Initial research showed that while the predominant types of music on SoundCloud are in electronic music genres, SoundCloud tracks are often tagged as belonging to a subgenre of electronic music, rather than as 'electronic'. To locate data corresponding to electronic musicians, we could not merely search for those who published music tagged as 'electronic', nor would it be appropriate to treat all electronic music genres as belonging to one community (as confirmed by our interviewees). Instead we made use of the fact that most musicians actively participating on SoundCloud (uploading music, interacting with other users) were electronic musicians. In our data collection, then, we collected data on randomly chosen musicians such that we could later filter the data by genres or other pertinent factors (to be informed by our qualitative research).

**Working out what data to look for** In interviews, we asked if there were valuing activities the participants would highlight as important on SoundCloud, and if so, which ones. In general, even minimal acts of valuing such as playing someone's track were considered to have some value. Participants highlighted indication of a longer term public support base via number of followers, and the use of the commenting facility for people to leave messages on individual uploaded tracks. Further, participants valued activities which arose from or led to offline connections and collaborations, although this type of activity is difficult to track quantitatively.[7] Activities such as playing or 'liking' someone's track or including a track in a personal playlist or group were not highlighted, possibly because it is less easy to trace the provenance of this kind of valuing activity to individual musicians and hence less easy to judge the credibility of the person being interacted with.

The facility to follow and be followed by other SoundCloud users was widely used by users, and afforded analysis or user interaction on a wider scale than at the level of individual comments, allowing us to detect general trends in much larger samples of data. While the follow activity does not require much engagement compared to making a comment on someone's track, nevertheless this activity identifies a SoundCloud user as showing their valuing of another user, in a publicly accessible manner. Qualitatively, we found that there was value attached to having large numbers of followers, though the participants disagreed as to how important this was to them personally. Quantitative analysis revealed, however, that SoundCloud is not a media which compares to YouTube or Twitter in terms of magnitude of followers. In our 150000 user sample, only three accounts had over 100000 followers and all of these accounts represented agents involved in music that had enjoyed significant commercial/popular recognition, above the subcultural recognition that is more common in electronic music scenes.

**Interim results and redirections in our quantitative research** Following Allington (under review), initial quantitative research (Jordanous, Allington, and Dueck 2014) saw us seek the top-ranked users according to centrality measures. (Centrality measures highlight the most influential nodes in a network.) We also attempted to visualise the networks but found that graphs for samples greater than 500 users would be unreadable. We measured recommendation and influence through indegree rankings (a measure based around how many users follow another user). In an initial test sample of 1500 users, we identified key users. This ranking did find some key players in electronic music whose data had been captured in our sample, such as Tiésto. Our results, however, did not help us understand the network at a deeper level, particularly regarding our search for cultural value through peer esteem. While indegree is more sophisticated than merely measuring the number of followers per account, there was some similarity between these two rankings. A 'Justin Timberlake' account, for example, comes in at position 20, despite having no interactive activity on SoundCloud and therefore no identifiable contribution to cultural value through SoundCloud interactions.

We started to explore more sophisticated methods such as PageRank and eigenvector rankings to help identify key players in SoundCloud's networks. However we started to notice a mismatch between qualitative findings and the shape of our quantitative data, stemming from earlier observations about the nature of sub communities within electronic music. In interviews, when we asked questions about valuing and appreciation, participants often replied in terms

[7] Collaborations between two SoundCloud musicians are tricky to detect in SoundCloud data, as tracks on SoundCloud can only be attributed to a single creator. Tracks with two or more associated creators tend to either be uploaded by one of the collaborators with text pointing to the other collaborator(s), or via the creation of a new SoundCloud account representing all the collaborators, which is distinct from the collaborators' personal accounts.

of relationships and interaction. When we probed further, the participants tended to answer in terms of the genre(s) they produced music in, reframing the question to focus on that sub-community they were part of.

Understanding that we should look for subnetworks and cliques within our data, we investigated on what grounds we should cluster our data, through interviews and through inspection of our data for commonly occurring links. Genre was one important clustering factor suggested in the interviews. Somewhat surprisingly for an online network, geographical location was another factor we were guided to investigate. Participants reported how offline interactions at particular places fed back into the social network interactions. The importance of offline contacts could not be ignored, especially given the social network 'fatigue' reported by some participants in building their profiles. In terms of location having an influence on a musician's perceived value, our interviewees talked about the importance of their location for raising their profile and credibility. Though some had experience of being based elsewhere, many of our interviewees were based in London, which - as we find below - is an important centre for electronic music. One participant in particular reported a conscious decision to base themselves in London for profile-raising reasons.

**Analysis of clusters of users and sub-networks** Learning from experience, our quantitative research focused on what sub-communities and clusters existed in our data. We took two directions: 1. constructing and studying multiple networks of electronic music producers and their connections, and 2. using the comments-based data to identify the language used between peers to express value.

We built networks of accounts and tracks, based on 'follow' relationships, which we could re-apply centrality measures to. Clusters and cliques in these networks were also identified where possible, based on available metadata about users and tracks such as genre. We should note here that many users do not provide location information, particularly if not active users (though we focus on those users who actively engage with other users on SoundCloud).

Inspecting the data on comments about tracks, we noted that the overwhelming majority of comments tended to be positive, unlike commenting activity typically observed on sites such as YouTube (Pihlaja 2012). In our analysis of the comments data (filtered from spam where possible) we used the Open Office dictionaries for English, French, Spanish and Italian to identify and extract English language comments to reasonable accuracy.[8] We treated the English-language comments on tracks as corpora based on track genres. Corpus analysis allowed us to identify evaluative vocabularies associated with particular genres, groups, and locations, by comparing these subcorpora on the lexical level. Given that SoundCloud comments were typically positive (or spam), we posit these vocabularies as genre-specific indicators of value as expressed in that genre.

[8]Our approach did not pick up comments such as 'wooooot!!!' or 'loveeeeeeeeee', the type of which occur frequently in our data.

Table 1: Follow relationships by frequency of locations

| | Location of followed | Location of follower | n |
|---|---|---|---|
| 1 | London | London | 3799 |
| 2 | Melbourne | Melbourne | 2274 |
| 3 | Berlin | Berlin | 1375 |
| 4 | Paris | Paris | 1253 |
| 5 | New York | New York | 1190 |

## Computational analysis: Results and discussion[9]

**Geography** Analysis of locations in our random sample revealed London as the most common city location for music makers (users who had uploaded tracks to SoundCloud); 200 accounts out of the 17357 eligible accounts were attached to users based in London. London music-makers had the highest mean number of followers, though a disproportionately high standard deviation reveals results were skewed by a small number of very highly followed accounts.

On analysing individual ego-networks of our participants, we could identify clear clusters within the ego-network based on location of the users, indicating a preference for users to follow other users in the same geographical area as them. This hypothesis was supported by evidence in the larger random sample (see Table 1).

Other key cities identified through our random sample behind London were New York (171 accounts belonging to music-makers), Los Angeles (93), Chicago and Paris (both 81). In terms of followers, strong bidirectional links were identified between London, New York and Los Angeles (UK/US), and then between London, Berlin and Paris (major European capitals). Given that this part of our analysis was genre-agnostic, it was surprising to see cities such as Nashville and Mumbai, with strong musical connections to country music and Bollywood music respectively, featuring little in the interconnected data. Perhaps this is because these types of music do not enjoy the same associations with online/digital technologies and, more specifically, with SoundCloud (emphasising the need to ensure that the social interactions you are analysing are relevant to the creative communities you study).

Using eigenvector centrality based on a graph connected by follow relationships, we identified similar rankings; the central node in this graph was London (0.90093 centrality), followed by New York (0.24838), Berlin (0.20645), Los Angeles (0.20121) and Paris (0.10437). By country, the United States was top by some degree (0.96823 centrality, with the second highest centrality at 0.21216 for the UK). Germany, Canada and France were next in influence, with centrality of 0.07380, 0.05749 and 0.05193 respectively.

**Genre** In raw frequencies, hiphop producers were most prevalent in our sample, with 155 users uploading tracks

[9]The following is a synopsis of findings that are relevant to developing computational analysis of cultural value. Fuller reports of our findings are described in (Allington, Dueck, and Jordanous, submitted) and (Allington, Jordanous, and Dueck 2014).

Table 2: Follow relationships by genre

| | Follower | Following | n |
|---|---|---|---|
| 1 | hiphop | hiphop | 2443 |
| 2 | house | house | 2276 |
| 3 | techno | techno | 1415 |
| 4 | progressive house | house | 800 |
| 5 | dubstep | dubstep | 679 |

tagged as 'hiphop'. House music was second (90 users), followed by rock (61), rap (59) and pop (49). However once we start to study the inherent cultural value through interactions between producers, we see different results as to the influence of different genres. We used eigenvector centrality based on follow relationships to study how producers of music within one genre interacted with music-makers in other genres. In our sample, house music producers were most influential, followed by hiphop, techno, and deephouse. Music tagged as 'electronic' is still prevalent, though its subcategories are widely used as tags instead. These fuller results (Allington, Jordanous, and Dueck 2014, Table 27) also evidenced the influence of electronic musicians (as opposed to musicians of other genres) on SoundCloud.

Many tracks were tagged with more than one genre term, and Figure 1 reveals patterns within genre tagging that empirically support existing genre classifications. Clustering together tags that frequently occurred together on tracks, we identified three macro-genres that could be categorised as 'EDM' (Electronic Dance Music), 'urban', and a miscellaneous 'other' category. The two named macro-genres 'EDM' and 'urban' align with an analysis of data from 2007 on all musical genres on MySpace by Lee & Silver (2014), identifiably corresponding to two clusters that they tagged as 'Electro/Dance' and 'Black & Brown' respectively.

Focusing on activity in the EDM and urban clusters (as the 'Other' cluster contains negligible activity) typically EDM producers follow other EDM producers, and similarly Urban producers follow other Urban producers. Looking at the genre level, a similar pattern of following producers within the same genre is noted (see Table 2).

**Follower activity** A common-sense hypothesis was supported by results: users who uploaded tracks to SoundCloud typically had more followers than those who did not (a mean of 127 followers per account for those who uploaded public tracks, compared to a mean of 19 per all types of users in our 150000-users sample). If we take our qualitative findings that number of followers is generally positively associated with value recognition, then we can underline that music-makers are valued in the SoundCloud community.

**Commenting activity** Taking the three macro-genres we identified, EDM producers were the most prolific commenters with 11711 comments, compared to 3673 comments by urban producers, and 2982 comments by producers of the 'other' genres. By genre, dubstep producers engaged in commenting behaviour the most (2569 comments), then techno (2254), hiphop (2081) and house (1725).

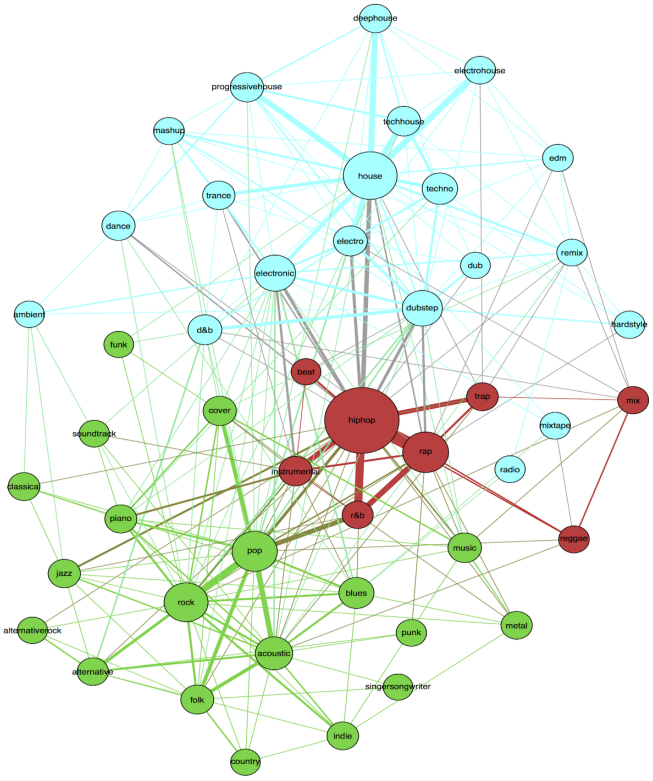



Figure 1: Co-occurrence of genres in track tags

From the comments we have (as described above) identified genre-specific English-language vocabularies indicating value expressions. Keywords are presented for the top genres in Table 3, in order of 'keyless' (decreasing proportional frequency). This table shows the different types of vocabulary prevalent per genre, for example keywords in comments on techno tracks appear more polite than on hiphop tracks.

## Evaluation of our approach

**When is social network analysis appropriate as a proxy for measuring cultural value?** As shown by the lack of useful results of SoundCloud users in cities like Nashville and Mumbai, one needs to ensure they are analysing appropriate social networks for their specific creative domain. There may not be a relevant social network directly for these acoustic-music-based communities, but general social networks such as Twitter may prove useful.

**How could the VEM findings be useful to computational creativity researchers?** Cynically, perhaps, we could set up a London-based SoundCloud account for a hypothetical electronic music computational creativity system we want to promote the work of, ensuring we (or the system) upload(s) tracks produced by our system. We could concentrate efforts on developing our hypothetical system's ability to interact with other music-makers' tracks who work in similar

Table 3: Genre-specific keywords for expressing value

| | Dubstep | Techno | Hiphop | House |
|---|---|---|---|---|
| 1 | sick | set | dope | nice |
| 2 | tune | great | shit | house |
| 3 | nice | tracks | beat | super |
| 4 | big | loved | leave | production |
| 5 | mix | fantastic | song | support |

genres, commenting on such tracks and responding to comments on its own tracks using keywords which have been identified as commonly used in the genre we are working in. We could develop our system to follow other music-makers based in strategically important cities such as London, New York, Los Angeles, Paris or Berlin, or who upload music of similar genres. While this would not necessarily develop the musicality of our hypothetical artificial electronic musician, we argue such moves (if executed plausibly) would help increase the cultural value attributed to our musician (notwithstanding the debate about the effects of identifying the account - or not - as that of an artificial musician (Moffat and Kelly 2006; Cook and Colton 2014)).

**How could social network analysis be used more broadly within computational creativity?** For this work to be most useful to computational creativity researchers, it could a. show how cultural value can be identified and gauged through research and/or b. offer a way of autonomously making value judgements about computational creativity systems. We believe that our work above demonstrates point a., how to tangibly identify markers that indicate cultural value. Allington (under review) has previously used similar network analysis to study Interactive Fiction.

What we pursue now is the afore-mentioned point b., a methodology for using computational network analysis to gauge the cultural value associated with a creative entity such as a computational creativity system. For such an approach to work, we need the system to be capable of interacting with relevant online communities in a plausible manner, as suggested above for our hypothetical electronic computer musician. We also need there to exist an appropriate social network for such interactions to take place in, or as a fascinating alternative, a multi-agent system or similar digital environment containing several interacting agents. For the actual analysis, we advocate using a combination of initial quantitative data analysis and qualitative research to identify key indicators of cultural value that can be traced in the social network interactions. With these conditions in place, we can analyse interactions in the network and compare our creative system or agent to others within the network to gauge the value inherent in its interactive social behaviour.

**Future work** Our results show that electronic music subcultures are geographically influenced and, within the UK, heavily London-centric. Our quantitative methodology could reveal important scenes associated with other cities, and whether we could identify musicians that are considered heavily influential and 'valuable' to the local scene(s). Somewhat inspired by the 2014 Scotland independence referendum, we plan to examine electronic music scenes within Scotland to test our methodology. Our next step will be to apply the same approach to other creative domains to see if social network analysis can be applied more broadly for computationally analysing cultural value. We would welcome collaborations.

Further useful information may be gained from quantitative analysis of comments made by users on each other's tracks, though this was not so straightforward to analyse during the project's funded time. In this work we would have liked to explore and build networks of users based around 'comment' relationships between users. Such work will require considerably more intricate and varied analysis to filter links based around genuine comments. Ongoing work is currently examining the links between users based on commenting behaviours. We would also like to examine conversations; repeated comments or comments on multiple tracks from a user should indicate greater peer engagement. Conversations proved rather difficult to detect quantitatively due to the lack of a standard way to indicate who your comments are directed towards, but their analysis would be useful.

## Conclusions

Value is recognised as a key aspect of creativity. In evaluating computational creativity, one large problem we face is in gauging the value of the work generated by our systems. Such evaluation is particularly problematic when we consider that value is often a cultural and intangible resource apportioned subjectively through the actions of peers.

To what extent can we use social network activity to identify the cultural value of creative entities? Here we addressed this question through a case study investigating how electronic musicians place value in each other's work. The Valuing Electronic Music (VEM) project combined ethnographic observation/interviewing with automated collection of quantitative data from the SoundCloud music networking site. Our approach has implications for how we could measure cultural value in other domains, as well as contributing to our understanding of cultural value in electronic music.

Challenges and rewards alike come from combining situated qualitative research with quantitative analysis of large datasets gathered online. Learning from (and feeding back into) the findings from interviews with electronic musicians, we used computational analysis to study interactions in social networks.[10] Through such analysis we extrapolated information about how musicians interact with each other on SoundCloud, and how they express appreciation of each other's work. Typically, it was more productive to study clusters of strongly connected cliques within the SoundCloud network, rather than a sample of the entire network. The SoundCloud user community tends to cluster according to several factors. We found empirical evidence of clusters forming around common musical genres, and also of clusters around certain privileged geographical locations such

[10]Our approach echoes (Jordanous 2012b): to better represent creative activities using quantitative models, we need good understanding of the creative domain as well as the models themselves.

as London. One key 'take-home' finding from this work is that one can study cultural value computationally by studying social activity, but often it is most useful to study interaction between smaller sub-groups of a network, rather than taking an overall view of the entire network as a whole. In other words, to understand how people express value for each other's work, we should look for social interactions and the building of relationships within a community.

We found that while certain kinds of activity on SoundCloud have little apparent economic value (e.g. commenting on each others' tracks, publishing free downloads) these activities seem to generate cultural value that facilitates more economically valuable work. For the most part, music-makers assert their concern for all listeners, but close attention to their activity (and how they describe it) suggests that interactions with peers (i.e. fellow music makers, preferably within similar genres, areas or with other links) are especially important for the production of value for their work.

Our computational analysis of SoundCloud data allowed us to approximate the value placed in electronic musicians' work, showing that we can use social network analysis as a proxy for measuring certain types of musical and cultural value in a creative domain. We hypothesise that our methodology can be extended to analyse quantitatively the value inherent in other social networks centred around creative activity. We believe that this work contributes towards a significant type of tool in our 'computational creativity toolkit': an automatable method for evaluating social/cultural value.

**Acknowledgments** This work was funded by an AHRC Cultural Value grant "Online networks and the production of value in electronic music". We thank our interviewees, workshop and public engagement event participants and advisory board members.

## References

Allington, D. Producing the Cultural Value of Interactive Fiction: Field Theory, Social Network Analysis, and the Text Adventure Game.

Allington, D.; Dueck, B.; and Jordanous, A. 2015. Networks of Value in Electronic Dance Music: SoundCloud, London, and the Importance of Place.

Allington, D.; Jordanous, A.; and Dueck, B. 2014. Online Networks and the Production of Value in Electronic Music.

Blacking, J. 1973. *How musical is man?* Univ. Washington Press.

Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. London, UK: Routledge, 2nd edition.

Bourdieu, P. 1993. *The Field of Cultural Production: Essays on Art and Literature*. Cambridge, UK: Polity Press.

Colton, S. 2008. Creativity versus the Perception of Creativity in Computational Systems. In *Proceedings of AAAI Symposium on Creative Systems*, 14–20.

Cook, M., and Colton, S. 2014. Ludus Ex Machina: Building A 3D Game Designer That Competes Alongside Humans. In *Proceedings of 5th International Conference on Computational Creativity*.

Csikszentmihalyi, M. 1988. Society, culture, and person: a systems view of creativity. In Sternberg, R. J., ed., *The Nature of Creativity*. Cambridge, UK: Cambridge University Press. Ch. 13, 325–339.

De Nooy, W. 2003. Fields and Networks: Correspondence Analysis and Social Network Analysis in the Framework of Field Theory. *Poetics* 31(5-6):305–327.

DHA Communications. 2012. The Working Musician. *Report for the Musicians' Union*.

Dueck, B. 2013. Jazz Endings, Aesthetic Discourse, and Musical Publics. *Black Music Research Journal* 33(1):91–115.

Elafros, A. 2013. Greek Hip Hop: Local and Translocal Authentication in the Restricted Field of Production. *Poetics* 41(1):75–95.

Finnegan, R. 2007. *The Hidden Musicians: Music-Making in an English Town*. Middletown, CT: Wesleyan Univ. Press, 2nd ed.

Jordanous, A.; Allington, D.; and Dueck, B. 2014. Using online networks to analyse the value of electronic music. In *Proceedings of 5th International Conference on Computational Creativity*.

Jordanous, A., and Keller, B. 2012. What makes musical improvisation creative? *Journal of Interdisciplinary Music Studies* 6(2):151–175.

Jordanous, A. 2011. Evaluating Evaluation: Assessing Progress in Computational Creativity Research. In *Proceedings of the Second International Conference on Computational Creativity (ICCC-11)*.

Jordanous, A. 2012a. 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. 2012b. *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application*. Ph.D. Dissertation, University of Sussex.

Lee, M., and Silver, D. 2014. Testing the Springsteen Conjecture: "Exploring the Post-Authentic Musical World" with big, messy internet data. http://badhessian.org/2014/07/testing-the-springsteen-conjecture-exploring-the-post-authentic-musical-world-with-big-messy-internet-data/ last accessed Feb 2015.

Lena, J. C., and Pachucki, M. C. 2013. The Sincerest Form of Flattery: Innovation, Repetition, and Status in an Art Movement. *Poetics* 41(3):236–264.

Lopes, P. 2000. Pierre Bourdieu's Fields of Cultural Production: a Case Study of Modern Jazz. In Brown, N., and Szeman, I., eds., *Pierre Bourdieu: Fieldwork in Culture*. New York: Rowman and Littlefield. 165–185.

McCormack, J. 2007. Creative Ecosystems. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 129–136.

Moffat, D. C., and Kelly, M. 2006. An investigation into people's bias against computational creativity in music composition. In *Proceedings of the 3rd International Joint Workshop on Computational Creativity ECAI06 Workshop*.

Pease, A.; Guhe, M.; and Smaill, A. 2010. Some Aspects of Analogical Reasoning in Mathematical Creativity. In *Proceedings of the International Conference on Computational Creativity*, 60–64.

Pérez y Pérez, R.; Aguilar, A.; and Negrete, S. 2010. The ERI-Designer: A Computer Model for the Arrangement of Furniture. *Minds and Machines* 20(4):533–564.

Pihlaja, S. 2012. *The development of 'drama' in YouTube discource*. Ph.D. Dissertation, Open University.

Ritchie, G. 2007. Some Empirical Criteria for Attributing Creativity to a Computer Program. *Minds and Machines* 17:67–99.

Turino, T. 2008. *Music as Social Life: The Politics of Participation*. Chicago: University of Chicago Press.

# Conceptualizing Creativity: From Distributional Semantics to Conceptual Spaces

**Kat Agres, Stephen McGregor, Matthew Purver, and Geraint Wiggins**
School of Electronic Engineering and Computer Science
Queen Mary University of London
London E1 4NS UK
kathleen.agres, s.e.mcgregor, m.purver, geraint.wiggins (@qmul.ac.uk)

**Abstract**

This paper puts forth a method for discovering computationally-derived conceptual spaces that reflect human conceptualization of musical and poetic creativity. We describe a lexical space that is defined through co-occurrence statistics, and compare the dimensions of this space with human responses on a word association task. Participants' responses serve as external validation of our computational findings, and frequent terms are also used as input dimensions for creating mappings from the linguistic to the conceptual domain. This novel method finds low-dimensional subspaces that represent particular conceptual regions within a vector space model of distributional semantics. Word-vectors from these discovered conceptual spaces are considered, and argued to be useful for the evaluation of creativity and creative artifacts within computational creativity.

## Introduction

This paper presents a computational-linguistic model for mapping lexical spaces populated by statistical representations of words to conceptual spaces defined in terms of feature dimensions of conceptual representations. This research has three main goals. The first is to compare the features of a distributional semantic vector space with the results of an empirical word-association task completed by human subjects. This empirical corroboration serves to demonstrate that the model can capture meaningful aspects of human conceptualizations of queried topics, which are "musical creativity" and "poetic creativity" in the present study. The second goal is to use novel methods inspired by computational linguistics to map terms from the linguistic domain to representations in the conceptual domain. To this end, the terms generated by participants are used as input parameters for our computational model that uses co-occurrence statistics and linear algebraic metrics to quantify conceptual proximity. The third motivation of this work concerns the evaluation of creativity. In the field of computational creativity (CC), the evaluation of creative output is often either subjective on the part of the developer/researcher or unsystematic. We offer our own fundamentally computational approach as a means of identifying facets of the investigated concept or domain. Put another way, our model can generate terms within a conceptual space that may be used to query different aspects of creative output or creative behavior.

Vector space models of distributional semantics are currently a popular approach for quantifying linguistic similarity, but many contemporary studies need grounding and external validation. Much of the work in this area compares model performance to semantic databases, but does not directly relate results to the cognitive performance of humans, or uses very restricted tasks, such as similarity judgments, rather than imploring subjects to elaborate on *concepts*. Because our aim is to elucidate how humans conceptualize creativity, sampling from people's own formulation of conceptual spaces is essential. Therefore, in the present work, our ground truth is derived from human responses stemming from direct queries about creative concepts. Because human response data is a limited and expensive resource, we hope that our comparison to human data will inform how conceptual spaces may be discovered as autonomously as possible in the future (that is, without the requirement of subjective user-input or parameter-tweaking). We also believe that this multidisciplinary and externally validated approach produces a more robust system.

In order to pinpoint the relationship between the output of our computational model and the results of our empirical study, we take the human-generated terms and investigate their situation within the multidimensional space of our distributional semantics model. We then determine the characteristic co-occurrence dimensions of sets of words associated with concepts, and apply appropriate methods to reduce the dimensionality of the space in order to map broader clusters of linguistic terms to conceptual regions. We argue that the online generation of a reduced lexical space corresponds to the contextualization inherent in the momentary way in which concepts are necessarily formed in response to situations in a cognitive environment. We expect that this methodology will be a useful applied approach to formalizing the geometrical representation of conceptual spaces.

Our research explores two related concepts: musical creativity and poetic creativity. There are several reasons for this choice. First, we are interested in computational creativity, and in particular in the evaluation of creative systems and their output. In order to evaluate creativity, it is necessary to characterize features of this concept using the expressive affordances of language. Our computational methods seek to capture these features of the conceptual space. Our model may also be used to discover conceptually-related terms that a human might not necessarily immediately con-

sider. We hope this approach may be used to elaborate abstract concepts by elucidating an extensive set of terms that correspond to the queried conceptual spaces. We therefore offer this methodology as a novel approach for exploring and elaborating concepts, both for the evaluation of creative systems and for potentially contributing to creative pursuits themselves (such as poetry generation). Furthermore, we apply our method to a more concrete domain, extending a small subset of terms relating to the concept WILD ANIMALS in order to indicate the anticipated generality of our model.

The organization of the paper is as follows: first we offer a summary of computational approaches to conceptual creativity, situating our research within the field. This overview leads into a discussion of computational approaches to the topics of conceptual spaces and geometric representations of concepts. An explanation of our computational model is then provided, including a description of how we have modeled a lexical space populated by word-vectors. This is followed by a description of our empirical study with human participants, and findings from this questionnaire-based study are reported. Given this context, we then discuss two ways in which the participants' responses contribute to our computational approach. The first is a comparison of computationally-derived terms with human-generated terms. The second contribution will be to treat the salient features of the word-vectors corresponding to the most frequently reported human terms as an indication of the dimensions of a vastly reduced subspace of our distributional semantic model. We then discuss how terms that fall near the centroid of the positively valued surface of the discovered lexical spaces may be used for the evaluation of creativity.

## CC and Concept Discovery

Computational approaches to creative conceptualization have provided a target that is both elusive and essential to the identity of a field that incorporates a particularly diverse range of topics. Creativity itself has been interpreted by Koestler (1964) as a kind of meshing of disparate conceptual schemes, by which expectations are violated in favor of interesting new combinations of frames of reference. Presciently, Koestler has couched his model of creativity in terms of vector spaces and transformations, an idea which is broadly shared by the model presented in the present paper. In the same spirit of conceptual exploration, Hesse (1963) argued that the formation of creative analogies is the essence of scientific discovery, an idea demonstrated by the primacy of analogical modeling in fields such as physics, where there is no realistic way to literally conceive of phenomena that occur on obscurely minuscule or vast scales.

In the specifically computational domain, Veale (2006) has proposed a system for the dynamic generation of new, non-literal conceptual categories based on a computational analysis of a taxonomical database such as WordNet. Likewise, other researchers are developing formal models of conceptual blending (Fauconnier and Turner, 2008) that seek to discover novel combinations of familiar ideas, targeting domains such as mathematical reasoning and story generation (Ontanón, Zhu, and Plaza, 2012). These approaches make clever and effective use of heuristics to pick out interesting new conceptual representations based on preconceived patterns identified by programmers. As such, the output of these methods is compelling and valid, but the conceptualization itself is arguably handed to the system in the prepackaged form of externally grounded symbols.

Elsewhere, Heath et al. (2013) have taken a more connectionist approach to conceptual creativity, combining human based word associations with statistical models of distributional semantics to design a system that infers conceptual categories from lists of terms, and likewise generates lists of terms from linguistic input that is interpreted conceptually. In a similar vein, Jäger (2009) has performed a statistical analysis on a set of human reported color terms and used this analysis to generate a geometric representation of certain consistencies in the ways that color is conceptualized across cultural linguistic boundaries. In their commitment to building models based on low level, non-symbolic observations about the world, these statistical approaches to creative conceptualization are in the same spirit as the work presented in the present paper.

The model described here has been designed to engage with the field of computational creativity on two different planes. Principally, our method seeks to implement a low level approach to the delineation of conceptual regions based on the geometry of a distributed semantic space. By viewing concepts as momentary and pragmatic phenomena, we are able to use *ad hoc* reductions of a high dimensional lexical space to map concepts creatively based on situational contexts which do not have to be preformulated in the design of the model. Furthermore, our target domains of musical and poetic creativity play nicely into a salient issue in the field of computational creativity: the analysis of creativity itself, a difficult procedure that necessarily involves some degree of conceptualization about creativity. This secondary aspect of the work, the potential for meta-analysis inherent in the question of whether our model's output will be useful for guiding an evaluative discussion of creative work elsewhere, is intended to give the work its own pragmatic grounding, in that this suggests a practical application for the creative output described in the following pages.

## Spaces of Meanings

This project uses computational methods as a platform for exploring the relationships between words and concepts within the context of a cognitive system. In the pragmatic spirit of Wittgenstein (1953) and Grice (1969), language is presented as a system defined by its own functionality, with meaning emerging from the use of words in the course of accomplishing communicative goals. To the extent that language is used to communicate ideas, statements are formed contextually, with reference to expectations about how relationships between words will suggest hierarchies of categorization relative to a particular situation. Barsalou (1993) characterizes the relationship between words and concepts in terms of the *linguistic vagary* inherent in the application of names to ideas: words represent concepts in a way that is fleeting and mutable. Fundamentally, words stand as indices to concepts, and the relationship between language and ideas is best understood as a mapping between two separate domains. The project presented in this paper is therefore motivated by a desire to model the relationship between two

different spaces, one of words and one of concepts, and to explore the ways in which these spaces might be aligned in terms of the computationally tractable elements of their geometries.

Gärdenfors (2000) has presented a spatial theory of concepts, by which the dimensions that determine the geometric situation of a conceptual region within a space of concepts correspond to the attributes which characterize that particular region. So, for instance, the concept RIPE BANANAS would occupy a region towards the higher end of the dimensions of curviness, yellowness, and sweetness within a conceptual space. This literal and factual quality of dimensions grounds conceptual spaces in low level observations about the world, giving regions within the space a geometric dynamism that lends itself to doing higher level work with the entities that emerge from the space as symbolic representations. In particular, well defined conceptual regions are characterized by convexity, a property that ensures that any intermediate point between two outlying extensions of a region will likewise belong to that domain.

Vector space models of distributional semantics, on the other hand, offer an approach to language modeling involving a distinctly unstructured computational analysis of linguistic data. In the tradition of Harris (1957), the distributional hypothesis holds that there is semantic information inherent in the statistical comportment of language: linguistic meaning can be found in the quantifiable contextual relationships between words. This insight has motivated a productive field of research, with computational analyses of large scale corpora yielding distributional semantic models in which the meanings of words, sentences, and documents are rendered in terms of mathematically tractable representations (Schütze, 1992; Landauer et al., 1997). Distributional semantic models treat words as vectors, with the dimensions of these vectors representing, either directly or abstractly, the likelihood of a word occurring in a given context. The closeness of vectors in a lexical space, which reflects the tendency of the proximal vectors to occur in similar contexts, has been shown as an indication of lexical similarity between the words tied to the vectors. In their most straightforward implementation, distributed semantic spaces are constructed by counting the frequency with which each word in the model co-occurs with all other terms in a base corpus (see Turney and Patel, 2010; Clark, 2015, for an overview).

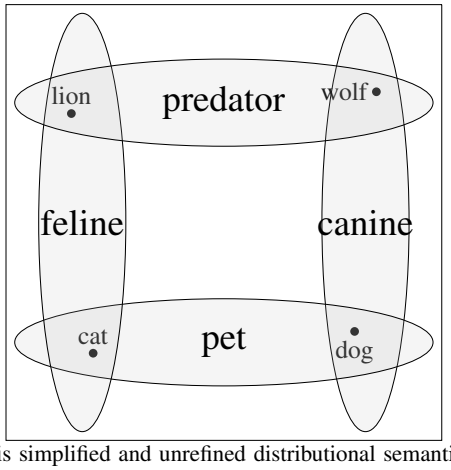
There is an important difference between lexical spaces and conceptual spaces: the dimensionally regimented quality of coherent domains within a conceptual space is not reflected in the distribution of vectors in a lexical space, where the dimensions of word-vectors correspond simply to the context in which those words are likely to occur, and therefore capture all the flexibility and uncertainty of language in use—the *linguistic vagary* of Barsalou's system of conceptual symbols. So, for instance, the other vectors in the proximity of the word-vector $\overrightarrow{pet}$ in a distributed semantic model cannot be expected to contain only terms corresponding to domesticated animals, not least because the word "pet" itself has other uses. In this sense, where conceptual spaces are marked by a tidy taxonomy facilitated by the clarity of a region's dimensional substrate, distributed semantic spaces embody the pragmatic messiness of language as it is encountered in its natural, operational environment. Therefore, while lexical spaces and conceptual spaces both utilize geometry as a vehicle for semantics, the arrangement of a lexical space is in an essential way less ordered.

An example of the difficulty of delineating conceptual regions within a lexical space is illustrated in Figure 1a. In the rudimentary distribution of words presented here, concepts are required to stretch and overlap in order to maintain their lexical constituencies. This simplified depiction of the potential uncertainty of conceptual membership doesn't demonstrate the even more fundamental problem of picking out salient words in regions that are littered with noise: in practice, in the densely and unevenly populated territory mapped out by a vector space model, many unwanted terms will be discovered in the region generally between two other terms. For instance, in the unrefined version of our model, there are 14 terms essentially between the word-vectors $\overrightarrow{cat}$ and $\overrightarrow{dog}$, including such unlikely candidates as "during", "eventually", and "featuring". There is thus an inherent patchiness to the mapping of concepts that might be read in an unrefined vector space model of distributional semantics.

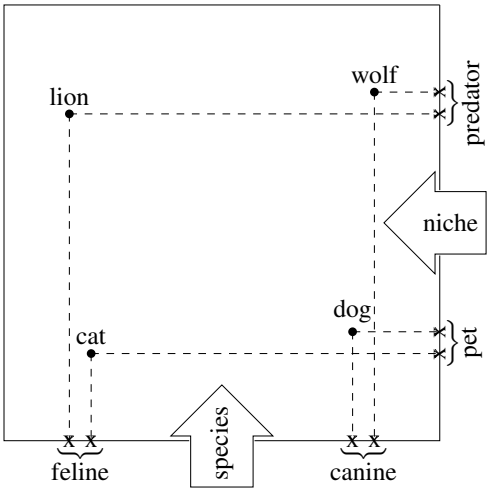
Here we propose a system for mapping lexical spaces to conceptual spaces by considering a conceptualization as a particular and temporary perspective on a space of distributed semantics. The idea behind this system is that, for any desired clustering of words corresponding to a particular conceptualization, there is some subset of a distributional space's dimensions that will render a subspace in which that clustering is realized. This intuition is illustrated in Figure 1b, where the conceptually entangled space of Figure 1a collapses into a particular conceptual regime depending on the axis along which the space is projected, which is to say, the perspective from which the space is considered. The task of our system is therefore to determine the dimensions which should be picked out of a higher order vector space model in order to realize a grouping of terms that is conceptually homogeneous by virtue of the contextualization imposed by a particular perspective on the space.

It is precisely the massive dimensionality of the space which facilitates the method's ability to pick out various successful conceptual perspectives on the space in a momentary and continuous way. With each additional contextual dimension introduced to a vector space, there is an exponential increase in the lower-dimensional combinations available to map corresponding spatial relationships of words to conceptual subspaces. Moving from the linguistic realm native to vector spaces back to the cognitive domain targeted by Gärdenfors, these dimensional perspectives might be construed as corresponding to a contextualized perception of a situation. In this respect, our system models the *haphazard* quality of conceptualization described by Barsalou, as well the *ad hoc* nature of concept formation discussed more recently by Allott and Textor (2012), who suggest that meaning is appropriated *in situ* to endow statements with contextually relevant implicature. This phenomenon of conceptualization arising from pragmatic communicative affordances is what our method seeks to computationally model.

Figure 1: Conceptual Perspectives of Vector Spaces




(a) In this simplified and unrefined distributional semantic space, the conceptual regions suggested by the spatial arrangement of terms are indeterminate. Each word is roughly equidistant from two other terms, either of which could be linked in a distinct linguistic depiction of a concept. The conceptual domains which are delineated by this arrangement of words are awkwardly elongated.




(b) If the same space illustrated above is considered from two different perspectives, the indeterminate arrangements of words collapse into lower dimensional spaces (one dimensional, in this simple example) in which the clustering of terms suggests straightforward conceptual domains. These perspectives effectively contextualize the meanings inherent in the distributional characteristics of the language model, and this context facilitates the mapping of the linguistic space to sets of conceptual regions.

## A Literal Lexical Space

Our lexical model has been constructed based on the distribution of words found in the textual component of articles on the English language Wikipedia website.[1] The xml code of the site was downloaded and then parsed into a text-only format, eliminating images, tables, lists, captions, and section titles, leaving only the well formed sentences composing the content of the site's articles. Sentences were separated by identifying terminal punctuation followed by whitespace, then punctuation was removed and all characters were converted to lower case. Articles ("a", "an", and "the") were stripped from the text. Sentences containing less than five words were discarded. The resulting corpus consists of almost 60 million sentences, containing about 1.1 billion word tokens (individual words) corresponding to about 7.4 million word types (classes of words).

From this base corpus, we took the 200,000 most frequent word types to form our system's vocabulary. Our full lexical space is represented as a matrix $M_{w,c}$, where rows correspond to vectors representing words, and columns correspond to co-occurrence terms. The cell for word $w$ and co-occurrence term $c$ contains the *mutual information* $MI_{w,c}$ as described in Equation 1. Here $n_{w,c}$ is the frequency with which a term $c$ is observed to co-occur within a context window of two words on either side of a vocabulary word $w$; $n_w$ represents the total count of $w$ in the corpus; $n_c$ is the total count of $c$; and $a$ is a smoothing constant.

$$MI_{w,c} = \log_2 \left( \frac{n_{w,c} \times N}{n_w \times (n_c + a)} + 1 \right) \quad (1)$$

The constant $a$ reduces the undesirable effect of contextual words that occur very rarely throughout the corpus, but with a high frequency in the context of certain target words—we found 10,000 to be a good value for $a$ based on trial and error. The value 1 is added to the probabilistic ratio in order to render all dimensions within the space positive: this means that the logged $MI$ value of target words and context words that never occur together will be 0, and the value for terms that co-occur less frequently than would be expected in a random distribution will be between 0 and 1. Each word vector consists of a set of dimensions derived through this calculation, and each of these vectors is normalised to the scale of a unit vector. The result of this process is a distributional semantic space in which each of the 200,000 vocabulary terms sits in the positive region of the high-dimensional surface of a hypersphere.

One notable feature of our vector space is the literal correspondence of its dimensions to co-occurrence terms. In general, state-of-the-art systems apply some form of dimensional reduction to the overall space, either using linear algebraic transformations to perform a principal component analysis (Pennington, Socher, and Manning, 2014), or by weighted networks to train abstract lower-dimensional word representations that predict the context in which that word is encountered in the course of training (Mikolov et al., 2013). While these techniques certainly make the space less expensive to compute, and arguably improve results for a variety of semantic tasks, our system is specifically geared towards the identification of salient, literal co-occurrence dimensions, and as such the space is, for the purposes of our initial analysis, maintained in its raw high-dimensional form. The dimensionality of our space is therefore on the

[1] The December 8, 2014 dump, downloaded from http://meta.wikimedia.org/wiki/Data_dump_torrents on January 23, 2015, parsed into plain text using the "Wikipedia Extractor" software, downloaded on February 13, 2015 from http://medialab.di.unipi.it/wiki/Wikipedia_Extractor.

order of 7.4 million, as every word token in the corpus is a potential context for the 200,000 words in our vocabulary.

## Empirical Validation

In order to provide grounding and validation for our computational model, human participants were asked to generate terms during a word association task, and to reflect upon how they would evaluate creativity in the musical and poetic domains. Provided terms were analyzed for comparison with the vector space model.

### Method

**Participants** Twenty participants (avg age = 30 yrs, stdev = 5.2 yrs) volunteered to take part in the study, of which 11 were female. Sixteen individuals indicated that their career is either inherently creative or that they apply creativity to improve their job performance, and all but two of the participants engage in creative pursuits outside of work. Seven currently practice or perform music, and five individuals currently engage in creative writing.

**Procedure** After reading an information form and providing informed consent, participants were given a brief questionnaire to complete. Two of the questions consisted of a word association task in which participants were asked to list three terms they associate with "musical creativity" or "poetic creativity". The order in which musical or poetic terms were prompted was counterbalanced across subjects. The other two questions requested participants to write one sentence describing how they would evaluate whether a new piece of music or poetry sounds creative (the order of these questions were similarly counterbalanced across subjects). These results will only briefly be touched upon in the current paper; although certainly of interest, due to space constraints, an in-depth analysis of the evaluation sentences must be saved for an expanded version of this work.

After providing their responses, participants were given questionnaires requesting general demographic information (age, ethnicity, etc), and information about their past and current involvement in creative pursuits, e.g., "Do you currently play music or engage in creative writing?" Upon completing these, participants were debriefed as to the goals of the experiment and paid £2 each for their participation.

### Results

For both musical and poetic creativity, participants' terms were placed into two lists: An exhaustive list of all terms provided for the concept, and a short list for terms cited by more than two participants (per concept). In the case of musical creativity, this yielded an exhaustive list of 52 distinct terms, and for poetic creativity, a set of 42 terms. The short list of musical terms included the following six terms: *innovation, sound, instruments, novelty, emotion, and expression*. The short list of poetic terms included these six terms: *emotion, rhythm, expression, structure, flow, and words*. We interpreted these concise lists of most frequent words to reflect dimensions of the concept that are more central to the conceptual space they populate. This resulted in discarding more peripheral terms such as "sensitive" that are undoubtedly related to creativity, but not cited frequently as an associated concept. Plurals and conjugations were considered to be the same category of term, e.g., "emotions", "emotional", and "emotion" were tallied together as "emotion". We continue the discussion of empirical findings in the next section, as we compare the model's performance with human results.

## Mapping Words to Concepts

We began the exploration of mappings from our space of distributed semantics to a conceptual space with a top down approach, investigating the way our system reacted to the same kind of input that we presented to our human subjects. Along these lines, we examined the vectors for the word pairing "musical" and "creativity", and likewise for the pairing "poetical" and "creativity". In each instance, we calculated the mean value for each dimension that had a non-zero value for both words – that is, for each dimension corresponding to a term that co-occurred with both words at least once in the corpus – and returned a ranked list of average scores, running from high to low. Out of the 7.5 million co-occurrence features across the entire model, 4,772 were non-zero for both "musical" and "creativity", and 2,673 for both "poetic" and "creativity", statistics which highlight the sparsity of the base space. Our objective was to examine the nature of the terms that tended to come up in the context of our query as phrased for our human subjects. Results are listed in Table 1.

The first thing to note about these results is that they are, in a qualitative sense, coherent descriptions of properties typically associated with the two concepts being explored. To frame this more empirically, these results can be extended in order to discover how far down the list of top mean dimensions the terms reported by humans lie. Of the exhaustive list of terms reported by human subjects in response to the "musical creativity" query, 4 fall within the top 15 results generated by our model; likewise, 4 human responses fall within the top 15 mean dimensions for "poetic creativity" (these terms are italicised in Table 1). Considering that 200,000 words were used as the vocabulary of the model, yielding 4 of the top 15 dimensions in common with humans' responses for both concepts is quite compelling. This outcome may be interpreted as indicating that there is a high degree of mutual information between the query words and terms that humans would consider as conceptually descriptive of those queries. In other words, there is a high likelihood of conceptually relevant co-occurrence within the context of terms that summarize these creative conceptual domains.

These positive results do not hold up, however, for more concrete queries. For instance, when the mean dimensions for the query pair "wild" and "animal" are explored, top ranking results include some conceptually appropriate terms such as "boars", "deer", and "feral", but less directly relevant words like "skins" and "vegetable", and even antonymic terms like "domesticated" are also returned. It would seem that, in the case of words indexing more concrete concepts, the likelihood of co-occurrence in the conceptual context moves away from terms that generically describe components of the concept in question. This distinction is corroborated by Hill, Korhonen, and Bentz (2014),

| "musical" & "creativity" | "poetic" & "creativity" |
|---|---|
| *innovation* | genius |
| imagination | *imaginative* |
| inventiveness | *metaphors* |
| *improvisation* | originality |
| talent | prose |
| talents | creativity |
| *experimentation* | artistry |
| versatility | craftsmanship |
| artistic | intuition |
| creativity | *imagery* |
| ingenuity | inspiration |
| aesthetics | talents |
| *spontaneity* | lyrical |
| individuality | talent |
| artistry | *self-expression* |

Table 1: The top 15 dimensions with the highest mean scores between the word-vectors for each of our queries as given to human subjects. Terms in italics denote dimensions that were also cited by humans.

who have used computational analyses of both corpora and semantic graphs to illustrate a distinction between the way that abstract and concrete concepts are arranged in a cognitive linguistic system. It is hardly surprising, given the inherent ambiguity of language use – replete, as it is, with metaphor and implication – that simple co-occurrence probability statistics do not generally map neatly on to well defined conceptual spaces.

## Projecting Words to Conceptual Subspaces

Motivated by this predictable shortcoming of a simple dimensional analysis, we developed a more sophisticated approach for delineating conceptual regions within dimensionally reduced subspaces of our language model. Our technique involves first hand-picking a small set of terms that might be considered as paradigmatic descriptions of components of a conceptual domain (in the present example, WILD ANIMALS). We perform an analysis similar to the one described above on these conceptual component terms, selecting the word-vector for each term and then extracting those features with non-zero values for all input terms. Once again, we compute a ranked list of these mean feature values and choose the co-occurrence dimensions which scored highest on average. These salient dimensions for the small set of words analyzed are again somewhat scattered: some of the highest mean dimensions correspond to relevant animal names, but the results also stray into the more conceptually ambiguous territory signified by words like "sightings", "chases", and "fat". There are 827 universally non-zero dimensions found between the word-vectors of the six input terms describing exemplars of WILD ANIMALS listed in the first column of Table 2.

We use these salient dimensions to define a drastically simplified subspace of our lexical model. Specifically, we reduce the model to the top 30 dimensions associated with the set of sample words (we arrived at the number 30 through trial and error; lower values tended to invite some unusual vectors into the crucial region of the subspace). After normalizing the new subspace, we then identify the central point on the surface of the positively valued quadrant of the reduced hypersphere—effectively the vector defined by 30 dimensions each with the the value $1/\sqrt{30}$. This positive centroid is then taken as the epicenter of a linguistic mapping of a new conceptual region, and we expand the region outward concentrically from this point, returning an ordered list of the points closest to the center of the positive surface of our space's low dimensional projection. Euclidean proximity is calculated by computing the square root of the sum of the squared feature-wise differences between the unit centroid and each of the 200,000 vocabulary words projected into the subspace. The top fifteen terms encountered using this method are reported in Table 2. Please note that the input terms for WILD ANIMALS are used as a preliminary test of the model's performance for concrete concepts; the input was hand-selected by the investigators, while the collection of terms relating to concrete concepts is the subject of ongoing emprical study.

This same technique for expanding conceptual regions through a dimensional reduction of a distributional language model is applied to our target domains of musical and poetic creativity, again with compelling outcomes. In this case we were able to make use of our results from our survey: for each of our two target domains, we choose all the terms that were reported by three or more human subjects and analyze these for their most salient dimensions of co-occurrence. Again, the system's output for these terms is not entirely unexpected, but also not conceptually completely cohesive. In the case of the highest mean dimensions for the human reported constituents of MUSICAL CREATIVITY, a number of predictable terms are returned, but somewhat less obvious dimensions such as "lab", "mere", and "shapes" also rank towards the top of the list.

Despite the conceptual uncertainty in the dimensional analysis, when a new subspace is constructed based on these dimensions, the central region of this space is replete with terminology appropriate to the example words at the base of the process. Interestingly, the original input words are only partially rediscovered in this new space, at least within the set of vectors most central to the positive surface of the new subspace. This indicates that some of the input word-vectors (used to select dimensions for creating the new subspace) are, in terms of the probability of regular co-occurrence with all the dimensions that underwrite the subspace, relative outliers which nonetheless make an essential contribution to the delineation of this linguistic representation of a conceptual region. It is also notable, and perhaps even remarkable, that in the case of the mapping of the conceptual region of poetic creativity, quintessential new terms such as "phrasing" and "inflection", arguably more intricately associated with the prosodic nature of the target domain than the original human generated terms, arise independently.

When examining how well the model captures relevant terms to a conceptual query, it is informative to cluster human responses into semantic categories (such as *emotion* and *structural elements*), and compare these results to apparent categories of output vectors. For example, considering the sentences that the 20 participants wrote about how

| WILD ANIMALS | | MUSICAL CREATIVITY | | POETIC CREATIVITY | |
|---|---|---|---|---|---|
| human input | model output | human input | model output | human input | model output |
| lion | bobcat | innovation | novelty | emotion | phrasing |
| wolf | alligator | sound | liveliness | rhythm | intonation |
| coyote | raccoon | instruments | spontaneity | expression | musicality |
| alligator | opossum | novelty | innovation | structure | nuances |
| bear | armadillo | emotion | expressiveness | flow | timbre |
| snake | white-tailed | expression | refinement | words | sprightly |
| | anteater | | nuance | | rhythmical |
| | ocelot | | ingenuity | | nuance |
| | peccary | | believability | | expressiveness |
| | pronghorn | | newness | | rubato |
| | cougar | | sophistication | | instinctive |
| | cottontail | | dynamism | | bluesy |
| | rattlesnake | | subtlety | | directness |
| | skunk | | vibrancy | | modal |
| | boar | | elusiveness | | inflections |

Table 2: The output vectors most central to the positive regions of the subspaces reduced in terms of the salient dimensions of a small set of conceptually exemplary input terms.

they would evaluate the creativity of a new song, many individuals referred to the notion of *novelty* in musical creativity, but used various terms to do so. In addition to explicitly using the term "novelty", participants made reference to "unexpected", "new", and "surprising elements" that were "like nothing else I'd heard before", as well as "melodic originality" and cases in which "known musical concepts or styles [are] combined in a novel/innovative way." Similarly, when considering the model's conceptual space of musical creativity, the words "novelty", "innovation", "ingenuity", "newness", "inventiveness", "distinctiveness", and "uniqueness" are found within the top 30 model output vectors. Although this is a qualitative assessment of the results, it does seem clear that the precise terms from humans and the model might not be exactly the same, but there is significant categorical or conceptual overlap between the two.

One may also note that the output vectors for poetic creativity appear to be rather "musical". This may reflect the fact that the input dimensions were provided by people who, overall, have significant musical experience - all but three of the participants have had musical training or have played music informally, whereas only four of the participants have experience with creative writing. People's experience with music might frame the way they think about other creative domains, or at the very least influencing the terms used to describe poetic creativity; consequently, this has led to a subspace that highlights the musical nature of this sample.

In light of our model's ability to find conceptually proximal terms, we propose that this method has the potential to be practically applied to the discovery of unexpected and valuable terms for the evaluation of creative output. Importantly, this approach may be applied to different corpora; for example, Wikipedia pages in different languages may be explored to address the difficult issue of identifying conceptually similar spaces across languages. The model's conceptual spaces concretely delineate evaluative terms that one person alone may not consider. For example, the terms "distinctiveness", "finesse", "artistry", and "stylization" were not cited by humans, but were within the top 30 output vectors discovered by the model. Future work may build on these findings, by using the model's discovered terms as criteria for subjective evaluation of creative output. In addition, discovering the geometry, flexibility, and contextual specificity of conceptual spaces may be very useful for assessing products or systems based on specific underlying concepts (or developed to address particular conceptual issues).

More generally, our method is presented as an implementation of the mapping of words to concepts: this approach charts a passage from a statistically tractable lexical space to the abstract but natively cognitive domain of ideas. The temporary and contextual aspect of this mapping is essential to its success: it is the flexibility of the model that allows for the bespoke generation of subspaces, just as it is the pragmatic frangibility of language that permits the ready-to-hand adaptation of meaning for unfolding expressive purposes. As can be seen in our results, the same terms arise in different constellations of meaning depending on the contextual perspective taken on the space. It is the strength of our language model that it can be adapted in this way, with the high dimensional arrangement of words allowing for their projection as multitudinous conceptual representations.

## Conclusion

We investigated the terms and concepts that individuals most strongly associate with creativity in the musical and poetic domains, and described a computational methodology for modeling these conceptual relationships. Our multidisciplinary approach employs methods inspired by computational linguistics, as well as methods from empirical psychology. There were several outcomes of this work: the output from our distributional semantics vector space model was compared with human responses on a word association task. Human-generated terms were found within the top 15 dimensions of our model's lexical space, despite the model's very large vocabulary. This served as validation that the

model discovers a lexical space that encapsulates the kind of terms humans use to describe these concepts.

Subsequently, the most frequently reported human terms were used as model input parameters for discovering conceptual spaces of lower dimensionality. Our model was able to find vastly reduced subspaces corresponding to MUSICAL CREATIVITY and POETIC CREATIVITY, which again captured semantically relevant terms, many corresponding directly to participants' terms, and others extending the list of terms to insightful new dimensions. In addition, by sampling word-vectors that fall near the centroid of the discovered conceptual mappings, we aimed to find potentially useful terms for the evaluation of creativity. Although computational and AI methods have generated many systems which aim to display creative behavior or produce creative artefacts, the evaluation of computational creativity remains distinctly problematic. Therefore, we offer our method and results as a formal approach to delineating conceptually-relevant criteria on which to base the evaluation of creativity and creative artefacts in future studies.

We saw, in terms of the most common dimensions in lexical space and the highest-mean word vectors in conceptual space, that the model is able to discover semantic categories and indices of concepts that are alligned to human conceptualizations. This said, the model did not capture all of the semantic categories cited by humans. The most noteworthy omission is in regards to emotion, as terms relating to affect and evoked emotional response were some of the most frequently cited terms for both musical and poetic creativity. Accordingly, future work will investigate why the model does not capture this cluster of emotion-related terms.

Further directions for the future include the application of this computational approach to other domains, such as "culinary creativity", both for the ontologically useful task of elaborating concepts themselves, and to create well-tailored terminology for the assessment of creative output from the corresponding domains. This methodology may also be used to approach the task of conceptual blending: rather than specifying input vectors that belong to only one concept, one may supply input dimensions from several. This could result in output terms discovered at the intersection of the lexical regions specified by the vectors' different input dimensions.

## Acknowledgments

The projects Lrn2Cre8 and ConCreTe acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grants number 610859 and 611733, respectively. This research is also supported by EPSRC grant EP/L50483X/1.

## References

Allott, N., and Textor, M. 2012. Lexical pragmatic adjustment and the nature of ad hoc concepts. *International Review of Pragmatics* 4(2).

Barsalou, L. W. 1993. Flexibility, structure, and linguistic vagary in concepts: Manifestations of compositional system of perceptual symbols. In Collins, A. F.; Gathercole, S. E.; Conway, M. A.; and Morris, P. E., eds., *Theories of Memory*. Hove: Lawrence Erlbaum Associates. 29–101.

Clark, S. 2015. Vector space models of lexical meaning. In Lappin, S., and Fox, C., eds., *The Handbook of Contemporary Semantic Theory*. Wiley-Blackwell.

Fauconnier, G., and Turner, M. 2008. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Gärdenfors, P. 2000. *Conceptual Space: The Geometry of Thought*. Cambridge, MA: The MIT Press.

Grice, H. P. 1969. Utterer's meaning and intention. *The Philosophical Review* 78(2):147–177.

Harris, Z. 1957. Co-occurrence and transformation in linguistic structure. *Language* 33(3):283–340.

Heath, D.; Norton, D.; Ringger, E.; and Ventura, D. 2013. Semantic models as a combination of free association norms and corpus-based approaches. In *IEEE International Conference on Semantic Computing*, 48–55.

Hesse, M. B. 1963. *Models and Analogies in Science*. New York: Sheed and Ward.

Hill, F.; Korhonen, A.; and Bentz, C. 2014. A quantitative empirical analysis of the abstract/concrete distinction. *Cognitive Science* 38:162–177.

Jäger, G. 2009. Natural color categories are convex sets. In *Logic, Language and Meaning - 17th Amsterdam Colloquium*.

Koestler, A. 1964. *The Act of Creation*. : Hutchinson.

Landauer, T.; Laham, D.; Rehder, B.; and Schreiner, M. E. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, 412–417.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR Workshop*.

Ontanón, S.; Zhu, J.; and Plaza, E. 2012. Case-based story generation through story amalgamation. In *Proceedings of the ICCBR 2012 Workshops*, 223–232. Citeseer.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*.

Schütze, H. 1992. Dimensions of meaning. In *Proc. ACM/IEEE Conference*, 787–796.

Turney, P. D., and Patel, P. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* (37):141–188.

Veale, T. 2006. An analogy-oriented type hierarchy for linguistic creativity. *Knowledge-Based Systems*.

Wittgenstein, L. 1953. *Philosophical Investigations*. Oxford: Basil Blackwell. 3rd ed. Trans. G. E. M. Anscombe.

# Player Responses to a Live Algorithm: Conceptualising computational creativity without recourse to human comparisons?

**Oliver Bown**
Design Lab
University of Sydney
NSW, 2006, Australia
oliver.bown@sydney.edu.au

**Abstract**

Live algorithms are computational systems made to perform in an improvised manner with human improvising musicians, typically using only live audio or MIDI streams as the medium of interaction. They are designed to establish meaningful musical interaction with their musical partners, without necessarily being conceived of as "virtual musicians". This paper investigates, with respect to a specific live algorithm designed by the author, how improvising musicians approach and discuss performing with that system.

The study supports a working assumption that such systems constitute a distinct type of object from the traditional categories of instrument, composition and performer, which are capable of satisfying some of the expectations of an engaging improvisatory performance experience, despite being unambiguously distinct from a human musician. I investigate how the study participants' comments and actions support this view. Specifically: 1) participants interacting with the system had a stronger sense of the nature of the interaction than when they were passively observing the interaction; 2) participants couldn't tell what the "rules" of the interactive behaviour were, and didn't feel they could predict the behaviour, but reported this as being a positive, engaging aspect of the experience. Their actions implied that the improvisation had purpose and invited engagement; 3) participants strictly avoided discussing the system in terms of virtual musicianship, or of creating original output, and preferred to categorise the system as an instrument or a composition, despite describing the interaction of the system as musically engaging; 4) participants felt the long-term structure was lacking.

Such results, it is argued, lend weight to the idea that as CC applications in real creation scenarios grow, the creative contribution of computer systems becomes less grounded in comparison with human standards.

## Introduction

Live algorithms (Blackwell, Bown, and Young, 2012) are software systems designed to autonomously perform music with live musicians, typically in an improvised music format. There has been a great deal of activity in this area recently, owing to the increasing ease with which artist programmers can put together powerful realtime systems incorporating machine listening, realtime synthesis and patterning, and forms of adaptive behaviour. Recent concerts, attached to electronic arts and music conferences such as the International Symposium on Electronic Arts (ISEA) 2013, and New Interfaces for Musical Expression (NIME) 2014, have demonstrated the diversity of approaches to live algorithms (see Bown et al. (2013) for a discussion of these concerts).

As in all aspects of computational creativity, the question of evaluation in live algorithms requires detailed consideration, as there are no simple, objective measurables that indicate when computer generation of output has been creatively successful. Two issues are important: how system output is evaluated by humans, and the extent to which we can attribute the creative component of the output to the system, rather than to its maker or to the 'inspiring set' (Ritchie, 2007): the set of all examples given to the system.

In live algorithms, the creative process is somewhat different from many instances of automated creative generation, since the output is always the result of the interaction between a human and a computer system. It is an interactive creative scenario. This muddies the issue of the attribution of the creativity further, but at the same time presents alternative, more tractable questions regarding the success of the system in its collaborative, improvisatory role.

Whilst it should be borne in mind that such questions regarding the interactive experience of live algorithms are separate from the core questions of computational creativity evaluation, there is still much to be learnt from such an analysis. In Bown (2014), I argue that a human-focused, qualitative, and strongly context-aware approach to studying computational creativity systems is important to advancing evaluation. In the case of an artistic robot, for example, one should begin by examining the full set of interactions between the system, its maker, its operator, its audience and so on, before deciding how one should frame questions of creative ability. This is to avoid the danger of inappropriately framing the activity and the agency of participants in that activity. How is the creative attribution divided between these actors? How do people perceive the system, not only in terms of good or bad output, but in terms of the way in which the system's activity is presented in a social context? Others, particularly Colton (e.g., (Colton et al., 2014)) have emphasised the management of the social interactive context in computational creativity, presented as a means to enhance

the perception of creativity, rather than as a means to better understand interaction with creative systems to improve their design and efficacy in areas of application.

Such developments point to the possibility that any proposed *measure* of the creativity of a system is significantly less fruitful than a *rich description* of the system as an agent with creative affordances described by its networks of interaction. This neutralises the crisis of working out how to score creativity, and provides simple practical analysis which can support real applications in the way that human computer interaction (HCI) and interaction design does to great success. Thus a qualitative descriptive approach is pursued in the present research, in order to build a rich descriptive understanding of human-machine creative interactions in practice in the context of live algorithms.

A central motivation for conducting the following study is to conduct computational creativity research that is more focused on the details of a participant's interaction with a creative system, involving a number of dimensions of experience that are relevant to creativity, and in doing so contribute to an understanding of how such systems work in practice in real creative contexts.

In this paper I study the responses of improvising musicians to Zamyatin, a live algorithm system that I have developed and worked with artistically since 2010. Zamyatin has performed with a wide variety of musicians. It is conceptually speaking a very simple system as far as creative systems go, in terms of the generation of original content *on its own*. Specifically it is less driven by the use of musical intelligence than by an interest in low-level gestural interaction. But in light of the value of diverse approaches to computational creativity, I view the system as a useful experiment in computational creativity in that it is successful in establishing an autonomy of behaviour, both conceptually and as perceived.

The questions the study looks at are focused on the ways in which participants experience and benefit from the creativity of a system: (1) how effective the system is at contributing to an effective performance; (2) the extent to which the participant experiences the system as autonomous, and also human-like, and how this influences other aspects of the perception of the system, and; (3) whether the participant experiences the system as originating novel output, and how this influences (and is possibly influenced by) the general perception of the system.

These are issues that we must clearly gain an understanding of as part of a body of knowledge in applied computational creativity. The computational creativity literature remains lacking in work that formally studies these basic forms of interaction and experience using qualitative methods.

The first question has self-evident value, and in one form or another is naturally asked in the course of creating any musical system. A challenge for a more experience- and interaction-focused computational creativity research program is to balance this goal with that of advanced computational generative sophistication. The perception of autonomy addressed in the second question is an important topic for the study of computational creativity. Autonomy is a critical component in the making of creativity: a system can only be called creative insofar as it possesses some degree of autonomy in the output it creates. Perceived autonomy may not be actual autonomy and vice versa, and actual autonomy anyway lacks a robust applicable definition. The distinction between software autonomy in general and human-like autonomy is one that will need to be unpacked further as we witness computationally creative systems at play in real interactive scenarios, and it is important to understand how individuals experience that autonomy and how that influences their behaviour towards the system and their own activities. Finally, in the context of interactive music creation we are interested in how the system can drive surprise and intrigue in the co-performer, and under what circumstances the performer acknowledges something as either creative, or in terms that connote creativity. Here it is particularly interesting to look at the language used, as this is an area where the anthropomorphism of cognition comes up easily.

I begin by describing the motivations behind the design of Zamyatin in the following section, before moving onto describing the study and results.

## Zamyatin

Zamyatin is a software system in ongoing development since 2010 (Bown, 2011). Before describing the design of Zamyatin, it is necessary to explain some of the design considerations, including a number of aesthetic decisions. An earlier description of Zamyatin's design is given in Bown (2011).

One of Zamyatin's main goals was to emphasise the experience of interacting with something that 'felt' autonomous and engaged in interaction, even if, it does not make sophisticated use of musical knowledge. For this, the free improvised mode provides a context that allows one to explore behaviour in a more abstract way than is afforded by many musical genres. Improvising software agents are a longstanding area of activity. George Lewis' Voyager system(Lewis, 2000) is a widely known example, and uses a hand-coded complex of interacting generative elements to create rich, diverse and musically responsive behaviour. Musicians performing with Lewis' system can be seen deeply engaged in the musical interaction as if performing with another human improviser. The use of a Disklavier (an acoustic piano that can be controlled by MIDI via mechanical actuators) limits the sense of a computer being involved. Artists such as Lewis have reported the responses of musicians performing with their systems, but such reports increasingly show that it is hard to pin down exactly how musicians think about, understand and evaluate such systems, suggesting the need for studies that get into more detail about the conceptual language and approaches used. Banerji (2012), for example, takes an anthropological approach, with a strong focus on working in real contexts, and looking as much at how the system influences the performer's behaviour as at how the performer judges the system. Other projects such as the work of Plans Casal and Morelli (2007), focus strongly on using low-level realtime audio analysis and resynthesis to give the performer a strong sense that the software acts as a responsive agent, through interactive immersion. Pachet's (2004) approach to establishing engagement is to mimic the

style of the improviser in a call and response fashion. Similarly with Blackwell and Young (2004) and Brown, Gifford, and Voltz (2013), who draw on a style analysis and resynthesis of the performer's input to establish a strong sense of engagement. Although these projects report on user-responses, further research is needed to determine whether these are indeed effective strategies for creating desirable interactive musical experiences.

A common challenge for the makers of generative systems is how to endow the system with autonomous behaviour that transcends the rules put into it by the programmer. That is, if your system is a collection of procedural instructions defined by the programmer, then even if the specific behaviour of the system is original, being some possibly unexpected product of the interacting rules, the general nature of the system's behaviour remains down to the programmer, since no new knowledge has been gained by the system.

There are three commonly cited ways around this problem (Todd and Werner, 1999). The first is already implied above: if the set of rules I provide are complicated enough, then from the interaction of these elements there will emerge new, higher-level behaviours that were not anticipated. The classic example is flocking behaviour, where the programmer defines the behaviour of individual 'boids' (Reynolds, 1987), but nowhere dictates that the system should start forming oscillating blobs on a macroscopic scale. Classical work from the generative art canon also highlight the value of this approach. Both Harold Cohen's celebrated AARON system (McCorduck, 1990) and George Lewis' Voyager system (Lewis, 2000) consist of complex rule sets that result in outcomes even their makers find surprising. In this case, it is perhaps wrong to describe what emerges from these systems as new knowledge.

The second approach is that the system learns. This is easily understood by analogy with how humans acquire knowledge that they are not born with. A large number of systems use learning to build musical knowledge, and famous examples include David Cope's EMI (Cope, 1996) and François Pachet's Continuator (Pachet, 2004). In these cases, the input knowledge now comes from a body of input musical data as well as the programmer. One problem then is how to avoid the system becoming just a copycat. The system needs not only to learn the style but to learn how to produce new material in that style. Current systems have yet to show how the learning itself can perform this extrapolation.

A third approach uses targeted evolution or another form of optimisation, dictated either by a measurable target behaviour, or user-feedback applied to a population of evolving behaviours. The rationale goes that a target behaviour itself does not contain the knowledge about how to achieve that behaviour, but running an evolutionary system to achieve that target can discover novel solutions which themselves constitute knowledge. Experiments in artificial evolution have shown the discovery of such solutions. For example, the coevolution of predator and prey systems reveal the emergence of specific hunting or hiding techniques (Cliff and Miller, 1995). Here the knowledge is produced through interaction, or learning-by-doing. Thus by specifying a target behaviour in the form of an evolutionary goal, one can drive a system to discover component behaviours that are not specified in that goal.

Unlike the majority of live algorithm approaches to deriving behaviour, Zamyatin is not a corpus or machine-learning based system, and employs this third approach to achieving autonomy. I draw on Blackwell and Young's PfQ framework to describe the system (Blackwell, Bown, and Young, 2012). Passing from the input (P) layer to the inner 'patterning' (f) layer are low-level feature values derived from the input musical data of the system. Passing from the inner layer to the 'instrument' or 'sounding' layer (Q) are control signals. These can be thought of as the equivalent to the human physical control 'signals' applied to a musical instrument, i.e., the movement of the hands, feet, breath, etc., although the object being controlled might involve its own generative elements. In Zamyatin, the inner patterning system is a type of decision tree, coupled with a internal array of states, that together feedback on themselves. This inner patterning system is connected to the outside world though the input layer and output layer. Somewhat like a traditional feedforward multilayer neural network, the connections between these layers flow in the forward direction only.

A decision tree is a binary tree that propagates a decision making iteration from the root of the tree to one of the leaves (leaves represent decisions), at each junction choosing to go one way or the other based on whether a single numerical value is above or below a single threshold. Decision trees are used commonly as efficient classifiers. The internal state array is simply an array of floating point values in the range [0,1]. In addition to the internal state, the system is constantly being fed an input state derived from low-level features of the incoming audio. Decisions at each node in the decision tree are made based on either the current state of the low-level audio features being passed into the system, or the internal state array. A leaf in the decision tree contains a list of actions which include passing on control commands to the musical system (Q) and also updating the state array. In this way, the decision tree and state array form a feedback system that can exhibit complex dynamics in the absence of any input, and can also be driven by changes to the input. Previous work (Bown and Lexer, 2006) has looked at the musical use of neural networks with similar properties.

An evolutionary approach is applied to the design of the decision tree, including the architecture of the tree (which can grow or shrink over time), the parameters of each decision node (which value to query and what threshold to apply) and the (variable length) list of actions to perform at each leaf. Actions control how the internal state array is updated, applying simple arithmetical operations to the state values.

The inner layer updates at a 'control rate' of around 20hz. It outputs two forms of control data at each update: a single integer, representing its current decision state, and an array of floating point values in the range [0-1], representing its internal state. Both are actually used to control the musical output. In evolving the system behaviour, a fitness function is hand-coded, that takes into account the pattern complexity, and other patterning properties such as degree of vari-

ability and repetition, of the system's output under various input conditions. Different variant fitness functions are used to create large populations of decision tree candidates, which are then creatively explored during the preparation of musical work.

Like procedural systems, Zamyatin does not draw on a corpus of musical knowledge, but instead attempts to establish novel behaviour through the interplay between the programmer specifying a behavioural target and the system evolving novel behaviours that achieve that target. The target behaviour does not describe the final musical output, but the output of the nested control system (f) that operates a number of virtual musical instruments. This target behaviour is defined by the programmer and the selection or definition of different target behaviours to suit the performing musician becomes part of the creative process of preparing Zamyatin for each new performance.

## Musical Study

Three improvising musicians (P1, P2, P3) were invited to attend a focus group to investigate musician responses to Zamyatin. The goal was not to set up a musical Turing test: there was no attempt to conceal the computational status of the system. Instead, the study looked at questions of engagement, experience and perception in improvised interaction.

The study was set up as a focus group in order to stimulate interaction between the participants, to look at the way they discussed musical interaction, and to get them to observe each other playing.

Participants were first shown a video recording of the system performing with a musician and asked questions about how they perceived the interaction with the musician. They were then played an audio recording of an earlier manifestation of the system performing with another musician and asked similar questions. They were then asked to perform with the system and develop their responses to it.

The author initially did not explain the design of the system, but later answered questions and provided more context as the study progressed, in response to the participants' questions.

Several other interviews with performers conducted prior to the focus group have influenced the expectations of the author in approaching the focus group. These will be reported in full in a forthcoming journal paper.

## Results

Three main results are considered here:

1. **Participants interacting with the system had a stronger sense of the nature of the interaction than when they were passively observing the interaction.**

   During the initial observation of the pre-recorded concert, all three participants said that they did not see any clear clues as to how the system was responding, what information it had access to from the musician, and what the interaction paradigm was. This was manifest largely in the sense of uncertainty surrounding the interaction. The musicians had no way of identifying clear paths of causality from the musician to the system.

   Of the system in general, P2 says the following:

   > The system of interaction is not obvious to me. . . . I can't tell. At times [the musician] is loud and I don't think the software's responding, or vice versa, and then sometimes the two things are loud or the two things are soft. The obvious parameters you can sample and listen to are like dynamics and pitch, timbral stuff . . . There doesn't seem to be any clear one-to-one relationships with what the software does, or it changes over time? Sometimes it reacts in a particular way and sometimes it doesn't.

   In performing with the system the musicians' responses shifted from this ambiguity to a greater sense of awareness that the system was responding to their playing. A good deal of uncertainty remained about precisely how the system behaviour was influenced by the musician, and as discussed below this is a theme in itself.

   After watching P1 performing, P2 says:

   > It was way more dynamic than it comes across in the flat stereo recordings, it was actually really good. It surprised me a few times how loud it was prepared to go and transgressive of the duo in a way . . . mainly with dynamics but sometimes placement too . . . it did some bizarre things and you go "oh that's cool". . . . but when it's compressed . . . you don't understand the dynamics that much. . . .
   > (Participant was asked to explain 'transgressive') It did naughty things, to do with timbres and placement. If it was someone playing that material you'd go, they're being a bit upfront, kicking the thing along a bit, putting provocations in. I like that.

   P3 adds:

   > That's the weird thing about it; you can really sense that something's happening but I can't tell what it is.

   Interestingly, also, the critical analysis of the system naturally extended to the performing musician as well. The evaluation of the improvisation by the participants naturally applied as much to the performers as to the systems. This may be more their habit, but of some relevance, Banerji (2012) has proposed looking at the impact on musicians' playing as a form of ethnographic approach to studying the qualities of live algorithms.

   > P3: One thing I found that the second musician wasn't interacting with the software at all. I felt like maybe they were just playing. I didn't hear too much active listening, they were obviously playing with it, but didn't really feel like they were kind of . . . that level of interactivity wasn't really there from their performance. . . . It was a real contrast of style I thought. I thought the first guy was really overtly interacting with it to quite a large extent, and the second I thought wasn't. But it's hard to say what the agenda is. . . . It's not that I enjoyed the first one better. It's more that if someone told me if the second player was in another room not being able to hear the performance I could believe you.

2. **Participants couldn't tell what the "rules" of the interactive behaviour were, and didn't feel they could predict the behaviour, but reported that they did experience the behaviour as interactive, and presented this uncertainty as being a positive, engaging aspect of the experience. Their actions implied that the improvisation had purpose and invited engagement.**

   In discussing what if any cues revealed the nature of interaction between system and human performer during the video playback section of the study, participants noted that any candidate explanations they developed for the behaviour of the system were frustrated by its seemingly changing interactive behaviour. For example, one participant began thinking that the system was matching the intensity of the performer's behaviour, but then found that the oppose suddenly occurred.

   During interaction with the system, performers remained unsure about exactly what the responsive behaviour consisted of, but reported that they did feel that there was some sort of complex interaction taking place, and finding this particularly engaging, owing to the uncertainty of the system's behaviour.

   > P2 (describing performance with P1): That started off with a noisy atmospheric tone. P1 came in and it maintained its thing, it kept its thing for a while. which kinda surprise me. I thought the introduction of a strong tone would shift it, but it didn't shift it and I thought "that's cool". ... the fact that it doesn't jump the whole time makes it worth listening to. If it was jumping the whole time with your stimuli, with the distinction from the live instrument to a clear distinction from that it would drive you crazy.

   This uncertainty was also described as potential source of frustration. Equally, the stability of the system over the long-term was described as a potential source of boredom. But on the whole participants agreed that the balance between uncertainty and predictability was well measured to create an effective sense of engagement for the musician.

   > P1: To begin with, and that's the same with the other ones I saw, it takes the musician to initiate the interaction. ... it was playing a long granulated tone, I came on top of that with a between note, probably to create some symbiosis with what it was doing. Then I found as I went into it that I wanted to find out that it reacted to what I was doing, and this was less clear. Sometimes it did and sometimes it didn't.

   > P3: There was a really loud section with no stimulus behind it, and its like, where did that come from, but I'm getting closer to seeing [the relationship] ... actually I'd find it quite stressful to perform with.

3. **Participants strictly avoided discussing the system in terms of virtual musicianship, or of creating original output, and preferred to categorise the system as an instrument or a composition, despite describing the interaction of the system as musically engaging.**

   The participants were clear explicitly – in response to direct questions about it – and implicitly – in the way they described the interaction with the system – that they felt no compulsion to see the system as a 'performer', preferring instead to view it as a form of complex instrument, or interactive score. However, the participants equally acknowledged that the behaviour of the system made it stand out from other types of digital interactive systems or instruments, particularly in terms of the autonomy of behaviour. To some extent this afforded the use of terms such as a perceived volition, that are arguably not normally associated with machine behaviour.

   As an example of a clear shortcoming, P2 states:

   > It seemed a bit confused with the very high frequencies ... I felt that it kind of suddenly went "I can't actually see you" ... It was quite interesting. If it was another player you'd go, ok, that's working.

   They elaborate on their perception of the system in terms of humanness:

   > I've steered clear from [referring to] anything to do with a performer because it doesn't feel like a human being at all, but it feels interesting, you've set up a compositional tool that's not momentary predictable but in the long term its predictable.

   The participant describes this engagement further as follows:

   > It was good, it was something I wanted to do listening to the other things: give it its own space, do its thing. It's an intriguing notion that you didn't play for a while and then it comes up with something else. It kinda lets the audience know. It's not some sort of stupid device, something of its own volition.

   When asked how it compared to 'mere tape', P2 elaborates:

   > I think audiences are pretty smart, they understand what tape is, what predetermination is and what liveness is, and if the audience were sitting there knowing that it's a live system and it seems to have some initiative without the player, I think that's an interesting moment. ... But I'd say the choosing the samples becomes this overridingly important compositional decision. ... I feel that with this, whatever samples you put into the composition ... the machine has some sort of ability to stop and start things.

4. **Participants felt the long-term structure was lacking.**

   It was widely agreed that the system did not convincingly deal with long term structural management of the performance.

   > P2: Listening to both of those things a lot of the activity is very much less than 3 seconds, so there's a lot of active many-events-per-minute sort of feel to it, and because it goes on for some time in that way it then has a sort of strange flatness as a result, and after a while you settle into the fact that there aren't going to be any super-long events, and so in a sense it kind of flattens the whole thing down and makes it kind of amorphous.

The participant frames this in the context of contemporary improvised music:

> It's a subject right at the heart of what's going on in improvised music. Probably always has been, but seems really central to it these days. It feels to be generational as well. The older generation may feel that they're not interacting and reacting (themselves) but they tend to more than younger generations. . . . I feel like there's players around now who work in much longer structures and they don't want to have a dialogue which is over some 10 second framework.

## Discussion

The results of this study go some way to confirming existing assumptions and findings about evaluation of musical systems.

The first result affirms a general principle that certain knowledge is better acquired through active participation. Interacting with a system tells you more about its interactive capacity than watching an interaction with a third party. This may not be manifest in the form of a expressible understanding of what the system is doing, as was the case in the present study, but nevertheless the participant in the interaction gains a direct sense of the interactive nature of the system, that may be obscure from outside.

This has implications for the audience experience of the work. They may not be fully aware of the experience of the musician during the performance. On the other hand, the expression and observed response of the musician can be important to an observer understanding the interaction, and may indirectly reveal the experience. Pachet has shown how the video footage of participants, or simply composers engaged with the treatment of their own work, can do a fantastic job of revealing basic facets of user-experience (Addessi and Pachet, 2006; Pachet, 2014).

Related to this, a common theme in the evaluation of autonomous music and art system is the question of making use of a Turing-style test (e.g., (Ariza, 2009; Bishop and Boden, 2010; Pease and Colton, 2011)). Results such as those of Moffat and Kelly (2006) show that positive results can be easily achieved in situations where people try to guess whether artefacts were computer or human generated, i.e., the system generated output can pass as human. However, without involving any form of probing or interaction with the system, the test in this form doesn't really tell us anything about the system, its intelligence or creative capacity (Pease and Colton, 2011). Despite what is said about the great communicative power of art and music, these artefacts form a poor window onto their creators.

Nevertheless, it is still reasonable to expect that in general there are *cues* in creative outputs which reveal aspects of the nature of the system producing them, and which may be identified in interactive scenarios, but also possibly without the need for interaction. These cues may not be reliable identifiers of whether or not the system is computer or human, and should be better understood as contributing to a qualitative evaluation of creative or interactive behaviour. More generally, we may talk of the character of the system and how it contributes to or stimulates a productive musical process.

The musicians participating in the study did develop a sense of the cues that indicated Zamyatin's responsive behaviour in certain ways, sometimes, but without certainty. This led them to feel that the system was nontrivial and invited an engagement with the behaviour of the system.

Related to this are the other two results. The character of the system is one in which an actively obscure relationship between performer action and system result is sought. To this end the evolutionary strategy has proven to be a convenient approach to relieving the system designer from the task of dictating the system's response directly, working around the "Lovelace objection" that a computational system might only do what it has been programmed to do.

Finally we come to the issue of whether the system was at all perceived as bearing the qualities of a human performer. The response was resoundingly negative in answer to this question. Whilst, as stated in the introduction, the aim of the system design was never to simulate or mimic human behaviour, a stated goal has been to explore the middle ground between inanimate objects that do not exhibit adaptive or proactive behaviours, and sentient humans, or other creatures. The participants unambiguously placed the system in the category of objects, as opposed to any sort of 'performer', equating it either to an instrument or composition. It does not follow that they perceived this object as dumb or lacking lifelike properties.

### Scoring Zamyatin

From these results we can consider the questions posed at the beginning of the paper:

1. (Q1) *How successful is it as at creating effective performances with improvising musicians?* The participants' responses give enough support to a positive answer to this question, specifically in terms of the interesting dynamics produced by the system's interactive behaviour.
2. (Q2) *To what extent do performers conceptualise of and perceive Zamyatin as autonomous, as well as human-like, and how does this influence other aspects of the perception of the system?* The responses are more ambiguous with respect to this issue, not least because the definition of autonomy is itself hard to pin down in application. Because of this, participants were not asked to discuss autonomy directly, but we may make inferences based on their responses. Significantly, they perceived the system as being both (i) not passive in the form of its responses to input, and (ii) able to drive the performance through spontaneous action that appeared to come from nowhere. These support a technical definition of autonomy as behaviour that is not entirely determined from outside of the system, and the varying nature of the system's predictability supports an information theoretic form of this. However, there are other senses of perceived autonomy that could be achieved. Future studies could work towards understanding in greater detail the space of possible types of autonomy (for example as discussed in Eigenfeldt et al. (2013)) that might be perceived in a system.

3. (Q3) *Does Zamyatin actually originate novel responses as far as the performers are concerned?* The predominant response to this from participants was 'no'. They quickly perceived that the system worked within strict limits, with the musical style and much of the content (e.g., choice of sounds, pitch sets, etc.) dictated by the system designer. But equally the participants responses indicated that they did attribute actions to the idiosyncratic nature of the system, which was described to them as having resulted from an evolutionary search that went beyond the input of the designer. For example, on numerous occasions the behaviour of the system was described by participants as surprising, and not like anything a human would do, coupled with value judgements ranging from this being highly engaging, to it being frustrating. We could claim that a surprising and valued response is technically speaking creative, according to the most commonly agreed definition of the term. This would be a generous interpretation, since many 'dumb' processes might achieve some such level of surprise in interaction. If instead we were to apply Colton's 'creativity tripod' of imagination, skill and appreciation (Colton, 2008), we would have to accept that at best only skill could be claimed (I would claim that the system can appear skilful in the complex manipulation of electronic sound). An open question is what kinds of systems stimulate 'perceived imagination' and 'perceived appreciation', and whether these are in fact always relevant in contexts such as this: is it important to the perception of musical creativity that such elements are perceived?

## Conclusion

The evaluation of systems from computational creativity, using qualitative analysis grounded in specific contexts of creative interaction, is an important part of the emerging suite of research methods we use to discover and understand how systems can act successfully to support creativity or act as creative agents. This paper has attempted to dig deeper into how improvising musicians, presented with a live algorithm system, approach, interpret and engage with that system in an applied context.

The results suggest ways in which the system, Zamyatin, could be improved to create more compelling improvised musical experiences. Good long-term structure is a challenging area that this system could improve upon. The results appear to affirm the value of exploring forms of software-based musical agency that does not conform to human modes of behaviour but that still produce engagement. This could be developed further by categorising these kinds of behaviour.

The relationship between the behaviour of the system and the engagement of the performers could be developed by improving the user-interface to the underlying evolutionary techniques, possibly using interactive evolutionary techniques, so that there is a real capacity for a musician to feedback on and modify the behaviour of the agents. It was also apparent from the study that certain traits of the system, such as the degree of uncertainty of its behaviour, could be explicitly recognised as adding to the musicality of the system, and could be codified into future fitness functions. An immediate goal for Zamyatin is to create a modular system that can be easily incorporated into live performance sets by non-programmer musicians, and these ideas can be incorporated into that design.

In addition, the view held by this author is that questions of computational creativity are now shifting towards more applied areas where the comparison with human creative activity is less of a concern than a more open-ended understanding of how machines may act creatively. This is weakly supported by the research in this paper, in which a failure to stand up to any sort of Turing-style test does not diminish the discussion of the creative potential of the system. This is a perspective that warrants further study across a range of systems.

## References


Addessi, A. R., and Pachet, F. 2006. Young children confronting the continuator, an interactive reflective musical system. *Musicae Scientiae* 10(1 suppl):13–39.

Ariza, C. 2009. The interrogator as critic: The turing test and the evaluation of generative music systems. *Computer Music Journal* 33(2):48–70.

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 (AIIDE'12) Conference*.

Bishop, M., and Boden, M. A. 2010. The turing test and artistic creativity. *Kybernetes* 39(3):409–413.

Blackwell, T., and Young, M. 2004. Self-organised music. *Organised Sound* 9(2):137–150.

Blackwell, T.; Bown, O.; and Young, M. 2012. Live algorithms. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Springer-Verlag.

Bown, O., and Lexer, S. 2006. Continuous-time recurrent neural networks for generative and interactive musical performance. In Rothlauf, F., and Branke, J., eds., *Applications of Evolutionary Computing, EvoWorkshops 2006 Proceedings*.

Bown, O.; Eigenfeldt, A.; Martin, A.; Carey, B.; and Pasquier, P. 2013. The musical metacreation weekend: challenges arising from the live presentation of musically metacreative systems. In *Proceedings of the musical metacreation workshop, AIIDE conference, Boston*.

Bown, O. 2011. Experiments in modular design for the creative composition of live algorithms. *Computer Music Journal* 35(3).

Bown, O. 2014. Empirically grounding the evaluation of creative systems: incorporating interaction design. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Brown, A.; Gifford, T.; and Voltz, B. 2013. Controlling interactive music performance (cim). In *Proceedings of the Fourth International Conference on Computational Creativity*, 221.

Cliff, D., and Miller, G. F. 1995. Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. In *Advances In Artificial Life*. Springer. 200–218.

Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On acid drops and teardrops: observer issues in computational creativity. In *Proceedings of the 7th AISB Symposium on Computing and Philosophy*.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.

Cope, D. 1996. *Experiments in Musical Intelligence*. Madison, WI: A-R Editions.

Eigenfeldt, A.; Bown, O.; Pasquier, P.; and Martin, A. 2013. Towards a taxonomy of musical metacreation: Reflections on the first musical metacreation weekend. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Lewis, G. E. 2000. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal* 10:33–39.

McCorduck, P. 1990. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. W. H. Freeman and Co.

Moffat, D. C., and Kelly, M. 2006. An investigation into people's bias against computational creativity in music composition. *Assessment* 13:11.

Pachet, F. 2004. Beyond the cybernetic jam fantasy: The continuator. *IEEE Computer Graphics and Applications* 24(1):31–35.

Pachet, F. 2014. Non-conformant harmonization: The real book in the style of take 6. In *Proceedings of the Fifth International Conference on Computational Creativity*.

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 symposium on AI and Philosophy*.

Plans Casal, D., and Morelli, D. 2007. Remembering the future: An overview of co-evolution in musical improvisation. In *Proceedings of the 2007 International Computer Music Conference*.

Reynolds, C. W. 1987. Flocks, herds and schools: A dstributed behavioural model. In *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, volume 21, 25–34.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Todd, P. M., and Werner, G. 1999. Frankensteinian methods for evolutionary music composition. In Griffith, N., and Todd, P. M., eds., *Musical Networks: Parallel Distributed Perception and Performance*. Cambridge, MA: MIT Press/Bradford Books. 313–339.

# Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble

**Arne Eigenfeldt**
School for the Contemporary Arts
Simon Fraser University
Vancouver, Canada
arne_e@sfu.ca

**Oliver Bown**
Design Lab
University of Sydney
NSA, Australia
oliver.bown@sydney.edu.au

**Benjamin Carey**
Creativity and Cognition Studios,
University of Technology Sydney
NSW, Australia
Benjamin.Carey@uts.edu.au

**Abstract**

In this paper, we describe the musebot and the musebot ensemble, and our creation of the first implementations of these novel creative forms. We discuss the need of new opportunities for practitioners in the field of musical metacreation to explore collaborative methodologies in order to make meaningful creative and technical contributions in the field. With the release of the musebot specification, such opportunities are possible through an open-source, community-based approach in which individual software agents are combined to create ensembles that produce a collective composition. We describe the creation of the first ensemble of autonomous musical agents created by the authors, and the questions and issues raised in its implementation.

## Introduction

Musical metacreation (MuMe) is an emerging term describing the body of research concerned with the automation of any or all aspects of musical creativity. 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 and Young 2005). Metacreation (Whitelaw 2004) involves using tools and techniques from artificial intelligence, artificial life, and machine learning, themselves often inspired by cognitive and life sciences. MuMe suggests exciting new opportunities for creative music making: discovery and exploration of novel musical styles and content, collaboration between human performers and creative software partners, and design of systems in gaming, entertainment and other experiences that dynamically generate or modify music.

A recent trend in computational creativity, echoing other fields, has been to develop software infrastructures that enable researchers and practitioners to work more closely together, taking a modular approach that allows the rapid exchange of submodule elements in the top-down design of algorithms, facilitating serendipitous discovery and rapid prototyping of designs. It is widely recognised that such infrastructure-building can accelerate developments in the field for a number of reasons: getting large numbers of researchers to work together on larger-scale projects, forcing researchers to develop their software in a sharable format, enabling the like-for-like comparison of different system designs, education, and directly providing a large framework for further software development. Charnley *et al.* (2014), for example, has proposed a cloud-based collaborative creativity tool, supported by a web interface, that allows the rapid creation of text-based, domain specific, creative agents such as Twitter bots.

Our research in MuMe, which risks being too localised and insular, will benefit from a similar direction, and for this reason we have proposed the "musebot ensemble", a creative context designed to bring researchers together and get their realtime generative software systems playing together. We present a recent effort to design and build the infrastructure necessary to bring together community-created software agents in multi-agent performances, an elaboration on the motivation for doing so and the opportunities it offers, and some of the challenges this project brings. So far, we have set up a specification for musebot interaction, involving a community engagement process for getting a diversity of thoughts on the design of this specification, and we have built a number of tools that implement that specification, including musebots and a musebot conductor.

Following the outline of the system, we describe the creation of our first exploratory attempts to create and run a MuMe ensemble. We describe our initial experiences working creatively with networks of musebots. We conclude the paper with several open questions that were raised in the implementation of this collaborative compositional experience.

## Towards a Collaborative Composition by Creative Systems

The established practice of creating autonomous software agents for free improvised musical performance (Lewis 1999) – the most common domain of activity in MuMe research – often involves idiosyncratic, non-idiomatic systems, created by artist-programmers (Rowe 1992, Yee-King 2007). A recent paper by the authors (Bown *et al.* 2013) discussed how evaluating the degree of autonomy in such systems is non-trivial and involves detailed discussion and analysis, including subjective factors. The paper identified the gradual emergence of MuMe specific genres — i.e., sets of aesthetic and social conventions — within which meaningful questions of relevance to MuMe research could be further explored. We posited that through

the exploration of experimental MuMe genres we could create novel but clear creative and technical challenges against which MuMe practitioners could measure progress.

One potential MuMe genre that we considered involves spontaneous performance by autonomous musical agents interacting with one-another in a software-only ensemble, created collaboratively by multiple practitioners. While there have been isolated instances of MuMe software agents being set up to play with other MuMe software agents, this has never been seriously developed as a collaborative project. The ongoing growth of a community of practice around generative music systems leads us to believe that enabling multi-agent performances will support new forms of innovation in MuMe research and open up exciting new interactive and creative possibilities.

## The Musebot Ensemble

A musebot is defined as a "piece of software that autonomously creates music collaboratively with other musebots". Our project is concerned with putting together musebot ensembles, consisting of community-created musebots, and setting them up as ongoing autonomous musical installations. The relationship of musebots to related forms of music-making such as laptop performance is discussed in detail in our manifesto (Bown *et al.* 2015).

The creation of intelligent music performance software has been predominantly associated with simulating human behaviour (e.g., Assayag *et al.*). However, a parallel strand of research has shed the human reference point to look more constructively at how software agents can be used to autonomously perform or create music. Regardless of whether they actually simulate human approaches to performing music (Eldridge 2007), such approaches look instead at more general issues of software performativity and agency in creative contexts (Bown *et al.* 2014). The concept of a "musebot ensemble" is couched in this view. i.e., it should be understood as a new musical form which does not necessarily take its precedent from a human band.

Our initial steps in this process included specifying how musebots should be made and controlled so that combining them in musebot ensembles would be feasible, and have predictable results for musebot makers and musebot ensemble organisers. Musebots needn't necessarily exhibit high levels of creative autonomy, although this is one of the things we hope and expect they will do. Instead, the current focus is on enabling agents to work together, complement each other, and contribute to collective creative outcomes: that is, good music.

This defines a technological challenge which, although intuitive and easy to state, hasn't been successfully set out before in a way that can be worked on collaboratively. For example, Blackwell and Young (2004) called on practitioners to work collaboratively on modular tools to create live algorithms (Blackwell and Young 2005), but little community consensus was established for what interfaces should exist between modules, and there was no suitably compelling common framework under which practitioners could agree to work. In our case, the modules correspond clearly to the instrumentation in a piece of music, and the context is more amenable to individuals working in their preferred development environment.

In order for musebots to make music together, some basic conditions needed to be established: most obviously the agents must be able to listen to each other and respond accordingly. However, since we do not limit musebot interaction to human modes of interaction, we do not require that they communicate only via human senses; machine-readable symbolic communication (i.e., network messaging) has the potential to provide much more useful information about what musebots are doing, how they are internally representing musical information, or what they are planning to do. Following the open community-driven approach, we remain open to the myriad ways in which parties might choose to structure musebot communication, imposing only a minimal set of strict requirements, and offering a number of optional, largely utilitarian concepts for structuring interaction.

## Motivation and Inspiration

One initial practical motivation for establishing a musebot ensemble was as a way of expanding the range of genres presented at MuMe musical events. To date, these events have focused heavily on free improvised duets between human instrumental musicians and software agents. This format has been widely explored by a large number of practitioners; however, it runs the risk of stylistically pigeonholing MuMe activity.

For the present project, the genre we chose to target was electronic dance music (EDM), which, because it is fully or predominantly electronic in its production, offers great opportunities for MuMe practice; furthermore, metacreative research into this genre has already been undertaken (Diakopoulos *et al.* 2009; Eigenfeldt and Pasquier 2013). The 2013 MuMe Algorave (Sydney, 2013) showcased algorithmically composed electronic dance music, an activity originally associated with live coding (Collins and McLean 2014). However, rather than presenting individual systems with singular solutions to generating such styles, it was agreed that performances should be collaborative, with various agents contributing different elements of a piece of music. This context therefore embodies the common creative musical challenge of getting elements to work together, reconceived as a collective metacreative task. Although the metaphor of a *jam* comes to mind in describing this interactive scenario, we prefer to imagine our agents acting more like the separate tracks in a carefully crafted musical composition.

We acknowledge the relationship of musebot ensembles to multi-agent systems (MAS); however, rather than concentrate upon the depth of research within this field, we have designed the specification in such a way so as to combine generality and extensibility with domain specific functionality. As will be described, at heart the musebot project is simply a set of message specifications that are

domain specific to the idea of multiple *musical* agents. We feel that there is no need to draw on more specific MAS tools and specifications, as there is nothing that is not simply handled by the definition of a few messages. Taking this general approach has the advantage that if people want to incorporate the musebot specification into their MAS frameworks, they can. It is intentionally barebones so that it is simple for people to adapt their existing agents to be musebots. At the same time, we also acknowledge that MAS have been incorporated into MuMe in typically idiosyncratic ways, replicating the interaction between human musicians (Eigenfeldt 2007) while also exploring non-human modes (Gimenes *et al.* 2005); our intention is for musebots to explore both approaches.

We summarise the other opportunities we see in pursuing this project as follows, beginning with items of more theoretical interest, followed by those of more applied interest:

- Currently, collaborative music performance using agents is limited to human-computer scenarios. These present a certain subset of challenges, whereas computer-computer collaborative scenarios would avoid some of these whilst presenting others. Such challenges stimulate us to think about the design of metacreative systems in new and potentially innovative ways;
- It provides a platform for peer-review of systems and community evaluation of the resulting musical outputs, as well as stimulating sharing of code;
- It provides an easy way into MuMe methods and technologies, as musebots can take the form of the simplest generative units, whereas at present the creation of a MuMe agent is an unwieldy and poorly bounded task;
- It outlines a new creative domain, which explores new music and music technology possibilities;
- It encourages and supports the creation of work in a publicly distributed form that may be of immediate use as software tools for other artists;
- It allows us to build an infrastructure which can be useful for commercial MuMe applications. Specifically, it provides a modular solution for the metacreative workstations of the future;
- It defines a clear unit for software development. Musebots may be used as modular components in other contexts besides musebot ensembles.

## The Musebot Agent Specification

An official musebot agent specification is maintained as a collaborative document, which can be commented on by anyone and edited by the musebot team[1]. An accompanying BitBucket software repository maintains source samples and examples for different common languages and platforms[2].

A musebot ensemble consists of one musebot conductor (MC) and any number of musebots, running on the same machine or multiple machines over a local area network (LAN). The MC is notified of each musebot's location and paths to its directories, allowing it to build an inventory of the available musebots in the ensemble. Thus, for the user, adding a musebot to the ensemble simply means downloading it to a known musebot folder. Musebots contain *config* files that are controlled by the MC, and human/machine readable *info* files that give information about the musebots.

The MC is responsible for high level control of connected musebot agents in the network, setting the overall clock tempo of the ensemble performance and managing the temporal arrangement of agent performances (see Tables 1 and 2). The MC also assists communication between connected agents by continuously broadcasting a list of all connected agents to the network, and relaying those messages that musebots choose to broadcast. The MC is not necessarily "in charge". Currently, it is just a simple GUI program that allows users to control musebots remotely. Ultimately we will automate ensemble parameters such as tempo and key *either* by making specific variants of the MC, *or* by writing dedicated planning agents that issue instructions to the MC, *or* by allowing a distributed self-organising approach in which different agents can influence these parameters. These are all valid designs for a musebot ensemble.

| **/mc/time** <double: tempo in BPM> <int: ticks> |
|---|
| *This is the clock source and timing information. A beat/tick count, starting at zero and incrementing indefinitely, is sent at a rate of 16 ticks per beat at the specified tempo, to be used for synchronising your client bot. The downbeat is on (tick % 16 == 0).* |
| **/mc/agentList** <string: musebot ID> [<string: musebot ID> … ] |
| *List of connected musebots in your network. Use this list to reveal messages sent from specific musebots,* |
| **/mc/statechange** <string: {first,next,previous,any}> |
| *This parameter is designed to facilitate high level state changes, which could be anything, depending on the program; however some examples might be overall density of events, range/register, key changes, change in timbre etc.* |

**Table 1.** Example messages broadcast by the MC to all musebot agents.

| **/agent/kill** (no args) |
|---|
| *Exit gracefully upon receiving this message from the MC.* |
| **/agent/gain** <double: gain> [<double: duration ms>] |
| *Scale your output amplitude, used to apply a linear multiplication of your output audio signal.* |

**Table 2.** Example messages sent between the MC and specific musebot agents.

[1] tinyurl.com/ph3p6ax

[2] bitbucket.org/obown/musebot-developer-kit

Musebots may broadcast any messages they want to the network, providing they maintain their unique name space allocated for inter-musebot communication (see Table 3). Our musebot specification states that a musebot should also "respond in some way to its environment", which may include any OSC messages (Wright 1997) as well as the audio stream that is provided: a cumulative stereo mix of all musebot agents actively performing. It should also not require any human intervention in its operation. Beyond these strict conformity requirements, the qualities that make a good musebot will emerge as the project continues.

| **/broadcast/statechange <string: musebot ID> <string: {first,next,previous,any}>** |
|---|
| *Locally controllable high level state change. Use this parameter if you want to prompt other clients to make changes to their high level state. Equally, respond to this message if you want other musebots to prompt high-level changes.* |
| **/broadcast/notepool <string: musebot ID> <int_array: pitch class MIDI values>** |
| *A list of MIDI note values, pitch classes only, no octave info, to be shared with the network - e.g. chord or scale you are currently playing.* |
| **/broadcast/datapool** <string: musebot ID> <double_array: datapoints> |
| *Array of floating-point values.* |

**Table 3.** Example messages broadcast by musebot agents. These messages are speculative, and open for discussion.

## The First Musebot Ensemble

At the time of writing, a draft musebot conductor is implemented and published and a call has gone out for participation in the first public musebot ensemble. Our first experiments with making musebot ensembles followed the obvious path of taking the systems we have already created and adapting them to fit the specification. This step constituted provisional user testing of the specification and support tools and also gave us a sense of what sort of creative and collaborative process was involved in working with musebots.

We present two studies here. In the first case, the first author built a musebot ensemble entirely alone. The first author works regularly with multi-agent systems within his MuMe practice, so this was a natural adaptation of his existing approach. In the second study, each of the authors contributed a system that they had developed previously, and we looked at the ways that these systems could use the musebot specification to interact musically.

### First Author Working Alone

In the first study, several musebots were designed in isolation by the first author. While lacking the musebot goal of cooperative development, the situation did allow for the design of ensembles with a singular musical goal, including specific roles for each musebot. For example, a *ProducerBot* was created that functions to control various other instrumental bots – a *DrumBot*, a *PercussionBot*, a *BassBot*, a *KeyboardBot*, etc. – in a hierarchical fashion. The organisation of such an ensemble reflects one conception in achieving a generative EDM work, in which each run produces a new composition whose musical structure is generated by the *ProducerBot*, and the musical surface is produced and continuously varied by the individual instrumental musebots. Such a design has been previously implemented by the first author (Eigenfeldt 2014) to produce successful musical results. This top-down, track-by-track breakdown of relations between musical parts is of course completely familiar to users of DAWs, with the difference that each track is a generative process that receives high-level musical instructions from the *ProducerBot*. In this case, the *ProducerBot* sends out information at initialisation, including a suggested phrase length (i.e. 8 measures), and subpattern, which represents how the phrase repetition scheme can be represented (i.e. aabaaabc). It individually turns instrumental musebots on and off during performance, including syncronising them at startup. Furthermore, it sends a relative density request – a subjective number of possible events to perform within a measure – every 250 milliseconds, as well as progress through the current phrase. Lastly, at the end of a phrase, it may send out a section message (i.e. A B C D E). When an instrumental musebot receives this section descriptor, it looks to see if it has data stored for that section: if not, it stores its current contents (patterns), and generates new patterns for the next section; if it does have data for that section, it recalls that data, thereby allowing for large-scale repetition to occur within the ensemble.

As with a DAW, via the musebot specification, we inherently allow for community contributions that accept specific instructions from the *ProducerBot*: swapping a different *BassBot*, for example, in the ensemble would result in a different musical realisation, as it is left to the musebots to interpret the performance messages.

### Multiple Authors Working Together

In the second study, the three authors brought together existing systems into the first collaboratively made musebot ensemble. No assumptions were made in advance about how the systems would be made to interact, except that the second and third authors drew their contributions from existing work with live algorithms in an improvisation context (Blackwell and Young 2005), where the audio stream is typically the only channel of interaction.

A *BeatBot* was created by the first author, which combines the rhythmic aspects of both the former drum and percussion musebots, together with the structure-generating aspects of the *ProducerBot*, resulting in a complex and autonomous beat generating musebot. With each run, a different combination of audio samples is selected for the drums and four percussion players, along with constrained limitations to the amount of signal processing applied. A musical form is generated as a finite number of phrases, themselves probabilistically generated from weightings of 2, 4, 8, 16, and 32 measures. Each phrase has

a continuously varying density, to which each internal instrument responds differently by masking elements of its generated pattern. The metre is generated through additive processes, combining groups of 2 and 3, and resulting in metres of between 12 and 24 sixteenths. Finally, the amount of active layers for each phrase is generated. All of the generated material – metre, phrase length, rhythmic grouping, density, and active layers – is broadcast to the ensemble as messages.

The second author's *DeciderBOT* was adapted from his live algorithm system *Zamyatin*, an improvising agent that is based upon evolved complex dynamical systems behaviours derived from behavioural robotics (Bown *et al.* 2014). The internal system controls a series of voices that are hand-coded generative behaviours. *Zamyatin* is most easily described as a reactive system that comes to rest when presented with no input, and is jolted to live when stimulated by some input. The stimulation can send it into complex or cyclic behavior.

The final contribution to the first musebot ensemble was *_derivationsBOT*, designed by the third author. An adapted version of the author's *_derivations* interactive performance system (Carey 2012), *_derivationsBOT* was designed to provide a contextually-aware textural layer in the musebot ensemble, responding to a steady stream of audio analysis from the other bots connected to the network. During performance, *_derivationsBOT* analyses the overall mix of the musebot ensemble by segmenting statistics on MFCC vectors analysed from the live audio. The musebot compares these statistics with a corpus of segmented audio recordings, retrieving pre-analysed audio events to process, that compliment the current sonic environment. Synchronised to the overall clock pulse received from the MC, a generative timing mechanism conducts six internal players that process and re-synthesise these audio events via various signal processing. Importantly, the choice of audio events made available for processing is based upon comparisons both between statistics analysed from the live audio stream, as well as statistics passed between the internal players themselves. Thus, without audio input for analysis *_derivationsBOT* self-references, imbuing it with a sense of generative autonomy in addition to its sensitivity to its current sonic environment. To facilitate this, *_derivationsBOT* is randomly provided an internal state upon launch, enabling the musebot to begin audio generation with or without receiving a stream of live audio to analyse.

With the three musebots launched, a quirky, timbrally varied, somewhat aggressive, EDM results. Like much experimental electronic music, the listening pleasure is partly due to the strangeness and suspense associated with the curious interactions between sounds. The *BeatBot* was not designed to respond to any input and so drove the interaction, with the other two systems reacting. Thus, although very simple and asymmetrical as an ensemble, the musical output was nevertheless coupled. Since the *BeatBot* is not limited to regular 4/4 metre, it creates dubious non-corporeal beats to which both *DeciderBot* and *_derivationsBot* respond in esoteric fashions. In addition, *BeatBot* kills itself once its structure is complete, and the other two audio-responsive musebots, lacking a consistent audio stream to which to react, tend to slowly expire, bringing an end to each ensemble composition.

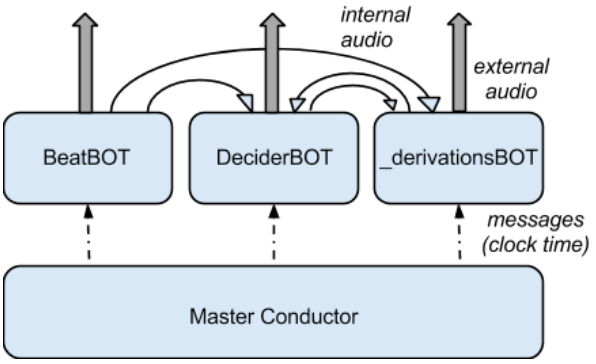



**Figure 1.** Audio and message routing in the second described musebot ensemble.

Example interactions between musebots, including this second example, are available online[3].

## Issues and Questions

These studies give insights into how a musebot approach can serve innovation in musical metacreation. Two areas of interest are: (1) what can we learn by dividing up musically metacreative systems into agents and thinking about how the communication between these serves musical goals? (2) related to this, how do we work with others and negotiate the system design challenges?

1. Increasingly, musicians are incorporating generative music processes into their work. Thus, the situation described above — managing several generative interacting processes — is not uncommon. The creative process is different to traditional electronic music composition because rather than making a specific change and listening to a specific effect that results from that change, one is in a state of continuous listening, as the result of a change might have multiple effects or take time to play out. It is common for electronic music composers to work with complex systems of feedback, and this process is similar, if more algorithmic. One effect of this is that it can dull decision making, as one gives over to the nature of the systems, or is unclear on what modifications will influence them effectively. Placing these musebots together in an ensemble positions us as both curator and designer: in the former case, one is forced to decide whether the musebots are interacting in a fashion that is considered interesting, and whether fewer, or more, musebots would solve any musical issues. We foresee such decisions to be more common as we accu-

[3] http://metacreation.net/musebot-video/

mulate more musebots, particularly those with clear stylistic bents. In the latter case, as designers we are placed into a more traditional role, in which continual iteration between coding, listening, critiquing, re-designing, and coding again guides both technical and aesthetic decisions. While we have no control over the other musebots, we can individually control how our own musebot reacts to other musebot actions, even if those actions are seemingly unpredictable.

2. Working together in this way offers a new approach to musical metacreation, along with a new set of challenges. In building systems, we are typically free to pursue our own aesthetic directions, and make individual decisions, both technical and aesthetic, as to how these systems should act and react. In the case of *BeatBot*, such a "closed system" is maintained, albeit with the addition of transmitting messages regarding its current state. In the case of *DeciderBot* and *_derivationsBot*, these existing systems had previously interacted with human musicians, and could rely upon the performer's intuitive musical responses to enhance those decisions made computationally. Within the musebot ensemble, both systems are now reacting to other machines: one that is essentially indifferent, and another whose reactions had previously been keyed to human actions.

As is often the case in experimental music production, having set up the interaction between agents and listening to how this interaction unfolds, we found clearly musically interesting content in this first attempt at a musebot ensemble. We anticipate many more musebots being designed and contributed, and imagine that through the unexpected combinations of such autonomous music-generating systems new thinking about automating musical creativity, and making it available to a wide community of users, might arise.

The current work is a small affirmation of the potential of a musebot approach, and several questions have arisen regarding the next stage of development. Our next step is to curate a number of musebots to be presented in an ongoing installation of interchangeable ensembles across different genres. In order to reach such a stage of development, the following questions need to be addressed:

**What kinds of interaction are useful – both computationally and musically?** At the moment, the three musebots are not sharing any information in the form of network messages. Firstly, the *BeatBot* is generating beats, entirely unaware of any reactions to its audio, and while the two responsive audio musebots generate emergent musical material driven by audio analysis, they are oblivious to any structural decisions being made by the rest of the ensemble due to their lack of messaging. While such independence is one aesthetic solution, a more responsive and self-aware environment will need to be explored, if for no other reason than structural variety. In the present ensemble, one approach could be to augment the capabilities of *DeciderBOT* and *_derivationsBOT* to allow network messages from *BeatBOT* to have an affect on their internal generative capabilities, such as levels of density and musical timing. Alternatively, an augmentation of *BeatBOT*'s capabilities as a producer could enable it to direct high-level changes in state in each of the connected bots, a possibility anticipated in the musebot specification by the availability of the statechange message.

**What is the minimum amount of information necessary to be shared between Bots to have a musical interaction?** A next step is determining the kind of information that should be shared between musebots. The MC is generating a constant click, which affords an acceptance over a common pulse: how that pulse is organised in time (i.e. the metre) is a basic parameter of which each musebot should be aware. However, where should this be determined? Sharing of pitch information is also natural, but should an underlying method of pitch organisation also be shared (i.e. a harmonic pattern)? What happens when conflicting information is generated? Lastly, how should form be determined? An accepted paradigm of improvised music is the evolutionary form produced by self-organisation resulting from autonomous agents (human or computer); however, EDM tends to display a more rigorous structure. How should this be determined?

**What relationship to human composition and performance should be incorporated?** Within the MuMe community, research has been undertaken to model human interaction within an improvisational ensemble of human performers (Blackwell *et al.* 2012). We suggest that musebots are not merely a "robot jam". To quote from the Call for Participation, "'human musicians having a jam' can make for a useful metaphor, but computers can do things differently, so we prefer not to fixate on that metaphor. Either way, getting software agents to work together requires thinking about how music is constructed, and working out shared paradigms for its automation."

**What aspects of the interaction can go beyond human performance modeling?** A great deal of what humans do in performance has been extremely difficult, if not impossible, to model. For example, simply tracking a beat is something we assume any musician can do with 100% accuracy, while computers are seldom better than 90% at this task. However, there are limits to human interaction, which computers can potentially overcome. For example, computers can share and negotiate plans, and thus exhibit a collective telepathic series of intentions. Young and Bown (2010) have offered some interesting possibilities for interaction between agents that could certainly be explored between musebots.

**What role should stylistic and aesthetic concerns play in formulating ensembles?** We imagine that in the future, musebots can query one another as to their stylistic proclivity, and generate interesting and unforeseen ensembles on their own. At the moment, the notion of human curation is still necessary. With only three musebots, the variety of musical output is obviously limited, but we imagine musebots being designed to produce specific stylistic traits. A

related question is how the musebots can, or should, deal with expectation: certain styles of EDM exhibit certain expectations in the listeners; while we acknowledge that we are not constrained to existing stylistic limitations, we are expecting humans to listen to, and hopefully appreciate, the generated music. Ignoring musical expectation outright is perhaps not the best strategy when offering a new paradigm in music-making.

**What steps would we need to take to make this a more intelligent system of interaction and/or coordination?** Many existing MuMe systems have already demonstrated musical intelligence in their abilities to self-organise, execute plans, and react appropriately to novel situations. However, the designers often rely upon ad hoc methodologies to produce idiosyncratic, non-idiomatic systems. How can such systems communicate their internal states efficiently, or is this even necessary?

**What are the emerging decisions that we would make about messaging? How could we categorise these and generalise them?** While audio analysis is one possible method for musebots to determine their environment, relying upon such analyses alone would take up huge amounts of processing cycles, without any guarantee as to an accurate cognitive conception of what is actually going on musically. Furthermore, given that each musebot would require its own complex audio processing module, the hardware demands would be inordinate. For this reason, having musebots simply tell other musebots what they are doing through messages seems much more efficient. However, how much information does a musebot need to broadcast about its current, or possibly future, state, in order for other musebots to interact with it musically?

**What is the furthest we could get with just "in the moment?"** From the above discussion, it is clear that an important concern for musebot ensembles is addressing the tensions that exist between self-organised generativity and coordinated, hierarchical musical structures. Clearly, 'in the moment' generation of musical materials is a trivial task for complex musical automata like the musebots described in this paper. A balance between autonomy on the one hand, and controlled, structural decisions will need to be carefully considered in the design of both musebots themselves, and their curation into musical ensembles. Ultimately, curatorial decisions surrounding style and musical aesthetic also go hand in hand with concerns regarding determinacy/indeterminacy in musical composition and performance, and we are excited to see how this ongoing tension will influence musebot designers and curators into the future.

## Conclusion

A primary goal in developing the musebot and musebot ensemble is to facilitate the exchange of ideas regarding how developers of musical metacreative systems can begin to collaborate, rather than continue to build individual idiosyncratic, non-idiomatic systems that rely upon ad hoc decisions. As we are targeting existing developers of MuMe and interactive systems, we recognize the variety of languages, tools, and approaches that are currently being used, and the reticence at adopting new frameworks that might inhibit established working methods. As such, our goal is to make the specification as easy as possible to wrap around new and existing systems and/or agents.

The specification uses a standard messaging system that can be incorporated within almost any language; however, we purposefully have not specified the messages themselves. Our intention is for these messages to evolve naturally, in response to the musical needs of developers. For example, through the use of machine- and human-readable info files, musebots and musebot developers can determine the messages a specific musebot receives and sends, while the open source specification allows for developers to propose new messages. Once these agents are performing together at a basic level, we feel that a community discussion will begin on the type of information that could, and should, be shared.

We have presented a description of our successful, albeit limited, first implementation of what we feel is an extremely exciting new paradigm for musical metacreation. Complex, autonomous musical producing systems are being presented successfully in concert, and the musebot platform is a viable method for these practitioners to collaborate creatively.

## References


Assayag, G., Bloch, G., Chemillier, M., Cont, A., and Dubnov, S. 2006. Omax brothers: a dynamic topology of agents for improvisation learning. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 125–132.

Blackwell, T., and Young, M. 2004. Self-organised music. In *Organised Sound*, 9(02): 123-126.

Blackwell, T., and Young, M. 2005. Live algorithms. In *Artificial Intelligence and Simulation of Behaviour Quarterly*, 122(7): 123.

Blackwell, T., Bown, O., and Young, M. 2012. Live Algorithms: towards autonomous computer improvisers. In *Computers and Creativity*, 147–174, Springer Berlin Heidelberg.

Bown, O., and Martin, A. 2012. Autonomy in music- generating systems. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference,* Palo Alto.

Bown, O., Eigenfeldt, A., Pasquier, P., Martin, A., and Carey, B. 2013. The Musical Metacreation Weekend: Challenges Arising from the Live Presentation of Musically Metacreative Systems. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, Boston.

Bown, O., Gemeinboeck, P., and Saunders, R. 2014. The Machine as Autonomous Performer. In *Interactive Experience in the Digital Age*, 75–90, Springer International Publishing.

Bown, O., Carey, B., and Eigenfeldt, A. 2015. Manifesto for a Musebot Ensemble: A Platform for Live Interactive Performance Between Multiple Autonomous Musical Agents. In *Proceedings of the International Symposium of Electronic Art 2015*, Vancouver.

Carey, B. 2012. Designing for Cumulative Interactivity: The _derivations System. In *12th International Conference on New Interfaces for Musical Expression*, Ann Arbor.

Charnley, J., Colton, S., and Llano, M. 2014. The FloWr Framework: Automated Flowchart Construction, Optimisation, Alteration for Creative Systems, in *Proceedings of the Fifth International Conference on Computational Creativity*, 315–323, Ljubljana.

Collins, N., and McLean, A. 2014. Algorave: A survey of the history, aesthetics and technology of live performance of algorithmic electronic dance music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, London.

Dahlstedt, P., and McBurney, P. 2006. Musical agents: Toward Computer-Aided Music Composition Using Autonomous Software Agents. *Leonardo*, 39(5): 469–470.

Diakopoulos, D., Vallis, O., Hochenbaum, J., Murphy, J., and Kapur, A. 2009. 21st Century Electronica: MIR Techniques for Classification and Performance In *International Society for Music Information Retrieval Conference*, Kobe, 465–469.

Downie, S. 2008. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. In *Acoustical Science and Technology*, 29(4): 247–255.

Eigenfeldt, A. 2007. Drum Circle: Intelligent Agents in Max/MSP. In *Proceedings of the 2007 International Computer Music Conference,* Copenhagen.

Eigenfeldt, A., and Pasquier, P. 2013. Evolving structures for electronic dance music. In *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, Amsterdam, 319–326, 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 Conference*, Boston.

Eigenfeldt, A. 2014. Generating Structure – Towards Large-scale Formal Generation. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, Raleigh.

Eldridge, A. 2007. *Collaborating with the behaving machine: simple adaptive dynamical systems for generative and interactive music*. PhD diss., University of Sussex.

Gimenes, M., Miranda, E., and Johnson, C. 2005. A Memetic Approach to the Evolution of Rhythms in a Society of Software Agents. In *Proceedings of the 10th Brazilian Symposium on Computer Music*, Belo Horizonte.

Lewis, G. 1999. Interacting with latter-day musical automata. In *Contemporary Music Review*, 18(3): 99–112.

Nierhaus, G. 2009. *Algorithmic composition: paradigms of automated music generation*, Springer Science & Business Media.

Rowe, R. 1992. Machine composing and listening with Cypher. In *Computer Music Journal*, 16(1): 43–63.

Rowe, R. 2004. *Machine musicianship*, MIT press.

Whitelaw, M. 2004. *Metacreation: art and artificial life*, MIT Press.

Wright, M. 1997. Open Sound Control-A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the 1997 International Computer Music Conference*, Thessaloniki, 101–104.

Yee-King, M. 2007. An automated music improviser using a genetic algorithm driven synthesis engine. In *Applications of Evolutionary Computing*, Springer Berlin Heidelberg, 567–576.

Young, M., and Bown, O. 2010. Clap-along: A negotiation strategy for creative musical interaction with computational systems. In *Proceedings of the International Conference on Computational Creativity*, Lisbon, Department of Informatics Engineering University of Coimbra.

# Generative Music for Live Musicians: *An Unnatural Selection*

**Arne Eigenfeldt**
School for the Contemporary Arts
Simon Fraser University
Vancouver, Canada
arne_e@sfu.ca

**Abstract**

*An Unnatural Selection* is a generative musical composition for conductor, eight live musicians, robotic percussion, and Disklavier. It was commissioned by Vancouver's Turning Point Ensemble, and premiered in May 2014. Music for its three movements is generated live: the melodic, harmonic, and rhythmic material is based upon analysis of supplied corpora. The traditionally notated music is displayed as a score for the conductor, and individual parts are sent to eight iPads for the musicians to sight-read. The entire system is autonomous (although it does reference a pre-made score), using evolutionary algorithms to develop musical material. Video of the performance is available online.[1] This paper describes the system used to create the work, and the heuristic decisions made in both the system design and the composition itself.

## Introduction

*An Unnatural Selection* can be classified as art as research: the author is a composer who has spent the previous thirty years coding software systems that are used as compositional assistants and/or partners. In the last ten years, these systems have explored greater autonomy, arguably creating computationally creative musical production systems that produce music that would be considered creative if the author had produced them independently.

Music has a long history of computational systems created by artist-programmers, in which many aspects of the musical creative process are automated (Chadabe 1980; Lewis 2000; Rowe 2004). Most of these systems have been idiosyncratic, non-idiomatic production systems specific to the artist's musical intention; however, some attempts have been made at evaluation (Eigenfeldt *et al*. 2012).

The author's own investigation into creative software have included multi-agent systems that emulate human improvisational practices (Eigenfeldt 2006), constrained Markov selection (Eigenfeldt and Pasquier 2010), and corpus-based recombination (Eigenfeldt 2012). All of these systems operate in real-time, in that they generate their output in performance using commercially available synthesizers, which, unfortunately, offer limited representations of their highly complex acoustic models (Risset and Matthews 1969, Grey and Moorer 1977)

In order to bypass these audio limitations, the author's more recent research investigates the potential for generating music directly for live performers (Eigenfeldt and Pasquier 2012b). Complex issues arise when generating music for humans, both in terms of software engineering – e.g. producing complex musical notation for individual performers – and human computer interaction: asking musicians to read music for the first time during the performance, without rehearsal, and without recourse to improvisation. See Eigenfeldt (2014) for a detailed discussion of these matters.

## Previous Work

*An Unnatural Selection* builds upon the work of others in several areas, including genetic algorithms, real-time notation, and generative music.

### Evolutionary Algorithms

Evolutionary computation has been used within music for over two decades in various ways. Todd and Werner (1999) provide a good overview of the earlier musical explorations using such approaches, while Miranda and Biles (2007) provide a more recent survey. Very few of these approaches have been compositional in nature; instead, their foci have tended to be studies, rather than the generation of complete musical compositions.

Several real-time applications of GAs have been used, including Weinberg *et al*. (2008), which selected individuals from an Interactive Genetic Algorithm (IGA) suitable for the immediate situation within a real-time improvisation. Another approach (Beyls 2009) used a fitness function that sought either similar or contrasting individuals to an immediate situation within an improvisation.

Waschka (2007) used a GA to generate contemporary art music. His explanation of the relationship of time within music is fundamental to understanding the potential for evolutionary algorithms within art-music: "unlike material objects, including some works of art, music is time-based. The changes heard in a piece over its duration and how those changes are handled can be the most important aspect of a work." Waschka's *GenDash* has several important attributes, a number of which are unusual: an individual is a measure of music; all individuals in all genera-

[1] https://aeigenfeldt.wordpress.com/works/music-for-robots-and-humans/

tions are performed; the fitness function is random, leading to random selection; the composer chooses the initial population. Of note is the second stated attribute, the result of which is that "the evolutionary process itself, not the result of a particular number of iterations, constituted the music". Waschka provides some justifications for his heuristic choices, suggesting that while they may not be observed in real-world compositional process, they do provide musically useful results.

EAs have been used successfully in experimental music and improvisation for several years. In most cases, artists have been able to overcome the main difficulty in applying such techniques to music – namely the difficulty of formulating an effective aesthetic fitness function – through a variety of heuristic methods. One particularly attractive feature of EAs to composers relates to the notion of musical development – the evolution of musical ideas over time – and its relationship to biological evolution. As music is a time-based art, the presentation of successive generations – rather than only the final generation – allows for the aural exposition of evolving musical ideas.

## Real-time Notation

The prospect of generating real-time notation is an established area of musical research, and has been approached from a variety of viewpoints: see Hajdu and Didkovsky (2009) for a general overview. Freeman (2010) has approached it as an opportunity for new collaborative paradigms of musical creativity, while Gutknecht *et al*. (2005) explored its potential for controlled improvisation. Kim-Boyle (2006) investigated open-form scores, and McClelland and Acorn (2008) studied composer-performer interactions. However, the complexity of musical notation (Stone 1980), limited these efforts to graphic representations, rather than traditional western music notation that affords more precise and detailed directions to performers.

Hajdu's Quintet.net (2005) was an initial implementation of MaxScore (Didkovsky and Hajdu 2009), a publically available software package for the generation of standard western musical notation, one that allows for complexities of notation on the level of offline notation programs. *An Unnatural Selection* uses MaxScore for the generation of the conductor's score, which is then parsed to individual iPads and custom coded software.

## Production Systems versus Compositions

The creation of a production system for *An Unnatural Selection* was concurrent with the conceptualization of the composition itself, which is often the case in the author's practice. The desired musical results are imagined through audiation, and the software is coded with these results in mind. The attraction of generativity rests in the ability for a musical work to be actuated in varying forms while still retaining some form of overall artistic control.

The author has chosen to create composition-specific, rather than general purpose, systems for two reasons: previous experience has shown that general systems tend to become so complex with added features as to obfuscate any purposeful artistic use, and secondly, specifically designed systems allow for a design with a singular artistic output in mind.

As a result, some modules within the system used in *An Unnatural Selection* are specific to that work; however, it also builds upon earlier work (Eigenfeldt and Pasquier 2010) as well as contributing to successive works. Specifically, the analysis engine and generation engine can be considered a free- standing system, which I refer to as *PAT* (Probabilities and Tendencies); the evolutionary aspects are specific to *An Unnatural Selection*.

## The GA and its role as "Development tool"

As already mentioned, the use of genetic algorithms – modified or otherwise – are attractive to composers interested in musical development. While this method of composition has its roots in the Germanic tradition of the 18th and 19th centuries, it remains cognitively useful, since it provides listeners with a method of understanding the unfolding of music over time (Deliege 1996; Deliege *et al*. 1996). A description of the work from the program notes – "musical ideas are born, are passed on to new generations and evolved, and eventually die out, replaced by new ideas" – may suggest principles of artificial life, or music based upon Brahms, Mahler, or Schoenberg.

A general conception of the first movement was a progression from chaos to order;

- an initial population of eight musical phrases are presented concurrently by the eight instrumentalists;
- the phrases are repeated, and each repetition develops the phrases independently;
- segments from the individual phrases infiltrate one another;
- the individual phrases separate in time, thus allowing their clearer perception by the listener.

While these concepts began with a musical aesthetic in mind, they were clearly influenced by their potential inclusion of genetic algorithms.

## The Score as template

*An Unnatural Selection* is the most developed system in my pursuit of real-time composition (Eigenfeldt 2011): the possibility to control multiple complex gestures during performance. As will be described, *An Unnatural Selection* involves a number of high-level parameter variables that determines how the system generates and evolves individuals; dynamically controlling these in performance effectively shapes the music. As the performance approached, I doubted my performative abilities, and instantiated a score-based system that allowed for the pre-determined setting of the control parameters for each successive generation: while the details of work would still be left to the system, the overall shape would be preset. The use of such templates is not uncommon in other computationally creative media: Colton *et al*. (2012) used similar design constraints in generating poetry in order to maintain formal patterns.

## Probabilities and Tendencies (PAT)

The heart of *PAT* rests in its ability to derive generative rules through the analysis of supplied corpora. Cope (1987) was the first composer to investigate the potential for style modeling within music; his Experiments in Musical Intelligence generated many compositions in the style of Bach, Mozart, Gershwin, and Cope. Dubnov *et al*. (2003) suggest that statistical approaches to style modeling "capture some of the statistical redundancies without explicitly modeling the higher-level abstractions", which allow for the possibility of generating "new instances of musical sequences that reflect an explicit musical style". However, their goals were more general in that composition was only one of many possible suggested outcomes from their initial work. Dubnov's later work has focused upon machine improvisation (Assayag *et al*. 2010).

The concept of style extraction for reasons other than artistic creation has been researched more recently by Collins (2011), who tentatively suggested that, given the state of current research, it *may* be possible to successfully generate compositions within a style, given an existing database.

For *An Unnatural Selection*, corpora included compositions by the following composers:

• Movement I: 19 compositions by Pat Metheny
• Movement II: 2 compositions by Pat Metheny and 2 by Arvo Pärt
• Movement III: 1 composition by Terry Riley and 2 by Pat Metheny

These specific selections were arrived at through trial and error, as well as aesthetic consideration. The contemporary jazz material of Metheny provided harmonic richness without the functional tonality of the 19th century. Combining this corpus with Pärt's simpler harmonies and melodies gave them an interesting new dimension, while the repetitive melodic material of Riley's *In C*, when combined with Metheny's harmonies created a new interpretation of minimalist melodic and rhythmic repetition with more complex harmonic underpinnings.

### Analysis of corpora

*PAT* requires specially prepared MIDI files that consist of a quantized monophonic melody in one channel, and quantized harmonic data in another: essentially, a lead-sheet representation of the music. Prior to the creation of melodic, harmonic, and rhythmic n-gram dictionaries (Pearce and Wiggins 2004), harmonic data is parsed into pitch-class sets (Forte 1973). Melodic data is stored in reference to the harmonic set within which it was found, both as an actual MIDI note number and pitch-class as relative to the set.

### Representation

Music representation, and its problematic nature, has been thoroughly researched: Dannenburg (1993) gives an excellent overview of the issues involved. Event-lists are the standard method of symbolic representation currently used, as they supply the minimally required information for representing music within a note-based paradigm. However, since event-lists do not capture relationships between events, they have proven problematic for generative purposes (Maxwell 2014). For this reason, *PAT* includes non-events that are displayed in music notation.

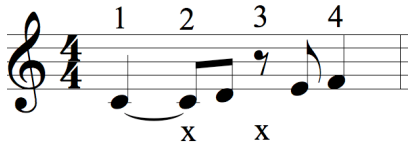



**Figure 1.** A notated melodic phrase, with beats 1 through 4 indicated, and non-events marked below.

Figure 1 presents a simple melodic phrase, and its event-based representation is shown in Table 1. While the onset times and durations are captured, their interrelationships, clearly shown in Figure 1, are difficult to determine. The initial event's prolongation into the second beat, as shown through the tie (marked with an x), is missing. Similarly, the rest on the third beat (also marked with an x) segments the second and third beats, also not obvious in Table 1.

| Event # | Beat | Pitch | Duration |
|---|---|---|---|
| 1 | 1.0 | 60 | 1.5 |
| 2 | 2.5 | 62 | 0.5 |
| 3 | 3.5 | 64 | 0.5 |
| 4 | 4.0 | 65 | 1.0 |

**Table 1.** The music of Fig. 1, represented as events.

The solution in *PAT* is to include all non-events: rests are represented as pitch 0 with appropriate durations, and ties are represented as incoming pitches with negative durations: see Table 2.

| Event | Beat | Pitch | Duration |
|---|---|---|---|
| 1 | 1.0 | 60 | 1.5 |
| 2 | 2.0 | 60 | -0.5 |
| 3 | 2.5 | 62 | 0.5 |
| 4 | 3.0 | 0 | 0.5 |
| 5 | 3.5 | 64 | 0.5 |
| 6 | 4. | 65 | 1.0 |

**Table 2.** The music of Fig. 1, showing the "non-events" 2 and 4.

Associations between events are retained within *PAT* through encoding by beat. As the generative engine uses Markov chains, the important relationships within and between beats are preserved through separate pitch and rhythm/duration n-gram dictionaries.

**Rhythm** Events are stored as onset/duration duples, grouped into beats, with onset times indicating offset into the beat. Thus, Figure 1, segmented into individual beats, is initially represented as:

(0.0 1.5)
(0.0 -0.5) (0.5 0.5)
(0.0 0.5) (0.5 0.5)
(0.0 1.0)

Each beat, as a duple or combination of duples, serves as an index into the rhythm n-gram dictionary, which stores all continuations and the number of times a continuation has been found. Thus, after encoding only Figure 1, the rhythm dictionary would consist of the following:

(0.0 1.5)
(0.0 -0.5) (0.5 0.5) **1**
(0.0 -0.5)(0.5 0.5)
(0.0 0.5)(0.5 0.5) **1**
(0.0 0.5)(0.5 0.5)
(0.0 1.0) **1**

**Pitch** Melodic events are stored in relation to the harmonic set within which they occurred. The total number of occurrences of each pitch-class (PC), relative to the set, are stored, as well as PCs that are determined to begin phrases (initial PCs) and end phrases (terminal PCs). Lastly, an n-gram for the continuation of each PC (n>) is stored, along with an n-gram of its originating PC (>n).

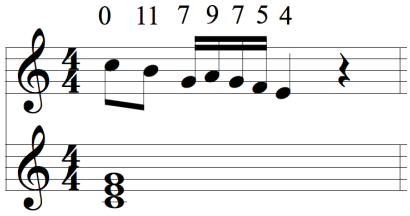



**Figure 2.** A melodic phrase with accompanying harmony; pitch-classes are indicated.

Thus, given the melodic and harmonic material of Figure 2, the melodic dictionary shown in Table 3 is constructed. Note that separate contour arrays are kept so as to retain actual melodic shapes.

| **Set:** | **0** | **4** | **7** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***Pitch Class*** | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* |
| **Total PCs:** | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 0 | 1 | 0 | 1 |
| **Initial:** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Terminal:** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0>** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| **5>** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **7>** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| **9>** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **11>** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **>4** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **>5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **>7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| **>9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **>11** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 3.** The music of Fig. 2, storing individual PC's movement to (n>) and from (>n), as well as a count of overall PCs for the set, and which PCs initiated and terminated phrases.

A similar system is used for harmony, with the n-gram storing the relative root movement of each set. Lastly, as well as melodic contours, an array of root movements (bass lines) is also kept. In both cases, these contours are normalized and their length's scaled. New contours are compared to those existing using a Euclidean distance function, and those below a user-set minimum similarity level are culled, in order to avoid excessive similarity.

## Generation

The generative and evolutionary algorithms within *An Unnatural Selection* utilize user-set parameters that define how the algorithms function; it is the dynamic control of these parameters over time that shapes the music. As has been mentioned, *An Unnatural Selection* employs a parameter score to control these values.

### Evolutionary Methods in *An Unnatural Selection*

*An Unnatural Selection* uses the architecture of *PAT* within a modified evolutionary system. Within this system, musical phrases operate as individuals, or phenotypes, and individual beats – a combination of rhythmic and melodic material – operate as chromosomes. Phrases are developed in such ways that they represent successive generations. Since all individuals pass to the next generation, there is no selection, and thus no fitness function; however, each individual experiences significant crossover and mutation. Several independent populations exist simultaneously.

The use of evolutionary methods are extremely heuristic; earlier uses of such techniques by the author are documented elsewhere (Eigenfeldt 2012; Eigenfeldt and Pasquier 2012a).

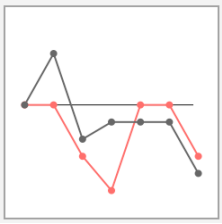


**Figure 3.** A root progression request (red), and the generated progression based upon possible continuations (grey).

### Generating Harmonic Progressions

A harmonic progression is the first generated element. A root progression is selected from the database as a target, and scaled by the requested number of chords in the progression. An initial chord is then selected from those sets that initiated phrases, and its continuations are compared to the next interval in the target. A Gaussian selection is then made from the highest probabilities. This process continues until a phrase progression is generated (see Figure 3). At this point, the progression has not been assigned individual durations.

### Generating Phrases/Individuals

A number of required parameter values are calculated through a combination of corpus data and user-set ranges. For example, in order to select a phrase length for an individual, the actual phrase lengths from the corpus are ordered, and a value is sampled from this list from within a

user-set range (in this case *phraseLengthRange*). Thus, if this range is fixed between 0.9 and 1.0, a random selection will be made from 10% of the corpus' longest phrase lengths.

Individual phrases are assigned to specific instruments; since *An Unnatural Selection* was composed for eight instrument, Disklavier, and robotic percussion, the population consisted of a maximum of 12 individuals (the piano and percussion used two independent phrases). An important user parameter is whether the instrument (and thus the phrase) is considered *foreground* or *background*: in the case of the former, rhythmic data is selected from the corpus based upon density, while in the latter, data is selected based upon complexity (syncopation). Foreground individuals are deemed to be more active and have more variation; background individuals are either more repetitive or of longer duration, as set by a user parameter.

**Foreground** The number of onsets per beat is determined by a user parameter, *phraseDensityRange*. At initialization, the corpus' average beat density is scaled between 0.0 (the least dense) to 1.0 (the most dense), and a selection is made within the user range.

**Background** At initialization, the corpus' onsets are also rated for complexity: the relative amount of syncopation within each beat. Background phrases are comprised of either rhythmic material or held notes; in the case of the former, an exponential selection is made from the top 1/3 of the corpora (the most syncopated), while a similar selection is made from the bottom 1/3 for held individuals. Background phrases are immediately repeated if they are less than one measure in total duration.

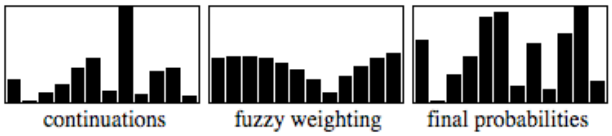



**Figure 4.** The continuations for a specific PC (7), left; a weighting that favors more distant PCs, center; the final probability for PC selection, right.

Once an initial selection is made for foreground or background individuals, the continuations from that beat are constrained by the same user parameters.

**Melodic material** Similar to harmonic and rhythmic generation, melodic generation selects an initial PC from those PCs in the corpus that began melodic phrases; continuations of that PC are then weighted to derive the probabilities for the next PC. In the case of foreground individuals, a fuzzy weighting is applied so as to avoid direct repetition and small intervals. (see Figure 4); for background phrases, the opposite weighting is applied to avoid large melodic leaps.

**Individual locations within overall phrase** Once all phrases have been generated, the maximum length is determined, in beats; this value is rounded up to the next measure, and becomes the overall phrase length to which the harmonic progression is overlaid.

Individuals are placed within the overall phrase, either attempting to converge upon other individual's locations, or diverge from them, depending upon a user-set parameter *phraseVersusPhrase*. Each phrase's current onset locations are summed, which will determine the probability for the placement of individuals in the next overall phrase while the inverse will provide probabilities for divergence (see Figure 5). Rests are added to the beginning and/or end of the individual in order to place them in the overall phrase: these rests are not considered part of the individual.

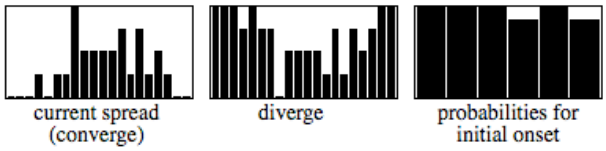



**Figure 5.** The number of total onsets per beat, left; the inverse as avoidance probability, center; the final probability for phrase starts, right. Because of the individual's length, its placement is limited to the first six locations of the overall phrase.

**Melodic Quantization**

With the harmony now in place, PCs are quantized to sets within which they are located. A PC is compared to the total n-gram for its current harmonic set, which acts as an overall probability function, scaled by intervallic closeness to the PC (see Figure 6). In this way, PCs are not forced to a pre-defined "chord-scale" for the set, but adjusted to fit the n-gram for the set within the corpus.

Pitch ranges are then adjusted for each individual, and dynamics, articulations, slurs, and special text (i.e. *arco* vs. *pizzicato*) are applied: space does not allow for a discussion of how these parameters are determined.

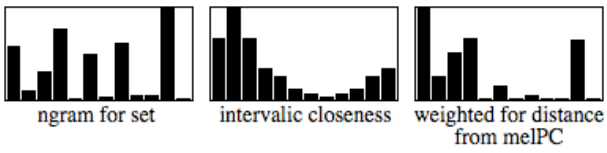



**Figure 6.** The n-gram for the set (0 3 7 10), left; a weighting for a raw PC (1) that favors intervallic closeness; the final probability for PC quantization, right.

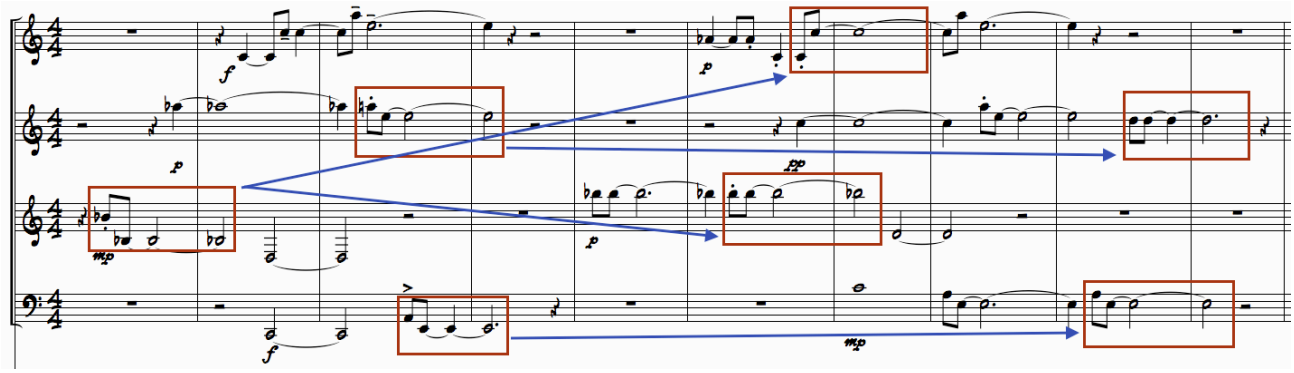


**Figure 7.** The first two generations of a population of four individuals, demonstrating crossover by segment.

## Evolving Populations

As mentioned previously, all individuals progress to the next generation, unless they are turned off in the user score. Evolution of individuals includes crossover (within set populations) and mutation.

**Crossover** The individual's chromosomes are its beats; as rests are considered events within *PAT*, every beat, including rests, constitutes a separate chromosome. Crossover does not involve the usual splicing of two individuals, but instead the insertion or deletion of *musical segments* between individuals. Segmentation is done using standard perceptual cues, including pitch leaps, rests, and held notes (Cambouropoulos 2001), resulting in segments of one to several beats (see Figure 7).

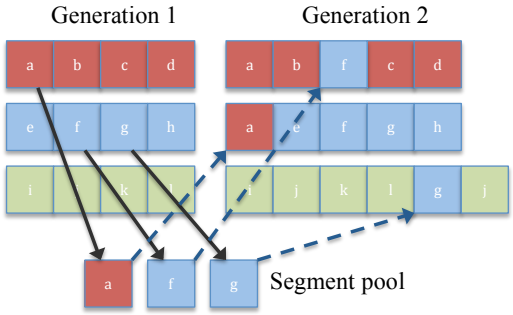



**Figure 8.** Two generations of three individuals (red, blue, green), showing expansion through crossover of segments. Segments **a**, **f**, and **g** are copied to the segment pool, potentially mutated, then inserted into other individuals

Individuals will either expand or contract during crossover, depending upon a user-set parameter. Contracting an individual involves deleting a segment, and splicing together the remaining parts in a musically intelligent way. Expansion involves copying segments from different individuals into a separate pool that contains a maximum of 16 segments, differentiated by individual type: foreground versus background (see Figure 8). Segments are potentially mutated (see next section), then inserted into individuals.

**Mutation** Mutation can occur on segments within the segment pool prior to insertion, or on the entire individual, depending upon the user-set parameter *multiBeatProbability*. Mutations are musically useful variations, including:

- scramble – randomly scramble the pitch-classes;
- transpose – transpose a segment up or down by a fixed amount, from 2 pitch-classes to 12;
- sort+ - sort the pitch-classes from lowest to highest;
- sort– - sort the pitch-classes from highest to lowest;
- rest for notes – substitute rests for pitch-classes, to a maximum of 50% of the onsets in the segment.

The type of mutation is selected using a roulette-wheel selection method from user-set probability weightings for each type.

## Logistics

*An Unnatural Selection* is coded in MaxMSP[2], using MaxScore for notational display. Custom software was written to display individual parts on iPads, which received JMSL (Didkovsky and Burke 2001) data wirelessly over a TCP network. The generative software composes several phrases in advance, and sends the MIDI data to Ableton Live[3] for performance (specifically the Disklavier and robotic percussion); Ableton Live provides a click track for the conductor, and sends messages back to the generative system requesting new material.

## Discussion

*An Unnatural Selection* is, first and foremost, an artistic system designed to create multiple versions of a specific composition – the author's interpretation of "generative music". Many aspects of the system's development – for example, the multiple populations – were arrived at through artistic reasons, rather than scientific. Algorithms were adjusted and parameters "tweaked" through many hours of listening to the system's output; as a result, heuristics form an important aspect of the final software.

Whether the system is computationally creative is a more difficult matter to determine. While I echo Cope's desire that "what matters most is the music" (Cope 2005), I am fully aware of Wiggins reservations that "with hand-coded rules of whatever kind, we can never get away from

[2] www.cycling74.com/
[3] www.ableton.com/

the claim that the creativity is coming from the programmer and not the program" (Wiggins 2008).

The overriding design aspect entailed musical production rules derived through analysis of a corpus; however, as I discuss elsewhere (Eigenfeldt 2013), how this data is interpreted is itself a heuristic decision, especially when being used to create an artwork of any value.

## Evaluation

While the intention of *An Unnatural Selection* was primarily artistic, the notion of evaluation was not entirely ignored, an issue the author has attempted to broach previously (Eigenfeldt *et al*. 2012). The work was clearly experimental: it would have been much easier to generate the music offline, and select the best examples of the system, allowing the musicians to rehearse and perform these in ways in which they are accustomed. However, the fact that the music was generated live was an integral element to the performance: in fact, interactive lighting was used in which the musician's chairs were lit only while they played, an effort to underline the real-time aspect.

While no formal evaluation studies were done, the musicians were asked to critically comment upon their experiences. Their comments are summarized here.

**Limited Complexity in Structure** Some musicians commented on the relatively unsophisticated nature of the overall form of the generated music:

> *"I didn't sense a strong structural aspect to the pieces. I thought the program generated some interesting ideas but I would like to see more juxtaposition, contrast of elements, in order to create more variety and interest."*
>
> *"I would venture to say… that the music… certainly wasn't as developed or thoughtful as something that a seasoned, professional composer would create."*
>
> *"…any of the versions would likely have struck me as somewhat interesting but fairly basic."*

Generating convincing structure is an open problem in musical metacreation, which is not surprising, as it is one of the most difficult elements to teach young composers.

**More Overall Complexity** When asked for specific suggestions, several musicians provided very musical suggestions, including a greater variety of time signatures, more subtle instrumentation and playing techniques, different groupings of musicians, *accelerando* and *rubato*. Many of these aspects can, and will be incorporated into future versions of the system.

**Positive comments** Keeping in mind that these are professional musicians specializing in contemporary music performance, I was happy to receive positive comments:

> *"I assume the software is going to continue to grow and become more accomplished through further exposure to, and analysis of, sophisticated compositional ideas."*
>
> *"I thought some of music was beautiful, especially in the second movement."*
>
> *"It seems to me that what you are doing is groundbreaking and interesting, even if still at a relatively primitive stage."*

## Conclusion

*An Unnatural Selection* was the culmination of my research into generating music in real-time for live musicians. Upon reflection after the fact, my goal was to present musical notation to the performers that was as close as possible to what they were used to, since no improvisation would be expected. Naturally, this would necessitate having the musicians perform the music without any rehearsal – and extremely demanding request. While the extended rehearsals did allow the musicians to gain some expectations of what to expect from the software, it failed to provide them with what rehearsals usually provide: a time to discover the required interactions inherent within the music. One musician suggested that these indications, normally learned during rehearsal periods, could somehow appear in the notation:

> *"Maybe the screen could indicate to the players when they have an important theme to bring out, and also indicate which instrument they are in a dialogue with or have the same rhythmic figure as?"*

Future versions of the system will explore this new paradigm, which also suggests the potential to involve the performers within the generative composition in ways that would not be possible without intelligent technology.

## Acknowledgements

This research was undertaken through a Research/Creation grant from Canada's Social Science and Humanities Research Council (SSHRC). Thanks to the Generative Media Research Group as well as the Metacreation Lab for their support and suggestions during the creation of this work. Particular thanks goes to James Maxwell for his thought-provoking work on cognitive music generative systems.

## References

Assayag, G., Bloch, G., Cont, A., & Dubnov, S. 2010. Interaction with Machine Improvisation. *The Structure of Style*, 219–245.

Beyls, P. 2009. Interactive Composing as the Expression of Autonomous Machine Motivations. *International Computer Music Conference (ICMC)*, Montreal, 267–74.

Chadabe, J. 1980. Solo: A Specific Example of Realtime Performance. *Computer Music-Report on an International Project. Canadian Commission for UNESCO*.

Cambouropoulos, E. 2001. Melodic cue abstraction, similarity, and category formation: A formal model. *Music Perception*, 18(3), 347–370.

Collins, T. 2011. "Improved methods for pattern discovery in music, with applications in automated stylistic composition," *PhD thesis,* Faculty of Mathematics, Computing and Technology, The Open University.

Colton, S., Goodwin, J., & Veale, T. 2012. Full face poetry generation. *International Conference on Computational Creativity (ICCC),* Dublin, 95–10.

Cope, D. 1987 An Expert System for Computer-Assisted Composition. *Computer Music Journal*, 11(4), 30–46.

Cope, D. 2005. *Computer models of musical creativity,* Cambridge: MIT Press.

Dannenberg, R. 1993. Music representation issues, techniques, and systems. *Computer Music Journal*, 17(3), 20–30.

Deliege, I. 1996. Cue abstraction as a component of categorisation processes in music listening. *Psychology of Music*, 24(2), 131–156.

Deliège, I., Mélen, M., Stammers, D., and Cross, I. 1996. Musical schemata in real-time listening to a piece of music. *Music Perception*, 117–159.

Didkovsky, N., Burk, P. 2001. Java Music Specification Language, an introduction and overview. *ICMC,* Havana, 123–126.

Dubnov, S., Assayag, G., Lartillot, O., & Bejerano, G. 2003. "Using machine-learning methods for musical style modeling," *Computer*, 36(10), 73–80.

Eigenfeldt, A. 2006. Kinetic Engine: toward an intelligent improvising instrument. *Sound and Music Computing Conference (SMC)*, Marseilles, 97–100.

Eigenfeldt, A., Pasquier, P. 2010. Realtime generation of harmonic progressions using controlled Markov selection. *ICCC*, Lisbon, 16–25.

Eigenfeldt, A. 2011. Real-time composition as performance ecosystem. *Organised Sound*, 16(02), 145–153.

Eigenfeldt, A. 2012. Corpus-based recombinant composition using a genetic algorithm. *Soft Computing,* 16(12), 2049–2056.

Eigenfeldt, A., Pasquier, P. 2012a. Populations of Populations - Composing with Multiple Evolutionary Algorithms, P. Machado, J. Romero, and A. Carballal (Eds.). *EvoMUSART 2012*, LNCS 7247, 72–83.

Eigenfeldt, A., Pasquier, P. 2012b. Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. *SMC*, Copenhagen, 181–186.

Eigenfeldt, A., Burnett, A., & Pasquier, P. 2012. Evaluating musical metacreation in a live performance context. *ICCC,* Dublin, 140–144.

Eigenfeldt, A. 2013. The Human Fingerprint in Machine-Generated Music. *xCoAx: Computation, Communication, Aesthetics, and X*, Bergamo, 107–115

Eigenfeldt, A. 2014. Generative Music for Live Performance: Experiences with real-time notation. *Organised Sound*, 19(3).

Forte, A. 1973. *The structure of atonal music*. Yale University Press.

Freeman, J. 2010. Web-based collaboration, live musical performance and open-form scores. *International Journal of Performance Arts & Digital Media*, 6(2), 149–170.

Grey, J., Moorer, J. 1977. Perceptual evaluations of synthesized musical instrument tones. *The Journal of the Acoustical Society of America*, 62(2), 454–462.

Gutknecht, J., Clay, A., & Frey, T. 2005. GoingPublik: using realtime global score synthesis. *New Interfaces for Musical Expression,* Singapore, 148–151.

Hajdu, G. 2005. Quintet.net: An environment for composing and performing music on the internet. *Leonardo*, 38(1), 23–30.

Hajdu, G., Didkovsky, N. 2009. On the Evolution of Music Notation in Network Music Environments, *Contemporary Music Review*, 28(4-5), 395–407.

Kim-Boyle, D. 2006. Real time generation of open form scores. *Proceedings of Digital Art Weeks*, ETH Zurich.

Lewis, G. 2000. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10, 33–39.

McClelland, C., Alcorn, M. 2008. Exploring new composer/performer interactions using real-time notation. *ICMC,* Belfast, 176–179.

Maxwell, J. 2014. Generative Music, Cognitive Modelling, and Computer-Assisted Composition in MusiCog and ManuScore. *PhD Thesis,* Simon Fraser University.

Miranda, E., Biles, J. eds. 2007. *Evolutionary Computer Music*. London: Springer.

Pearce, M., Wiggins, G. 2004. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4), 367–385.

Risset, J., Mathews, M. 1969. Analysis of musical instrument tones. *Physics today*, 22(2), 23–30.

Rowe, R. 2004. *Machine musicianship*. MIT press.

Stone, K. 1980. Music Notation in the Twentieth Century. W.W. Norton, New York.

Todd, P., Werner, G. 1999. Frankensteinian methods for evolutionary music composition. *Musical networks: Parallel distributed perception and performance*, Griffith, N., Todd, P., eds., Cambridge, MA, 313–339.

Waschka, R. 2007. Composing with Genetic Algorithms: GenDash. *Evolutionary Computer Music*, Springer, London, 117–136.

Weinberg, G., Godfrey, M., Rae, A., & Rhoads, J. 2008. A Real-Time Genetic Algorithm in Human-Robot Musical Improvisation. *Computer Music Modeling and Retrieval, Sense of Sounds*, Springer, Berlin, 351–359.

Wiggins, G. 2008. Computer Models of Musical Creativity: A Review of Computer Models of Musical Creativity by David Cope, *Literary and Linguistic Computing,* 10(1), 109–116.

# Generalize and Blend: Concept Blending Based on Generalization, Analogy, and Amalgams

**Tarek R. Besold**
Institute of Cognitive Science
University of Osnabrück
D-49069 Osnabrück, Germany
tarek.besold@uni-osnabrueck.de

**Enric Plaza**
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus U.A.B., 08193 Bellaterra, Catalonia (Spain)
enric@iiia.csic.es

**Abstract**

Concept blending, a cognitive process which allows for the combination of certain elements (and their relations) from originally distinct conceptual spaces into a new unified space combining these previously separate elements and allowing the performance of reasoning and inference over the combination, is taken as a key element of creative thought and combinatorial creativity. In this paper, we provide an intermediate report on work towards the development of a computational-level and algorithmic-level account of concept blending, presenting the theoretical background together with the main model characteristics, as well as two case studies.

## Creativity and Concept Blending

The term "combinatorial creativity" (Boden 2003) refers to creativity which arises from a combinatorial process joining familiar ideas (in the form of, for instance, concepts, theories, or artworks) in an unfamiliar way, thereby producing novel ideas. But although the overall idea of combining preexisting ideas into new ones seems fairly intuitive and straightforward, computationally modeling this form of creativity turns out to be surprisingly complicated: When looking at it from a more formal perspective at the current stage neither can a precise algorithmic characterization be given, nor are at least the details of a possible computational-level theory describing the process(es) at work well understood.

Still, in recent years a proposal by (Fauconnier and Turner 1998) called concept blending (or conceptual integration) has influenced and reinvigorated studies trying to unravel the general cognitive principles operating during creative thought. In their theory, concept blending constitutes a cognitive process which allows for the combination of certain elements (and their relations) from originally distinct conceptual spaces into a new unified space combining these previously separate elements and allowing the performance of reasoning and inference over the combination.

Unfortunately, a proper computational modeling of concept blending as cognitive capacity again is lacking. Neither do (Fauconnier and Turner 1998) provide a fully worked out and formalized theory themselves, nor does their informal account capture key properties and functionalities as, for example, the retrieval of input spaces, the selection and transfer of elements from the input into the blend space, or the further combination of possibly mutually contradictory elements in the blend.

These shortcomings notwithstanding, several researchers in AI and computational cognitive modeling have used the provided conceptual descriptions as a starting point for proposing possible refinements and implementations: (Goguen and Harrell 2010) propose a concept blending-based approach to the analysis of the style of multimedia content in terms of blending principles and also provide an experimental implementation, (Pereira 2007) tries to develop a computationally plausible model of several hypothesized sub-parts of concept blending, (Thagard and Stewart 2011) exemplify how creative thinking could arise from using convolution to combine neural patterns into ones which are potentially novel and useful, and (Veale and O'Donoghue 2000) present their computational model of conceptual integration and propose several extensions to the (at that time prevailing) view on concept blending.

Since 2013, another attempt at developing a computationally feasible, cognitively-inspired formal model of concept creation, grounded on a sound mathematical theory of concepts and implemented in a generic, creative computational system is undertaken by a European research consortium in the so called Concept Invention Theory (COINVENT) project (Schorlemmer et al. 2014)[1]. One of the main goals of the COINVENT research program is the development of a computational-level and algorithmic-level account of concept blending based on insights from psychology, AI, and cognitive modeling, the heart of which are made up by results from cognitive systems studies on computational analogy-making and knowledge transfer and combination (i.e., the computation of so called *amalgams*) from case-based reasoning. In the following we present an analogy-inspired perspective on the COINVENT core model for concept blending and show how the respective mechanisms and systems interact.

## Two Mechanisms at the Heart of COINVENT: Generalization-Based Analogy and Amalgams

As analogy seems to play a crucial role in human cognition (Gentner and Smith 2013), researchers on the computa-

[1] Also see `http://www.coinvent-project.eu` for details on the consortium and the project.

tional side of cognitive science and in AI also very quickly got interested in the topic and have been creating computational models of analogy-making since the advent of computer systems, among others giving rise to (Winston 1980)'s work on analogy and learning, (Hofstadter and Mitchell 1994)'s Copycat system, or (Falkenhainer, Forbus, and Gentner 1989)'s well-known Structure-Mapping Engine (SME).

Generally speaking there are (at least) two families of computational analogy models: one family is based on a (generalization-free) direct mapping approach, the other one relies on a two-step procedure with a generalization stage followed by a subsequent mapping stage. While the former type of analogy engine is, among others, exemplified in the SME and its immediate pairwise mapping of domain elements between elements of source and target of the potential analogy, followed by the accumulation of individual mappings into more complex structures, the latter category is represented by the Heuristic-Driven Theory Projection (HDTP) framework (Schmidt et al. 2014). As COINVENT, for principled conceptual reasons (see the section on the idea(s) behind concept blending in COINVENT below), relies on the generalization-based view on analogy-making, we shortly introduce this model category in the following subsection.

In a conceptually related, but mostly independently conducted line of work researchers in case-based reasoning (CBR) have been trying to develop problem solving methodologies based on the principle that similar problems tend to have similar solutions. CBR tries to solve problems by retrieving one or several cases relevant for the issue at hand from a case-base with already solved previous problems (*cases*), and then reusing the past case(s) to also solve the new task (Aamodt and Plaza 1994). While the retrieval stage has received significant attention over the last two decades, the transfer and combination of knowledge from the retrieved case to the current problem has been studied in an domain-specific way, with (Ontanón and Plaza 2012) being a recent attempt at also gaining insights on this phase of the CBR cycle by suggesting the framework of amalgams (Ontanón and Plaza 2010) as a formal model for reuse of multiple cases. The second subsection gives an overview of amalgams as used in COINVENT.

## Generalization-Based Models of Analogy

Generalization-based models of analogy-making share a close conceptual connection to models of inductive generalization (Smaling 2003). Similar to these, the basic principle is the recognition of a common core between source and target of the potential analogy, which is then used for guiding the formation process of the analogy and the subsequent content transfer and reasoning steps. Fig. 1 gives a schematic overview: The common conceptual elements between source $S$ and target $T$ correspond to a shared generalization $G$ (subsuming both, $S$ and $T$), which also induces mappings between the respective domain elements, establishing an analogical relation. These mappings, governed by the generalization, then also subsequently define how (previously unmatched) knowledge from the source domain can be transferred to and integrated into the target domain, namely by converting elements from $S$ into their corresponding counterparts within $T$.

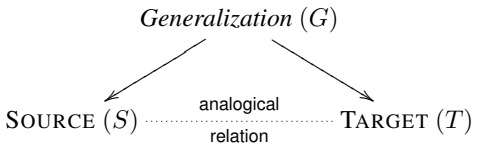



Figure 1: A schematic overview of a generalization-based approach to analogy.

The precise nature of the subsumption relation between generalization and source or target domain, respectively, is defined by the specific analogy model, possibly ranging from semantic subsumption in a suitable ontology, through taxonomic subsumption based on names and labels, logical subsumption in a model-theoretic sense, to purely syntactic subsumption in a formal language.

One example for a generalization-based computational analogy-model (and the system used in COINVENT) is the already aforementioned HDTP (Schmidt et al. 2014). The framework has been conceived as a mathematically sound theoretical model and implemented engine for computational analogy-making, on a syntax basis computing analogical relations and inferences for domains which are presented in (when allowing for re-representation possibly different) many-sorted first-order logic (FOL) languages. Source and target domains are handed over to the system in terms of finite axiomatizations and HDTP tries to compute a generalization between both domains. This is done by aligning pairs of formulae from the two domains by means of restricted higher-order anti-unification (Schwering et al. 2009): Given two terms, one from each domain, HDTP computes an anti-instance in which distinct subterms have been replaced by variables so that the anti-instance can be seen as a meaningful generalization of the input terms. As already indicated by the name, the class of admissible substitution operations is limited. On each expression, only renamings, fixations, argument insertions, and permutations may be performed. By this process, HDTP tries to find the least general generalization of the input terms, which (due to the higher-order nature of the anti-unification) is not unique. In order to solve this problem, current implementations of HDTP rank possible generalizations according to a complexity measure on the chain of substitutions — the respective values of which are taken as heuristic costs — and returns the least expensive solution as the preferred one.

HDTP extends the notion of generalization from terms to formulae by basically treating formulae in clause form and terms alike. Finally, as analogies rarely rely exclusively on one isolated pair of formulae from source and target domain, but usually encompass sets of formulae (possibly completely covering one or even both input domains), a process iteratively selecting pairs of formulae for generalization has been included. The selection of formulae is again based on a heuristic component. Mappings in which substitutions can be reused get assigned a lower cost than isolated substitu-

tions, leading to a preference for coherent over incoherent mappings.

Due to the use of many-sorted FOL as an expressive representation language, and the purely syntax-based generalization approach underlying HDTP, over the last years the framework has shown remarkable generalizability and generality. Having originally been conceived and applied for modelling the Rutherford analogy and poetic metaphors, as well as for providing an alternate account of (Falkenhainer, Forbus, and Gentner 1989)'s heat-flow analogy in (Schwering et al. 2009), without major changes to the model HDTP has by now been applied to different tasks from different domains, such as modeling a potential inductive analogy-based process for establishing the fundamental concepts of arithmetics (Guhe et al. 2010), or studies applying the framework to modeling analogy use in education and teaching situations (Besold 2014).

## Combining Conceptual Theories Using Amalgams

The notion of amalgams was developed in the context of CBR (Ontanón and Plaza 2010), where new problems are solved based on previously solved problems (or cases, residing on a case base). Solving a new problem often requires more than one case from the case base, so their content has to be combined in some way to solve the new problem. The notion of an amalgam of two cases (two descriptions of problems and their solutions) is a proposal to formalize the ways in which cases can be combined to produce a new, coherent case.

Formally, the notion of amalgams can be defined in any representation language $\mathcal{L}$ for which a subsumption relation $\sqsubseteq$ between the formulae (or descriptions) of $\mathcal{L}$ can be defined. We say that a description $I_1$ subsumes another description $I_2$ ($I_1 \sqsubseteq I_2$) when $I_1$ is more general (or equal) than $I_2$. Additionally, we assume that $\mathcal{L}$ contains the infimum element $\bot$ (or 'any'), and the supremum element $\top$ (or 'none') with respect to the subsumption order.

Next, for any two descriptions $I_1$ and $I_2$ in $\mathcal{L}$ we can define their *unification*, ($I_1 \sqcup I_2$), which is the *most general specialization* of two given descriptions, and their *anti-unification*, ($I_1 \sqcap I_2$), defined as the *least general generalization* of two descriptions, representing the most specific description that subsumes both. 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 saying that $I_1 \sqcup I_2 = \top$ (i.e., they have no common specialization except 'none'). The anti-unification $I_1 \sqcap I_2$ contains all that is common to both $I_1$ and $I_2$; when they have nothing in common, then $I_1 \sqcap I_2 = \bot$. Depending on $\mathcal{L}$ anti-unification and unification might be unique or not.

The notion of an *amalgam* can be conceived of as a generalization of the notion of unification: as 'partial unification' (Ontanón and Plaza 2010). Unification means that what is true for $I_1$ or $I_2$ is also true for $I_1 \sqcup I_2$; e.g., if $I_1$ describes 'a red vehicle' and $I_2$ describes 'a German minivan' then their unification yields a common specialization like 'a red German minivan.' Two descriptions may contain information that produces an inconsistency when unified; for instance

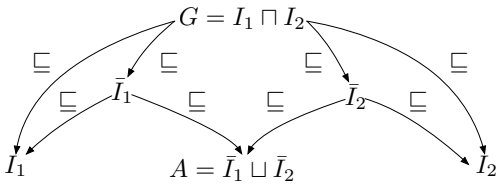



Figure 2: A diagram of an amalgam $A$ from inputs $I_1$ and $I_2$ where $A = \bar{I}_1 \sqcup \bar{I}_2$.

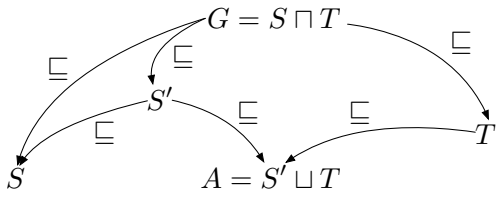



Figure 3: A diagram that transfers content from source $S$ to a target $T$ via an asymmetric amalgam $A$.

'a red French sedan' and 'a blue German minivan' have no common specialization except $\top$. An *amalgam* 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 can define an amalgam of two input descriptions as follows:

**Definition 1 (Amalgam)** *A description $A \in \mathcal{L}$ is an amalgam of two inputs $I_1$ and $I_2$ (with anti-unification $G = I_1 \sqcap I_2$) if there exist two generalizations $\bar{I}_1$ and $\bar{I}_2$ such that (1) $G \sqsubseteq \bar{I}_1 \sqsubseteq I_1$, (2) $G \sqsubseteq \bar{I}_2 \sqsubseteq I_2$, and (3) $A = \bar{I}_1 \sqcup \bar{I}_2$*

When $\bar{I}_1$ and $\bar{I}_2$ have no common specialization then trivially $A = \top$, since their only unifier is "none". For our purpose we will be only interested in non-trivial amalgams.

This definition is illustrated in Fig. 2, where the anti-unification of the inputs is indicated as $G$, and the amalgam $A$ is the unification of two concrete generalizations $\bar{I}_1$ and $\bar{I}_2$ of the inputs. Equality here should be understood as $\sqsubseteq$-equivalence: $X \equiv Y$ iff $X \sqsubseteq Y$ and $Y \sqsubseteq X$. Conventionally, we call the *space of amalgams* of $I_1$ and $I_2$ the set of all amalgams $A$ that satisfy Definition 1.

Usually we are interested only in maximal amalgams of two input descriptions, i.e., those amalgams that contain maximal parts of their inputs that can be unified into a new coherent description. Formally, an amalgam $A$ of inputs $I_1$ and $I_2$ is maximal if there is no other non-trivial amalgam $A'$ of inputs $I_1$ and $I_2$ such that $A \sqsubset A'$. The reason why we are interested in maximal amalgams is very simple: a non-maximal amalgam $\bar{A} \sqsubset A$ preserves less compatible information from the inputs than the maximal amalgam $A$. Conversely, any non-maximal amalgam $\bar{A}$ can be obtained by generalizing a maximal amalgam $A$, since $\bar{A} \sqsubset A$.

There is a special case of particular interest that is called

an *asymmetric amalgam*, in which the two inputs play different roles. The inputs are called source and target, and while the source is allowed to be generalized, the target is not.

**Definition 2 (Asymmetric Amalgam)** *An asymmetric amalgam $A \in \mathcal{L}$ of two inputs $S$ (source) and $T$ (target) satisfies that $A = S' \sqcup T$ for some generalization of the source $S' \sqsubseteq S$.*

As shown in Fig. 3, the content of target $T$ is transferred completely into the asymmetric amalgam, while the source $S$ is generalized. The result is a form of partial unification that preserves all information in $T$ while relaxing $S$ by generalization and then unifying one of those generalizations $S'$ with $T$ itself. As before, we will usually be interested in maximal amalgams: in this case, a maximal amalgam corresponds to transferring maximal content from $S$ to $T$ while keeping the resulting amalgam $A$ consistent. For these reasons asymmetric amalgams can be seen as models of a form of analogical inference, transferring information from the source to the target by creating a new amalgam that enriches the latter with the content of $S'$ (Ontanón and Plaza 2012).

## Analogy-Based Concept Blending in COINVENT

Taking the concept of generalization-based analogies (and HDTP as suitable framework for the computation of the latter) together with the notion of asymmetric amalgams, we now can introduce the core idea(s) behind concept blending as performed in COINVENT in the next subsection, subsequently also showing the feasibility of the approach in two examples. The general suitability of the approach is demonstrated revisiting the "sign forest" metaphor from (Kutz et al. 2012), an implementation using HDTP is exemplified (re-)constructing the concept of a foldable toothbrush.

### The Core Model: An Analogy-Inspired View

One of the early formal accounts on concept blending, which is especially influential to the approach applied in COINVENT, is the classical work by Goguen using notions from algebraic specification and category theory (Goguen 2006). This version of concept blending can be described by the diagram in Fig. 4, where each node stands for a representation an agent has of some concept or conceptual domain. We will call these representations "conceptual spaces" and in some cases abuse terminology by using the word "concept" to really refer to its representation by the agent. The arrows stand for morphisms, that is, functions that preserve at least part of the internal structure of the related conceptual spaces. The idea is that, given two conceptual spaces $I_1$ and $I_2$ as input, we look for a generalization $G$ and then construct a blend space $B$ in such a way as to preserve as many as possible structural alignments between $I_1$ and $I_2$ established by the generalization. This may involve taking the functions to $B$ to be *partial*, in that not all the structure from $I_1$ and $I_2$ might be mapped to $B$. In any case, as the blend respects (to the largest possible extent) the relationship between $I_1$ and $I_2$, the diagram will commute.

Concept invention by concept blending can then be phrased as the following task: given two representations of

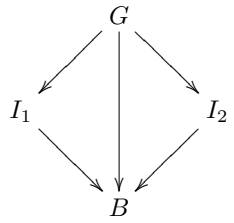



Figure 4: A conceptual overview of (Goguen 2006)'s account of conceptual blending.

two domain theories $I_1$ and $I_2$, we need first, to compute a generalized theory $G$ of $I_1$ and $I_2$ (which codes the commonalities between $I_1$ and $I_2$) and second, to compute the blend theory $B$ in a structure preserving way such that new properties hold in $B$. Ideally, these new properties in $B$ are considered to be (moderately) interesting properties. In what follows, for reasons of simplicity and without loss of generality we assume that the additional properties are just provided by one of the two domains, i.e., we align the situation with a standard setting in computational analogy-making by renaming $I_1$ and $I_2$. The domain providing the additional properties for the concept blend will be called source $S$, the domain providing the conceptual basis and receiving the additional features will be called target $T$.

In COINVENT's account, the reasoning process is then triggered by the computation of the generalization $G$ (generic space), where for concept invention we will only need the mapping mechanism and replace the transfer phase by a new blending algorithm. The mapping is achieved via the usual generalization process between $S$ and $T$, in which a generalized theory is created that reflects common aspects of both spaces. The generalized theory can be projected back into the original spaces by specializations $\phi_S$ and $\phi_T$, respectively. As $S$ and $T$ might contain elements which are not reflected in the shared generalization, it holds that $\phi_S(G) \subseteq S$ and $\phi_T(G) \subseteq T$. While in analogy making the analogical relations are used in the transfer phase to translate additional uncovered knowledge from the source to the target space, blending combines additional facts (i.e., elements from $S \setminus S_C$ or $T \setminus T_C$) from one or both spaces. Therefore the process of blending can build on the generalization and specializations provided by the analogy engine, but has to include a new mechanism for transfer and concept combination. Here, amalgams naturally come into play: The set of specializations can be inverted and applied to generalize the original source theory $S$ into a more general version $S'$ (forming a superset of the shared generalization $G$, also including previously uncovered knowledge from the source) which then can be combined into an asymmetric amalgam with the target theory $T$, forming the (possibly underspecified) proto-blend $T'$ of both. In a final step, $T'$ is then completed into the blended theory and output of the process $T_B$ by applying corresponding specialization steps stored from the generalization process between $S$ and $T$ (see also Fig. 5).

If we now take the domains to be represented in the form of finite axiomatizations as processed by HDTP, in an im-

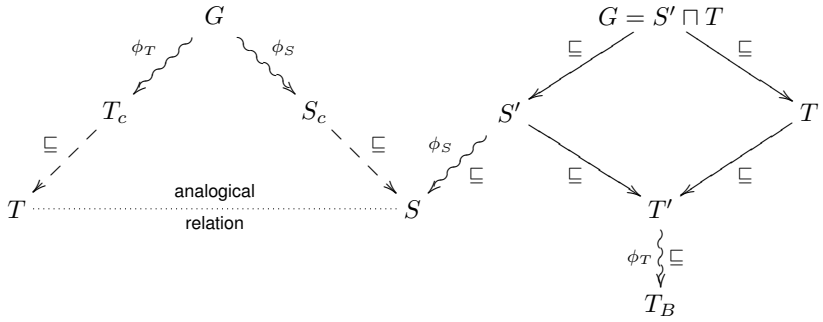



Figure 5: A general overview of COINVENT's account of concept blending using generalization-based analogy and asymmetric amalgams: The shared generalization $G$ from $S$ and $T$ is computed with $\phi_S(G) = S_c$. The relation $\phi_S$ is subsequently re-used in the generalization of $S$ into $S'$, which is then combined in an asymmetric amalgam with $T$ into the proto-blend $T' = S' \sqcup T$ and finally, by application of $\phi_T$, completed into the blended output theory $T_B$. (Here $\sqsubseteq$ indicates subsumption between theories in the direction of the respective arrows.)

plementation of the general model we can use the analogy-engine for computing the generalizations and deriving the corresponding substitutions. In the generalization step between $S$ and $T$, as usual pairs of formulas from the source and target spaces are anti-unified for deriving the generalized theory $G$, and the specializations $\phi_S$ and $\phi_T$ become substitutions which are computed during anti-unification.

## Example 1: The Sign Forest

We now want to revisit the example of the blend *sign forest* discussed in (Kutz et al. 2012), providing an interpretation of the concept from a metaphor-centered perspective and showing how the general COINVENT model can serve for reconstructing the blending process. In what follows we consider *sign forest* equivalent to the expression "a forest of signs", that shows more clearly its metaphorical nature.

The original *sign forest* blend was defined in the context of blending ontologies, which means that the involved inputs for blending were ontological descriptions of trees, forests, and (traffic) signs. This approach views a concept such as *tree* defined as an ontological specification of the concept of tree: a specification that is ideally so general as to cover *all kinds of trees*; the same can be said about forest, and (traffic) signs. As such, certain properties and relations are selected to form these specifications that are useful for an ontology framework. However, our approach follows the notion that concepts in human cognition can often be viewed, in cognitive science, as bundles of their most typical properties (albeit typicality may certainly be context-dependent). This view is also taken in examples by (Fauconnier and Turner 1998) that are used to show how conceptual blending works: a boathouse has typical properties of boat and house —but not other properties that may appear in an ontological specification of boat and house.

Thus, in this approach, the concept of tree is typically formed by a plant having roots, a trunk and a crown (even if there may be plants categorized as trees that do not have a trunk, this is ignored as it does not belong to the bundle of properties that are typical); this view is depicted as $I_2$ in the bottom right of Fig. 6, where other properties are included, like plants being not mobile and the roots fixing the (typical) tree to the ground. Finally, a forest is commonsensically defined as a group of trees. The second concept, (traffic) sign, may come in many forms (as we know from own experience), but the first that comes to mind is the most typical one: the signpost. The signpost is typically fixed on the ground near a road, and has a post supporting a surface panel depicting some traffic related information (labeled $I_1$ in the lower left corner of Fig. 6). The cognitive advantage of a signpost is that it has a recognizable physical structure, while "traffic sign" is so generic as to be a merely functional-based concept: any kind of surface panel depicting some traffic-related information is a traffic sign.

The generic space $G$ of conceptual blending corresponds to the anti-unification shown as $G = I_1 \sqcap I_2$ in Fig. 6; $G$ depicts common structure between a signpost and a tree: a stem-like object, fixed to the ground, and supporting another object on top. As discussed later, this common structure is the basis for a metaphor like "a forest of signs" to *make sense* — in contradistinction to a metaphor that does not *make sense* such as "a forest of chairs", even when a typical chair is made of wood.

Now, the construction of the blended metaphor for *sign forest* can be interpreted easily in the combined generalization-based analogy and amalgam framework: the input spaces can be generalized in different ways (although always satisfying what they already have in common, namely $G$). Different generalizations would yield different amalgams, but the one we are considering here can be seen as generalizing $I_2$ into $\bar{I}_2$, as shown in Fig. 6. Now this generalization $\bar{I}_2$ can directly be unified with $I_1$, since $\bar{I}_1$ is identical to $I_1$; this unification yields the amalgam $A = \bar{I}_1 \sqcup \bar{I}_2$ that, as shown in Fig. 6, represents a "forest of signposts". Moreover, since $I_1 \equiv \bar{I}_1$, this model is an asymmetric amalgam, as evidenced by the fact we generalize the source (Forest) until it unifies with the target (Signpost), while the latter remains fixed (i.e., is not generalized).

In order to support our perspective that a metaphor (viewed as an analogy and amalgam combination in natural language) is based on some (strong enough) common

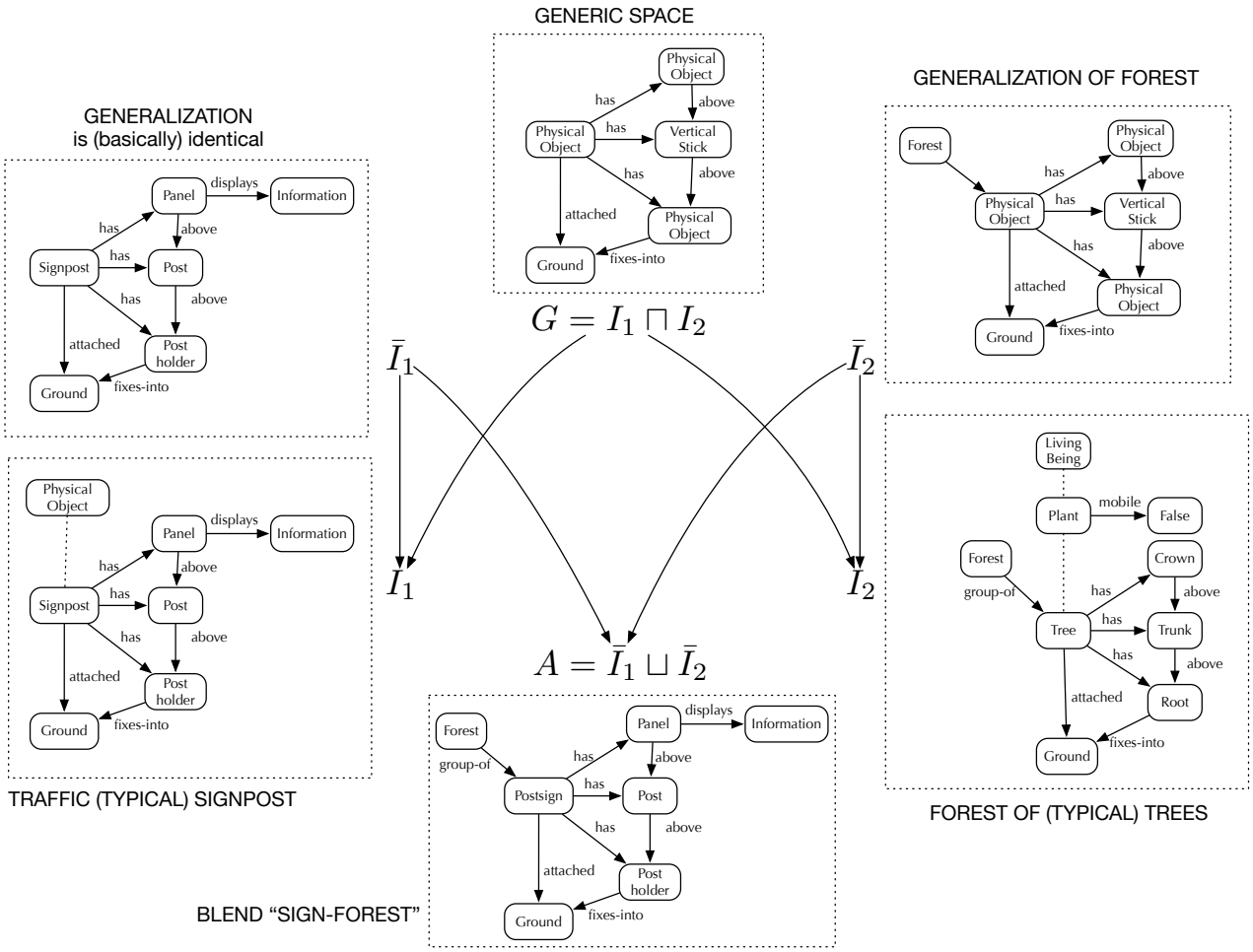



Figure 6: Blending schema for "Sign Forest" when inputs are typical concepts for "Sign" (traffic signpost) and "Forest" (forest of typical trees); the arrows indicate subsumption ($\sqsubseteq$) as in Figure 2.

structure of the typical concepts participating in the blending process, we checked if other metaphors can be constructed, or better yet, have already been constructed, that are based on the same kind of generic space $G$. We used Google's n-grams database to search for existing phrases in which "forest of X" is used metaphorically[2]. Most n-grams starting with "forest of" were about places or kinds of trees, as is to be expected; still, we found the following metaphors used on the web: (1) forest of spears, (2) forest of masts, and (3) forest of marble columns. These three cases have a generic space that is very similar to $G$: they all represent a multitude of vertical stem-like objects. Some differences are: while masts and columns are fixed, spears are not fixed to the ground, but may be used in a context where they are vertical and immobile stems, supporting a pointed tip; masts and columns support different kinds of objects, but all three examples have generic spaces resembling $G$ in Fig. 6.

What about counterexamples? We did not find "forest of chairs" of course, and there were other metaphors on forest, but they were based on different generic spaces and different input spaces; we found these metaphors: "forest of X", where X could be *opinions, possibilities, desires, words, human experience*. Clearly, these metaphors were not based on the trees being elements of a "forest", but on the human experience of (walking in) the forest as a place of multiplicity of paths, options, destinations. We think they are not counterexamples, but rather examples of blends from different input spaces.

## Example 2: The Folding Toothbrush

Having given an example for the general model in the previous subsection, we now want to also exemplify a concrete implementation of the approach using HDTP as anal-

[2]Google's 3-grams starting with "fo" are available at: `http://storage.googleapis.com/books/ngrams/books/googlebooks-eng-all-3gram-20120701-fo.gz`

Figure 7: Brillo, an example of a foldable toothbrush as produced by Metaphys.

ogy framework. As application example, we will use the blending-driven (re-)invention of foldable toothbrushes as, for instance, the one depicted in Fig. 7.

Currently, when using HDTP, the required subsumption relation between theories is given by logical semantic consequence $\models$, i.e., $A \sqsubseteq A'$ if $A' \models A$ for any two theories $A$ and $A'$. In order to make sure that this relationship is preserved by HDTP's syntax-based operations, the range of admissible substitutions for restricted higher-order anti-unifications has to be further constrained to only allow for fixations and renamings.

Foldable toothbrushes are a conceptual combination between a typical toothbrush and a folding mechanism like that of pocketknives. In order to reconstruct the underlying blending process, we start with the stereotypical characterizations of a toothbrush and a pocketknife in a many-sorted first-order logic representation from Table 1.

| |
|---|
| **Sorts:** |
| *entity*, *part*, *functionality* |
| **Entities:** |
| *toothbrush*, *pocketknife* : *entity* *handle*, *brush_head*, *blade*, *hinge* : *part* |
| *brush*, *cut*, *fold* : *functionality* |
| **Predicates:** |
| *has_part* : *entity* × *part*, *has_functionality* : *entity* × *functionality* |
| **Laws of the pocketknife characterization:** |
| ($\alpha_1$) *has_part*(*pocketknife*, *handle*) ($\alpha_2$) *has_part*(*pocketknife*, *blade*) |
| ($\alpha_3$) *has_functionality*(*pocketknife*, *cut*) ($\alpha_4$) *has_part*(*pocketknife*, *hinge*) |
| ($\alpha_5$) *has_functionality*(*pocketknife*, *fold*) |
| **Laws of the horse characterization:** |
| ($\beta_1$) *has_part*(*toothbrush*, *handle*) ($\beta_2$) *has_part*(*toothbrush*, *brush_head*) |
| ($\beta_3$) *has_functionality*(*toothbrush*, *brush*) |

Table 1: Example formalizations of stereotypical characterizations for a pocketknife $S$ and a toothbrush $T$.

Given these characterizations, HDTP can be used for finding a common generalization of both, for instance (due to the syntactic similarities and the system's heuristics) aligning and generalizing $\alpha_1$ with $\beta_1$, $\alpha_2$ with $\beta_2$, and $\alpha_3$ with $\beta_3$. Subsequently, reusing the same anti-unifications (corresponding to $\phi_S$), the source theory $S$ is generalized into $S'$ as given in Table 2: $\gamma_1$ corresponds to $\alpha_1/\beta_1$, $\gamma_2$ to $\alpha_2/\beta_2$, $\gamma_3$ to $\alpha_3/\beta_3$, and $\gamma_4$ and $\gamma_5$ are obtained by generalizing $\alpha_4$ and $\alpha_5$, respectively.

| |
|---|
| **Entities:** |
| $E$ : *entity*, $P$ : *part*, $F$ : *functionality* |
| **Laws:** |
| ($\gamma_1$) *has_part*($E$, *handle*) ($\gamma_2$) *has_part*($E$, $P$) |
| ($\gamma_3$) *has_functionality*($E$, $F$) |
| ($\gamma_4$*) *has_part*($E$, *hinge*) ($\gamma_5$*) *has_functionality*($E$, *fold*) |

Table 2: Abbreviated representation of the generalized source theory $S'$ based on the stereotypical characterizations for a toothbrush and a pocketknife (axioms not obtained from the covered subset $S_c$ are highlighted by *).

Computing the asymmetric amalgam of $S'$ with the (fixed) target theory $T$, we obtain the proto-blend $T'$ from Table 3. As $T'$ still features axioms containing non-instantiated variables, $\phi_T$ is applied to the theory resulting in the (with respect to $\phi_T$) fully instantiated blend theory $T_B$ from Table 4, describing the concept of a hinge-equipped toothbrush that can be folded.

| |
|---|
| **Entities:** |
| $E$ : *entity* |
| **Laws:** |
| ($\delta_1$) *has_part*(*toothbrush*, *handle*) ($\delta_2$) *has_part*(*toothbrush*, *brush_head*) |
| ($\delta_3$) *has_functionality*(*toothbrush*, *brush*) |
| ($\delta_4$) *has_part*($E$, *hinge*) ($\delta_5$) *has_functionality*($E$, *fold*) |

Table 3: Abbreviated representation of the proto-blend $T'$ obtained from computing the asymmetric amalgam between $S'$ and $T$.

| |
|---|
| **Laws:** |
| ($\delta_1$) *has_part*(*toothbrush*, *handle*) ($\delta_2$) *has_part*(*toothbrush*, *brush_head*) |
| ($\delta_3$) *has_functionality*(*toothbrush*, *brush*) |
| ($\delta_4$) *has_part*(*toothbrush*, *hinge*) ($\delta_5$) *has_functionality*(*toothbrush*, *fold*) |

Table 4: Abbreviated representation of $T_B = \phi_T(T')$.

## Conclusions

We presented a perspective on the blending of concept theories building on generalization-based analogy and the amalgam framework: Building upon analogy models of generalization and domain matching, asymmetric amalgams allow to provide a sound model for the controlled computation of the concept blend(s) of two input theories.

Clearly, this is not the only attempt at developing a computational model of (some facet of) concept blending: (Martinez et al. 2014) present an algorithmic approach for blending mathematical theories, (Kutz et al. 2015) give an account

of ontological blending, (Li et al. 2012) describe the goal- and context-sensitive blending-based production of creative artifacts, and (Martinez et al. 2012) consider concept blending in a human-level AI context. Still, in combining the generality of generalization-based analogies and the amalgam framework, COINVENT's approach stands out as high-level, cognitively-inspired perspective on concept blending.

## Acknowledgements

The authors acknowledge the financial support of the Future and Emerging Technologies within the 7th Framework Programme for Research of the European Commission, under FET-Open grant 611553 (COINVENT).

## References

Aamodt, A., and Plaza, E. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1):39–59.

Besold, T. R. 2014. Sensorimotor Analogies in Learning Abstract Skills and Knowledge: Modeling Analogy-Supported Education in Mathematics and Physics. In *Proc. of the AAAI Fall 2014 Symposium on Modeling Changing Perspectives: Reconceptualizing Sensorimotor Experiences*, volume FS-14-05 of *AAAI Press Technical Reports*.

Boden, M. A. 2003. *The Creative Mind: Myths and Mechanisms*. Routledge.

Falkenhainer, B.; Forbus, K.; 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.

Gentner, D., and Smith, L. A. 2013. Analogical learning and reasoning. In Reisberg, D., ed., *The Oxford Handbook of Cognitive Psychology*. Oxford University Press. 668–681.

Goguen, J. A., and Harrell, D. F. 2010. Style: A Computational and Conceptual Blending-Based Approach. In Argamon, S.; Burns, K.; and Dubnov, S., eds., *The Structure of Style*. Springer. 291–316.

Goguen, J. 2006. Mathematical models of cognitive space and time. In Andler, D.; Ogawa, Y.; Okada, M.; and Watanabe, S., eds., *Reasoning and Cognition; Proc. of the Interdisciplinary Conference Series on Reasoning Studies*, 125–128.

Guhe, M.; Pease, A.; Smaill, A.; Schmidt, M.; Gust, H.; Kühnberger, K.-U.; and Krumnack, U. 2010. Mathematical reasoning with higher-order anti-unification. In *Proc. of the 32nd Annual Conference of the Cognitive Science Society*, 1992–1997. Cognitive Science Society.

Hofstadter, D., and Mitchell, M. 1994. The Copycat project: a model of mental fluidity and analogy-making. In *Advances in Connectionist and Neural Computation Theory*, volume 2: Analogical Connections, 31–112. Ablex.

Kutz, O.; Mossakowski, T.; Hois, J.; Bhatt, M.; and Bateman, J. 2012. Ontological Blending in DOL. In *Proc. of the 1st International Workshop on "Computational Creativity, Concept Invention, and General Intelligence"*, Publications of the Institute of Cognitive Science, Univ. of Osnabrück.

Kutz, O.; Bateman, J.; Neuhaus, F.; Mossakowski, T.; and Bhatt, M. 2015. E Pluribus Unum. In Besold, T. R.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity Research: Towards Creative Machines*, volume 7 of *Atlantis Thinking Machines*. Atlantis Press. 167–196.

Li, B.; Zook, A.; Davis, N.; and Riedl, M. 2012. Goal-Driven Conceptual Blending: A Computational Approach for Creativity. In *Proc. of the Third International Conference on Computational Creativity*, 9–16.

Martinez, M.; Besold, T. R.; Abdel-Fattah, A.; Gust, H.; Schmidt, M.; Krumnack, U.; and Kühnberger, K.-U. 2012. Theory Blending as a Framework for Creativity in Systems for General Intelligence. In Wang, P., and Goertzel, B., eds., *Theoretical Foundations of Artificial General Intelligence*. Atlantis Press. 219–239.

Martinez, M.; Krumnack, U.; Smaill, A.; Besold, T. R.; Abdel-Fattah, A. M.; Schmidt, M.; Gust, H.; Kühnberger, K.-U.; Guhe, M.; and Pease, A. 2014. Algorithmic Aspects of Theory Blending. In Aranda-Corral, G.; Calmet, J.; and Martín-Mateos, F., eds., *Artificial Intelligence and Symbolic Computation*, volume 8884 of *LNCS*. Springer. 180–192.

Ontanón, S., and Plaza, E. 2010. Amalgams: A Formal Approach for Combining Multiple Case Solutions. In Bichindaritz, I., and Montani, S., eds., *Case-Based Reasoning. Research and Development*, volume 6176 of *LNCS*. Springer. 257–271.

Ontanón, S., and Plaza, E. 2012. On Knowledge Transfer in Case-Based Inference. In Agudo, B. D., and Watson, I., eds., *Case-Based Reasoning Research and Development*, volume 7466 of *LNCS*. Springer. 312–326.

Pereira, F. C. 2007. *Creativity and AI: A Conceptual Blending Approach*. Mouton de Gruyter.

Schmidt, M.; Krumnack, U.; Gust, H.; and Kühnberger, K.-U. 2014. Heuristic-Driven Theory Projection: An Overview. In Prade, H., and Richard, G., eds., *Computational Approaches to Analogical Reasoning: Current Trends*. Springer. 163–194.

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 *Proc. of the 5th International Conference on Computational Creativity, Ljubljana, Slovenia*.

Schwering, A.; Krumnack, U.; Kühnberger, K.-U.; and Gust, H. 2009. Syntactic Principles of Heuristic-Driven Theory Projection. *Journal of Cognitive Systems Research* 10(3):251–269.

Smaling, A. 2003. Inductive, analogical, and communicative generalization. *International Journal of Qualitative Methods* 2(1).

Thagard, P., and Stewart, T. C. 2011. 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* 11(3/4):253–281.

Winston, P. H. 1980. Learning and Reasoning by Analogy. *Commun. ACM* 23(12):689–703.

# *Vismantic*: Meaning-making with Images

**Ping Xiao, Simo Linkola**
Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
{ping.xiao, slinkola}@cs.helsinki.fi

### Abstract

This paper presents *Vismantic*, a semi-automatic system generating proposals of visual composition (*visual ideas*) in order to express specific meanings. It implements a process of developing visual solutions from 'what to say' to 'how to say', which requires both conceptual and visual creativity. In particular, Vismantic extends our previous work on using conceptual knowledge to find diverse visual representations of abstract concepts, with the capacity of combining two images in three ways, including *juxtaposition*, *replacement* and *fusion*. In an informal evaluation consisting of five communication tasks, Vismantic demonstrated the potential of producing a number of expressive and diverse ideas, among which many are surprising. Our analysis of the generated images confirms that visual meaning-making is a subtle interaction between all elements in a picture, for which Vismantic demands more visual semantic knowledge, higher image analysis and synthesis skills, and the ability of interpreting composed images, in order to deliver more ideas that make sense.

## Introduction

Aesthetics and meaning are two main concerns of art. The work presented in this paper focuses on meaning-making in image generation. Particularly, we are interested in conveying specific meanings, in contrast to vague or divergent interpretations. A common way of constructing meanings in images is combining objects, where meanings arise from the objects (denotation and connotation) and the relations between them. Such combination involves two main decisions: *which objects to combine* and *how to combine them*.

Contemporary print advertisements offer abundant examples of combining objects to express specific meanings. In general, an ad tells about a desirable attribute of a product. Hence, usually two objects are combined, the product (or something closely related) and another thing that embodies the attribute. For example, an ad for promoting dairy products shows a bone made of milk. Regarding how to visually combine two objects, Phillips and McQuarrie (2004) identified three ways (*visual operations*): *juxtaposition* (two objects side by side), *fusion* (two objects merged together), and *replacement* (only one object is present, which occupies the usual place of the other object).

Obviously, the above visual operations do not appear only in ads, and the relations between objects are not limited to attribute. The news collage in (Krzeczkowska et al. 2010) (see Related Work) is an example of juxtaposing more than two objects. Dalí's liquid clock[1] is an example of fusion, and Duchamp's urinal[2] surrounded by artworks in an exhibition can be seen as an example of replacement.

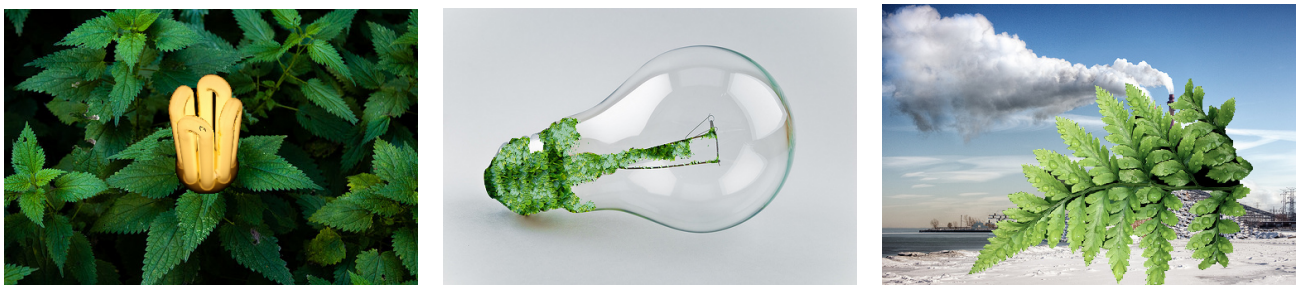
In this paper, we present *Vismantic*, a semi-automatic system combining pictures of objects to express simple meanings described by pairs of a subject word and a message word. A message may be an attribute of the subject, or have a causal or an opposite relation to the subject. Vismantic first searches for photos that represent the subject and the message respectively and are as diverse as possible. It then applies juxtaposition, fusion and replacement to the photos found. We provide the formalization and computational implementation of the three visual operations. Nevertheless, Vismantic is not yet fully automatic; it needs user filtering at intermediate stages.

Vismantic is a workflow of integrating conceptual and visual creativity in making images. Such integration is necessary, since both kinds of creativity are required in common visual communication tasks and there do not exist many such systems. We present here the first version of Vismantic and the results of an informal evaluation, which functions as identifying the problems in the field. Another important objective of the present work is using computational modeling for studying *visual compositional semantics*. The semantics of an image is a synergy of every element in it, including the subtle details. But, there has not been much formal study on it. Formalization and computational implementation are great tools for testing rules and hypotheses. Our newly gained insights are presented in the Evaluation and Analysis section.

Vismantic focuses on the *variety* and *novelty* of compositions (*visual ideas*), rather than generating perfect images. As an example, Fig. 1 shows some of the ideas generated by Vismantic in order to say "electricity is green (sustainable)".

In the remainder of this paper, related work is introduced first, followed by the details of how Vismantic works. We then present the experiment we conducted to evaluate its

[1] http://en.wikipedia.org/wiki/The_Persistence_of_Memory
[2] http://en.wikipedia.org/wiki/Fountain_%28Duchamp%29



(a) (b) (c)

Figure 1: Example visual ideas generated for Task 1 “electricity is green (sustainable)”. 1a: a light bulb replaces a tuft of green leaves; 1b: green leaves are fused with the screw base and wire filament of a light bulb; 1c: a branch of leaves replaces a power station.

ideation capacity, as well as our analysis of the test results. Finally, we give conclusions and propose future work.

## Related Work

Within the Computational Creativity community, the bulk of work on visual creativity has concentrated on aesthetics, while meaning creation has only come into focus lately.

Krzeczkowska et al. (2010) created a computer visual artist, which has a basic level of intention and expresses it with collages. At regular intervals it accesses news articles from a few internet sources, and takes the viewpoints of the authors by extracting most-content-indicative keywords (only nouns) from the articles. The keywords are used to retrieve digital images from a few online and local sources, including Corel, Flickr and Google Images. The retrieved images, in their whole or segments, are assembled according to one of the grid-based templates, which is then rendered with pencils, pastels or paints. In the example presented in the paper, ten nouns are extracted in order to cover all the central subjects of an article. The use of collage makes it easy to present the multiple facets of an event. In contrast, Vismantic relates only two objects and intends more specific messages. Moreover, it combines images in two additional ways, i.e. replacement and fusion.

Another computer visual artist, DARCI (Digital Artist Communicating Intent) (Norton, Heath, and Ventura 2010; 2011), renders a given image in order to represent a list of adjectives. It learns, from human-annotated images, the mappings between adjective synsets and low-level image features, including color, light, texture and shape. The mapping for each synset is encoded in a series of artificial neural networks (ANNs). To render an image, DARCI selects a set of image filters through an evolutionary mechanism, where the ANNs are used in each generation to assess how well a rendering reflects the specified adjectives. Unlike DARCI, which focuses on the overall impression of images and the meaning-carrying capacity of low-level image features, Vismantic primarily uses objects and their relations to convey meanings.

In addition, there is work on suggesting objects (in the form of concepts) for images to be generated. Xiao and Blat (2013) were interested in the use of pictorial metaphors in advertisements and created a program proposing metaphor vehicles, to which a product (metaphor tenor) and a few attributes with different levels of prominence are given as an input. The program first searches in several commonsense knowledge bases for concepts that have the main attribute as one of their stereotypical properties. Then, it evaluates the aptness of the concepts found as metaphor vehicles, in regard to imageability, affect polarity, attribute salience, secondary attributes and similarity with tenor. Another work is a software called Perception (De Smedt et al. 2013), which assists the brainstorming of artists in general. It is backed by a semantic network of concepts and their adjective properties. By concept clustering and graph path finding, Perception is able to find instances of novel concepts such as ‘creepy animals’, and make analogies, e.g., proposing a toad as a symbol of Brussels. Both works are made for creative visual tasks, and both touch only the conceptual aspect. They are relevant in augmenting the conceptual creativity of Vismantic.

Outside the Computational Creativity community, a relevant field is Content-aware Image Synthesis (Diakopoulos, Essa, and Jain 2004; Lalonde et al. 2007; Chen et al. 2009), which deals with composing scenes (images) using pictorial elements taken from photos. Its center of investigation is how to make a composition look as realistic as possible, considering that photos normally vary in camera pose, lighting, scale, resolution, etc. There is overlap in the image processing techniques used in this field and by Vismantic. The difference is that, in Content-aware Image Synthesis, it is the user who dictates the composite objects of an image, not the computer.

## Vismantic Workflow

Vismantic takes as an input a subject word and a message word. To generate visual ideas, it follows three major steps:

I Find representative photos of the subject and message, respectively;

II Preprocess photos found;

III Apply visual operations (juxtaposition, replacement and fusion).

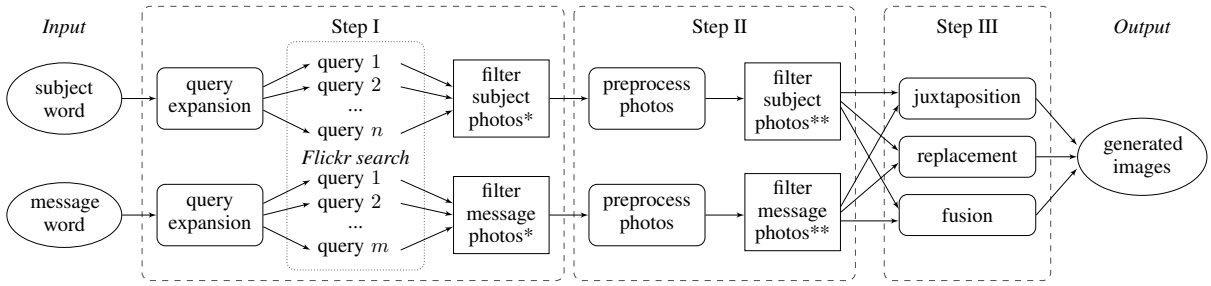



Figure 2: Vismantic workflow. * and **: two filterings have different content (see the text for details).

Step I and II both involve user filtering. The above workflow is also illustrated in Fig. 2 for clarity. The details of the three steps are presented below. Because the user filterings in Step I and II are influenced by how the visual operations are implemented, Step III is introduced first, followed by Step I and II.

## Implementation of Three Visual Operations

In this subsection, we introduce first the specifications we give to the visual operations and then how they are realized with several image processing techniques.

*Juxtaposition* means that two objects are shown side by side in an image. There is no restriction on whether the image of the subject or message should be on the left or right. Also, it does not rely on the context in the generated image to assist understanding.

*Replacement* means that an object takes the place of another object. The context of the replaced object has to be able to hint about it. Again, it is arbitrary whether the subject or the message object should be replaced.

*Fusion* means that an attribute of an object is fused with an attribute of another object, which creates a new object with mixed traits. The new object has to remind viewers of the original objects, which normally depends on the distinctiveness of the attributes.

The above visual operations suggest using pictures of objects in their natural surroundings. We chose Flickr[3] as image source, attempting to capitalize on its diverse content.

In order to implement juxtaposition, replacement and fusion, we have identified three image processing challenges. The first is discovering the most prominent object in an image. The second is removing an object from an image and filling the empty space left in order to make it a natural part of the background. For fusion, we currently use the texture of an object to blend with the object region (texture) in another image. This particular implementation does not require that the object region has a distinctive texture, except a good object extraction. Again, either the image of the subject or message can provide texture or object region. Hence, the third challenge is blending the texture of an object with another object so that traits of both objects are still recognizable.

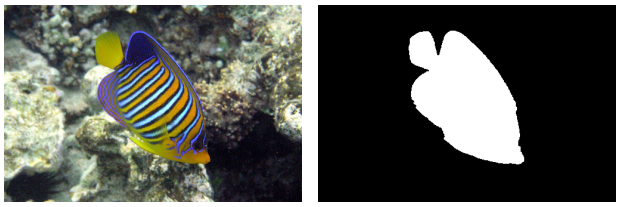
[3]www.flickr.com

The families of image processing techniques we have chosen for solving the above challenges are *saliency-based object extraction*, *inpainting* and *texture transfer*, correspondingly.

*Object Extraction* refers to finding the most prominent (salient) region in an image. The available algorithms usually provide floating point estimation of saliency for each pixel/segment, which is then binarized to obtain a mask of the most salient region (see Fig. 3b). This mask can then be used to extract the most prominent object (Fig. 3c). We use an algorithm created by Cheng et al. (2011,2015), which was concluded to be one of the better performing algorithms in a recent benchmark survey (Borji, Sihite, and Itti 2012). However, the robustness of object extraction algorithms is still far from perfection; the deficiencies include, e.g., partial object extraction and the object humans infer as the most prominent is not extracted. Furthermore, when there is no objectness estimation for the extraction results, regions in images with no clear separation of fore- and backgrounds, e.g., patterns, can be treated as objects.

*Inpainting* techniques were originally created for restoring damaged images or concealing unwanted objects from images. Our intention is to remove objects from images by filling the saliency masks generated by the object extraction algorithm (see Fig. 3d where the object in Fig. 3c is removed using the saliency mask in Fig. 3b). Inpainting algorithms have to deal with textural and structural soundness; textural soundness means preserving the observed textures around the mask, and structural soundness means merging the continuing isophotes (contours of equal luminance) around the mask. As in object extraction, no existing inpainting algorithm gives decent results across the board. Especially, when the removed object is big and/or its surrounding area is diverse, it is difficult to make the inpainted region a natural part of the original image without manual processing. Typical defects are clear patch borders and blurred images. Moreover, in Vismantic, the defects in object extraction may propagate to inpainting.

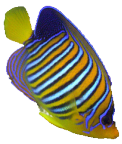
We use fast spatial patch blending (Daisy, Tschumperlé, and Lézoray 2013) as the inpainting algorithm. It iteratively fits small areas (patches) surrounding the saliency mask into the masked area. Patch-based inpainting algorithms are a reasonably fast and convenient way of taking both of the textural and structural soundness into account. The characteristic of spatial patch blending is that it blends overlap-



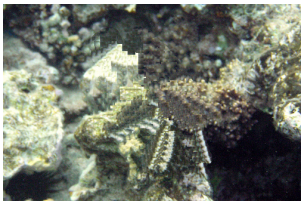
(a) Original



(b) Saliency mask



(c) Extracted object



(d) Inpainted

(e) Input texture

(f) Result of texture transfer

Figure 3: Results of image processing algorithms.

ping regions of adjacent patches making their seams less prominent. However, it has several parameters that should be tuned on an image-to-image basis in order to achieve satisfactory results.

*Texture Transfer* techniques take the texture of an image and apply it to another image so that the other image's characteristics are still recognizable. Comparing to more common texture synthesis methods, which only try to produce larger continuous texture based on a small sample image, texture transfer methods also take a map (usually a gray scale version of the other image or its segment) as an input, and generate texture to match the map's shape while trying to preserve the map's features. See Fig. 3f where the texture in Fig 3e has been transferred to the extracted object in Fig. 3c.

We use the texture transfer method by Harrison (2005), because we perceived it as more robust than other readily available methods in an informal evaluation. Unfortunately, it has the same shortcoming as the fast spatial patch blending – multiple input parameters need to be adjusted for each image in order to get the best results. Harrison's texture transfer method may produce inferior results on many occasions even with near optimal parameter settings. We noticed that, for our purposes, the best quality is obtained when the input texture and map exhibit similar features, but are still relatively different, e.g., the spatial variability of the texture and the map should be in the same order of magnitude.

Combining the three algorithms above, we can achieve our first implementation of the visual operations. Let $I_S$ and $I_M$ be the subject and message images, respectively; let $I_i^s$ be the saliency mask obtained from the image $I_i$; let $I_i^o$ be the object extracted from image $I_i$, given the saliency mask $I_i^s$; and, let $I_i^p$ be the image, where the area of saliency mask $I_i^s$ has been inpainted. With these notations, we can realize the visual operations as follows:

- Juxtaposition: Resize each of the extracted objects $I_S^o$ and $I_M^o$ to be within a bounding box of $240 \times 240$ pixels (refer to the resizing method below), and position the resized objects side by side on a blank $640 \times 400$ image, so that the centers of the objects' bounding boxes are vertically centered and at the $\frac{1}{4}$ and $\frac{3}{4}$ marks on the horizontal axis.
- Replacement: Resize $I_S^o$ to be within the bounding box of $I_M^o$, and layer $I_S^o$ to the same position as $I_M^o$ in the inpainted image $I_M^p$, i.e. overlapping the centers of the two objects' bounding boxes.
- Fusion: Transfer the texture from $I_M^o$ to $I_S^o$ and overlay the resulting object upon the original subject image $I_S$.

Here, we have defined the operations only in one way, but as we pointed out earlier, subject and message images are interchangeable.

For resizing, let $B_O^w, B_O^h$ be the width and height of the bounding box of the object to be resized, and $B_T^w, B_T^h$ the width and height of the target bounding box ($240 \times 240$ in juxtaposition and the bounding box of the object to be replaced in replacement). We formulate the resizing procedure as follows:

1. Calculate the width and height ratios between the bounding boxes: $r_w = B_T^w / B_O^w$ and $r_h = B_T^h / B_O^h$.
2. If $r_w \leq r_h$, then $r = r_w$, otherwise $r = r_h$.
3. Resize if $r < \frac{3}{4}$ or $r > \frac{4}{3}$ (this is for using the original image whenever we can, in order to avoid decreasing the image quality).

## Finding Representative Photos of Concepts

At the first step, Vismantic searches in Flickr for photos that can represent well the subject and the message, respectively. Other concerns are *diversity*, *photo quality* and *(image processing) algorithm-friendliness*. We also pay attention to the copyright of photos, only retrieving photos under Creative Commons license with modification permission.

Both the subject and message can be a physical or abstract concept. For physical concepts, such as an object, pictures of the object itself or something closely related, i.e. its internal components or other objects interacting with it, are used to represent it. On the other hand, abstract concepts are represented by pictures of entities through connotation.

When searching in Flickr, the subject and message words are used as free text search, sorted by relevance. Photos with more than 15 tags are rejected, considering that too many tags might imply photographers' intention of boosting the rankings of their photos in every query. We also avoid photos tagged with 'illustration', 'painting', 'graphics', 'infographic', 'text', 'collage', 'scrapbook', 'photoshop', etc. The photos downloaded are of medium size: at most 640 pixels on the longer side.

Additionally, we take advantage of the 'related-tags'[4] provided by Flickr and one of our previous works (Xiao and Blat 2012) in order to improve the diversity of the search results for abstract concepts. (Xiao and Blat 2012) finds physical concepts that have the intended abstract concept as one of their stereotypical properties. This is achieved by retrieving strong associations from four semantic knowledge bases and subsequently filtering associations that are low in concreteness and imageability. Take the task in Fig. 1 as an example, the concept 'electricity' is concatenated with each of 53 physical concepts to form Flickr queries, such as "electricity storm", "electricity pylon", "electricity bulb", "electricity windmill", "electricity plant", "electricity outlet", etc. The photos retrieved by multiple queries are organized in groups, one group per query.

**User Filtering** The photos retrieved from Flickr might not be sufficiently representative for the concept of interest. Also, the photo quality might be low, due to, e.g., under/over exposure, blur, highlight, low resolution, colorization, or using a fisheye lens. Besides, as mentioned in the previous subsection, the image processing techniques currently used by Vismantic only work well with certain images. Photos having a recognizable object that is neither obscured nor too small, and is situated in a simple context, are preferred. At present, Vismantic needs a user to choose quality photos that are representative and algorithm-friendly.

## Preprocessing

Each of the three visual operations has distinct requirements for a pair of input images:

- Juxtaposition: *good object extraction* for both images.
- Replacement: *good object extraction* for one image, and having a *suggestive context* for the other.
- Fusion: *good object extraction* for one image, and having a *distinctive texture* for the other.

The above requirements have to be satisfied before applying visual operations, which can be computationally expensive. Furthermore, they prevent unpromising results early on, which drastically saves the effort needed for evaluating the final output, since the number of images in the final output without filtering is quadratic to the number of input images (each photo of a subject is paired with every photo of a message).

At this step, object extraction and inpainting are applied to all the photos retrieved from Flickr and selected by a user. Currently, Vismantic does not have automatic means to judge the quality of object detection, the indicative capacity of a context, or the distinctiveness of a texture.

**User Filtering** For each photo, the object/region extracted is shown to a user, who is asked to decide if it represents the corresponding subject or message and if it has a distinct texture which alone cues the concept. The inpainted image is also shown to the user, together with the question whether the image reminds him of the concept.

[4] www.flickr.com/services/api/flickr.tags.getRelated.html

# Evaluation and Analysis

To get a first idea of what Vismantic generates, we put it in a test consisting of five typical visual communication tasks, where only the authors interacted with the system. In this section, we first present the output and curation coefficient at each step of the workflow, along with our analysis of the output. Next, we reveal the major factors that cause a generated image to be uninterpretable or end up with unintended meanings. The five tasks are the following (subject and message words in italic):

1. *Electricity* is *green* (sustainable).
2. *Music* is *powerful*.
3. *Lipstick* is associated with *love*.
4. Heating system makes *house warm*.
5. *Earplug* reduces *noise*.

At the first step, the purpose is to find representative, diverse, high-quality and algorithm-friendly photos of concepts (subjects and messages). In the test, 50 photos were collected for each subject and message. For abstract concepts, which lead to multiple queries for searching in Flickr, the photos were collected by visiting the photo groups (one group per query) one by one and picking up the first unpicked photo (the photos in a group are sorted by relevance). The upper part of Table 1 shows the number of disqualified, qualified, selected and surprising photos for each concept. Averaging across all ten concepts, 46.4% of the photos retrieved from Flickr are qualified. The disqualified photos are divided into three categories, i.e. 'non-representative', 'non-algorithm-friendly' and 'low-quality'. Non-representativenessness, amounting to 35.2%, was the top reason for rejecting photos. Non-representative photos either lack relevance or represent a sense of a concept other than the one intended.

We selected photos from the qualified ones and only kept those that look quite different from each other. On average, around 9 photos were selected for each concept. We also noticed that there were novel representations of concepts among the retrieved photos (the row of 'surprising' in Table 1), which counts for 4.2%.

At the second step, preprocessing, object extraction and inpainting were applied to the photos selected in Step I, and the output is shown in the lower part of Table 1. Averagely speaking, good object extraction was found in 69.3% of the selected photos. The major types of incorrect object extraction include: only part of an object was extracted and the part was not recognizable; the object was not extracted at all but some other part of the photo instead, e.g., another object or part of the background; or the whole photo was extracted. Within the properly extracted objects, distinctive textures were not so common, counting for 20.5%. Some examples are green grassland, red lipstick, water, brick wall, flame and textile. Besides, only 22.7% of the selected photos had suggestive context around an object region. In many photos, the object was relatively big and the context was too small to be distinguishable; or the context could not hint at the object if it were removed. However, surprisingly, the

Table 1: Output of Step I Finding representative photos of concepts and Step II Preprocessing.

| | electricity | green | music | powerful | lipstick | love | house | warm | earplug | noise | avg. | %avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **photos retrieved** | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | | |
| **non-representative** | 13 | 6 | 6 | 25 | 22 | 30 | 2 | 26 | 17 | 29 | 17.6 | 35.2 |
| **non-algorithm-friendly** | 6 | 2 | 5 | 1 | 9 | 4 | 10 | 4 | 21 | 2 | 6.4 | 12.8 |
| **low-quality** | 3 | 0 | 9 | 3 | 1 | 3 | 3 | 1 | 3 | 2 | 2.8 | 5.6 |
| **qualified** | 28 | 42 | 30 | 21 | 18 | 13 | 35 | 19 | 9 | 17 | 23.2 | 46.4 |
| **surprising** | 0 | 2 | 0 | 5 | 5 | 2 | 0 | 1 | 3 | 3 | 2.1 | 4.2 |
| **selected (at Step I)** | 10 | 7 | 9 | 9 | 13 | 8 | 8 | 9 | 6 | 9 | 8.8 | 17.6 |
| **good object extraction** | 6 | 5 | 5 | 5 | 10 | 5 | 8 | 6 | 5 | 6 | 6.1 | 69.3 |
| **distinct texture** | 0 | 5 | 0 | 3 | 3 | 1 | 2 | 4 | 0 | 0 | 1.8 | 20.5 |
| **has-context** | 0 | 4 | 4 | 4 | 0 | 2 | 0 | 3 | 2 | 1 | 2 | 22.7 |
| **suggestive context** | 3 | 5 | 6 | 6 | 2 | 4 | 0 | 3 | 5 | 5 | 3.9 | 44.3 |

Table 2: Output of Step III Applying visual operations. gene. = generated, expr. = expressive, supr. = surprise, % = ratio between the two numbers ahead.

| | electricity-green | | | music-powerful | | | lipstick-love | | | house-warm | | | earplug-noise | | | avg. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | genr. | expr. | % | genr. | expr. | % | genr. | expr. | % | genr. | expr. | % | genr. | expr. | % | genr. | expr. | % |
| **juxtaposition** | 60 | 0 | 0 | 50 | 46 | 92 | 100 | 54 | 54 | 96 | 56 | 58.3 | 60 | 44 | 73.3 | 73.2 | 40 | 54.6 |
| **replacement** | 45 | 12 | 26.7 | 60 | 28 | 46.7 | 50 | 27 | 54 | 24 | 21 | 87.5 | 55 | 30 | 54.5 | 46.8 | 23.6 | 50.4 |
| **fusion** | 30 | 9 | 30 | 15 | 0 | 0 | 25 | 1 | 4 | 44 | 3 | 6.8 | 0 | 0 | 0 | 22.8 | 2.6 | 11.4 |
| **total** | 135 | 21 | 15.6 | 125 | 74 | 59.2 | 175 | 82 | 46.9 | 164 | 80 | 48.8 | 115 | 74 | 64.3 | 143 | 66.2 | 46.4 |
| | expr. | supr. | % | expr. | supr. | % | expr. | supr. | % | expr. | supr. | % | expr. | supr. | % | expr. | supr. | % |
| **surprise** | 21 | 21 | 100 | 74 | 30 | 40.5 | 82 | 40 | 48.8 | 80 | 80 | 100 | 74 | 32 | 43.2 | 66.2 | 40.6 | 61.3 |

errors in object extraction sometimes provided suggestive context. When only part of an object was extracted, the remaining part might be able to cue the object. For instance, in Fig. 1c, the smoke coming out of the power station was not extracted. Including these cases, suggestive context were found in 44.3% of the selected photos.

At the third step, applying visual operations, on average 143 images were generated for each task (Table 2). Most of them were juxtapositions and replacements, because distinctive textures were rare. Our primary evaluation criterion is whether a generated image expresses the meaning specified in a task. Averaging across all five tasks, 46.4% of the generated images were considered expressive.

Regarding if there is a general trend that one visual operation works better than another, there was no significant difference between juxtaposition and replacement. Both operations produced expressive images about half of the time, 54.6% and 50.4% respectively. The difference showed more in specific tasks. For instance, there was no expressive juxtaposition for Task 1. As seen in Table 2, it seems that fusion yields less than juxtaposition or replacement. However, a factor that has to be taken into account is that fusion can easily go wrong in the current implementation. In juxtaposition and replacement, the image processing involved are mainly resizing and positioning, while the quality of object extraction and the indicative capacity of context have been evaluated in the previous step, preprocessing. On the other hand, the parameters of the texture transfer technique used in fusion have to be fine tuned for each image for optimal performance, which is not yet available in Vismantic. In addition, the number of textures available for fusion was quite small. When there are more varieties of textures, one set of parameters that does not work well with one image may work for another, which could bring us more expressive images.

A few examples of the images generated for each task are shown in Fig. 4. More examples can be visited online[5]. Besides, we have a few interesting observations. Firstly, Vismantic sometimes generates "perfect images" (see Fig. 1a), when some visual features of two objects, such as size, shape, angle and lighting, match by coincidence. Secondly, fusion sometimes produces images of high artistic skill (see Fig. 1b for an example).

As in Step I, we counted the number of surprising ideas among the generated images, and found that on average 61.3% of the expressive images were surprising. The surprise came from novel representations of concepts and unexpected combinations of objects in terms of the concepts they denote/connote or the exact visual representations. Additionally, we call attention to Fig. 1c. The meaning of this image is not as straightforward as "the power station (covered by the leaves) is as green as the leaves", which is what we had envisioned. A plausible interpretation is "the leaves (or the concept of 'sustainability' accompanying it) replaces the traditional power station". Owing to the drastic contrast in size and solidity between leaf and power station in common sense, this image exemplifies immense boldness, which has not been explicitly modeled in Vismantic.

In the following subsection, we analyze why some of the generated images do not express the intended meanings.

## Failure Analysis

We have observed that a generated idea may fail mostly in three aspects, namely *semantic interaction*, *visual operation implementation* and *object affordance and composition*.

**Semantic Interaction** For some generated images, there seems to be no plausible interpretation, divergent interpretations, or an interpretation either the one not intended or

[5]http://vismantic.hiit.fi/examples/

(a) (b) (c) (d)

Figure 4: Examples of generated visual ideas. 4a: for Task 2 “music is powerful”, a singer replaces part of waves; 4b: for Task 3 “lipstick is associated with love”, the heads of a kissing couple are fused with a red lipstick; 4c: for Task 4 “heating system makes house warm”, a house is fused with a pair of crochet mittens; 4d: for Task 5 “earplugs reduce noise”, a helicopter replaces part of a man’s head with fingers stuck in the ears.

the opposite. For instance, Fig. 5a is a juxtaposition generated for Task 4 “heating system makes house warm”. It seems rather difficult to get the feeling that the house is being warmed up. Nonetheless, this is well achieved by the fusion of the two objects (Fig. 4c). Another example is shown in Fig. 5b, a replacement (a power station (without smoke) replaces a line of trees) generated for Task 1 “electricity is green”. The image looks like a power station in its natural surroundings, which is unable to allure viewers into thinking of other connections between the two objects, such as grass gives energy to a power station (see Fig. 1a for a comparison). Moreover, semantic interaction may not happen as expected for other reasons, such as objects having opposing emotional valence.

(a) (b)

Figure 5: Semantic interaction.

**Visual Operation Implementation** As explained earlier, fusion requires fine tuning the parameters of the inpainting algorithm, which is not available in Vismantic at present. The current resizing method used in replacement does not produce ideal results when the objects involved have quite different shapes. Besides, we noticed that additional constraints might be applied to visual operations, e.g., texture-based fusion should avoid objects with similar colors.

**Object Affordance and Composition** Fig. 6 shows two different light bulbs placed in the same context, both of which are replacements for Task 1 “electricity is green”. Fig. 6a works while Fig. 6b does not. The difference between the light bulbs is that one is for putting on a horizontal surface, such as the ground, and the other is for hanging vertically, such as from a ceiling. The context is a forest with the ground covered by grass and leaves. The bulb for the horizontal plane suits this context well, which suggests that it is the forest where the bulb gets energy. In contrast, the vertical light bulb can not connect to the forest in a similar way. This comparison reveals that two objects can only be connected meaningfully at certain parts, but not every part.

Besides, the left and right order (orientation) of two objects sometimes can not be arbitrary. Consider whether the idea is still effective if the singer in Fig. 4a turns his head to the opposite direction.

(a) (b)

Figure 6: Object affordance.

In Table 3, the numbers of different types of failure are presented. It shows that semantic interaction was a major cause of failure for all three visual operations. Failure of visual operation implementation occurred mainly in fusion and replacement, since juxtaposition has less constraints on resizing and positioning. Failure of object affordance and composition happened largely in replacement and juxtaposition, because the current implementation of fusion primarily replies on texture.

Table 3: Failure type. The ratio in parenthesis is against the number of disqualified images generated by each operation.

| | **juxtaposition** | **replacement** | **fusion** |
|---|---|---|---|
| **disqualified images** | 166 | 116 | 101 |
| **semantic interaction** | 158 (95.2%) | 65 (56.0%) | 50 (49.5%) |
| **visual operation implementation** | 0 (0.0%) | 29 (25.0%) | 52 (50.5%) |
| **object affordance & composition** | 8 (4.8%) | 21 (19.0%) | 0 (0.0%) |

## Conclusions and Future Work

This paper presents Vismantic, a semi-automatic system for generating visual ideas. The workflow it exemplifies has generality, in the sense that it starts from a conceptual task (described in text) and outputs visual compositions, which fit real-life practice. Vismantic takes advantage of both conceptual and visual creativity in its ideation. At present, with basic conceptual knowledge (semantic associations) and the first implementation of three visual operations (juxtaposition, replacement and fusion), it demonstrated the potential of producing images that are expressive, diverse and surprising.

Vismantic generates surprising ideas by using novel representations of concepts and unexpected combinations of objects in terms of the concepts they denote/connote or the exact visual representations. It also generates images with certain particular flavors, such as extreme boldness, though it is not supposed to have such sense. In the future, when deciding objects to be combined, additional effects, such as surprise, boldness and humor, can be considered.

For Vismantic to have a higher level of automation and generate more ideas that make sense, we have identified challenges in three areas:

- *Visual Resources*: sources of photos with high relevance and diversity, sources of distinctive textures and sources of indicative context;
- *Image Processing*: automatic means of selecting photos that are high-quality and algorithm-friendly, automatic means of tuning algorithm parameters, taking into account visual features (such as color, shape, orientation and camera angle) when applying the visual operations, and making use of more sophisticated image analysis to accurately locate objects in complex scenes;
- *Visual Semantics*: more visual knowldge, such as object affordance and the meanings of visual features (e.g., orientation, position and contrast), and the ability of interpreting images, i.e. simulating the interaction between all the meaning fragments generated by visual cues at various levels.

Last but not least, other visual operations can be added to Vismantic.

## Acknowledgments

This work has been supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733 (ConCreTe). We would like to thank Hannu Toivonen for his valuable suggestion and Flickr users who grant their photos Creative Commons licenses.

## References

Borji, A.; Sihite, D.; and Itti, L. 2012. Salient object detection: A benchmark. In Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; and Schmid, C., eds., *Computer Vision ECCV 2012*. Springer Berlin Heidelberg. 414–429.

Chen, T.; Cheng, M.-M.; Tan, P.; Shamir, A.; and Hu, S.-M. 2009. Sketch2photo: Internet image montage. In *Proceedings of the ACM SIGGRAPH Asia 2009*, SA '09, 124:1–124:10.

Cheng, M.-M.; Zhang, G.-X.; Mitra, N. J.; Huang, X.; and Hu, S.-M. 2011. Global contrast based salient region detection. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, 409–416.

Cheng, M.-M.; Mitra, N. J.; Huang, X.; Torr, P. H. S.; and Hu, S. 2015. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3):569–582.

Daisy, M.; Tschumperlé, D.; and Lézoray, O. 2013. A fast spatial patch blending algorithm for artefact reduction in pattern-based image inpainting. In *SIGGRAPH Asia 2013 Technical Briefs*, SA '13, 8:1–8:4.

De Smedt, T.; De Bleser, F.; Van Asch, V.; Nijs, L.; and Daelemans, W. 2013. Gravital: Natural language processing for computer graphics. In Veale, T.; Kurt, F.; and Forceville, C., eds., *Creativity and the Agile Mind: A Multi-Disciplinary Study of a Multi-Faceted Phenomenon*. Berlin: Mouton. 81–98.

Diakopoulos, N.; Essa, I.; and Jain, R. 2004. Content based image synthesis. In Enser, P.; Kompatsiaris, Y.; O'Connor, N.; Smeaton, A.; and Smeulders, A., eds., *Image and Video Retrieval*. Springer Berlin Heidelberg. 299–307.

Harrison, P. 2005. *Image Texture Tools*. Ph.D. Dissertation, Monash University.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated Collage Generation – With Intent. In *Proceedings of the International Conference on Computational Creativity*, ICCC '10, 36–40.

Lalonde, J.-F.; Hoiem, D.; Efros, A. A.; Rother, C.; Winn, J.; and Criminisi, A. 2007. Photo clip art. In *Proceedings of the ACM SIGGRAPH 2007*, SIGGRAPH '07.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing Appreciation in a Creative System. In *Proceedings of the International Conference on Computational Creativity*, ICCC '10, 26–35.

Norton, D.; Heath, D.; and Ventura, D. 2011. Autonomously creating quality images. In *Proceedings of the 2nd International Conference on Computational Creativity*, ICCC '11, 10–15.

Phillips, B. J., and McQuarrie, E. F. 2004. Beyond visual metaphor: A new typology of visual rhetoric in advertising. *Marketing Theory* 4(1-2):113–136.

Xiao, P., and Blat, J. 2012. Image the imageless: Harvesting connotation knowledge for visual expression. In *Proceedings of the 7th International Conference on Design Principles and Practices*.

Xiao, P., and Blat, J. 2013. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the 4th International Conference on Computational Creativity*, ICCC '13, 8–15.

# The Good, the Bad, and the AHA! Blends

**P. Martins[1], T. Urbančič[2,3], S. Pollak[3], N. Lavrač[3,2], A. Cardoso[1]**
[1] CISUC, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal
[2] University of Nova Gorica, Nova Gorica, Slovenia
[3] Jožef Stefan Institute, Ljubljana, Slovenia

**Abstract**

We present and discuss quality assessment of visual blends based on how humans perceive them. This work represents part of a wider study aimed at determining the fundamental characteristics of a good blend. Based on the obtained insights, we hope to make a more comprehensible explanation of some less clear and not fully described aspects of the conceptual blending mechanism that play a fundamental role in creative thinking. Additionally, we intend to bring these insights into the design of artificial creative systems.

## Introduction

*Conceptual blending* (CB) is a vital cognitive mechanism by which two or more mental spaces are integrated to produce new concepts (Fauconnier and Turner 2002). Blending is at the heart of the origin of ideas; a new idea or thought can be seen as an insight gathered from a *blend*, i.e., the result from integrating different mental spaces. Not unexpectedly, the complexity and the quality of blends can be quite heterogeneous. The human brain continuously attempts to blend different concepts either by using a quite uncomplicated web of mental spaces or a more refined and complex network. The majority of these attempts fail in producing good blends, especially because the blend neither has sufficient novelty nor it has immediate purpose (Turner 2014).

What makes us prefer one blend to another? Is it sufficient to require novelty and value under a given context? Can quality simply depend on the coherence of the blend and on the easiness to interpret it? As CB theory inevitably links with the phenomenon of creativity, that is, the ability of producing new, surprising, and valuable ideas or artifacts (Boden 1991), it is expectable to regard as good blends the ones which imply a more creative thinking. However, the "intuitive" nature of creativity hinders the construction of a system of strict and immediate rules to explain which mental spaces should be selected and how they should be integrated in order to achieve creative thinking. Therefore, giving a more elaborate answer to the question "What makes a good blend?" is challenging.

In this paper, we present and discuss quality assessment of visual blends based on how humans perceive them – as *good*, as *bad*, or as *surprising* and *thought-provoking* (*AHA!*) blends. This is part of a wider study whose fundamental goal is to understand what are the key characteristics of a good blend. By finding them, we hope to make a more understandable explanation of some less clear and less described aspects of the blending mechanism that play a fundamental role in creative thinking. Another goal of our work is to bring these insights into the design of artificial creative systems to improve creation and curation processes, especially when they rely on computational models of conceptual blending. We are particularly interested in contributing to the design of frameworks for creativity assessment. While there are already noteworthy works in the field (Ritchie 2001; Colton, Pease, and Ritchie 2001; Wiggins 2001; Colton 2008; Jordanous 2012), there is still room for improvement.

To the best of our knowledge, an analysis of blends based on human perception was only followed by Joy et al. when analyzing conceptual blending in advertising (Joy, F. Sherry Jr., and Deschenes 2009). The authors conducted interviews with 28 volunteers who had to interpret several advertisements by describing what they thought was their main messages and how they arrived at such interpretation. The advertisements used in the experiment provided clear examples of conceptual blending.

We make use of an online-survey questionnaire in which participants are asked to evaluate criteria that we assume to be related to the quality of blends. In the likeness of the aforementioned work, the examples given to the participants can be easily perceived as instances of conceptual blending.

The remainder of this paper is organized as follows. In the upcoming section, we overview the conceptual blending framework and discuss its relation with creativity. Then, we present the content of the survey and discuss its results. Finally, we present concluding remarks and discuss future research.

## Conceptual Blending and Creativity

Fauconnier and Turner originally proposed conceptual blending theory as an attempt to explain cognitive and linguistic phenomena such as metaphor, metonymy, and counterfactual reasoning (Fauconnier and Turner 1998), but later it was extended to describe and explain different cognitive phenomena related to the creation of ideas and meanings (Fauconnier and Turner 2002; Turner 2014).

## Mental spaces network

A key element in conceptual blending is the *mental space*, which corresponds to a partial and temporary knowledge structure created for the purpose of local understanding (Fauconnier 1994). Mental spaces differ from *frames*, which are more stable knowledge structures. In the CB framework, there is a network comprising at least four connected mental spaces, as depicted in Figure 1. Two or more of them correspond to the *input spaces*, which are the initial domains. A partial matching between the input spaces is constructed. This association is 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 maintains partial structures from the input spaces combined with an emergent structure of its own.

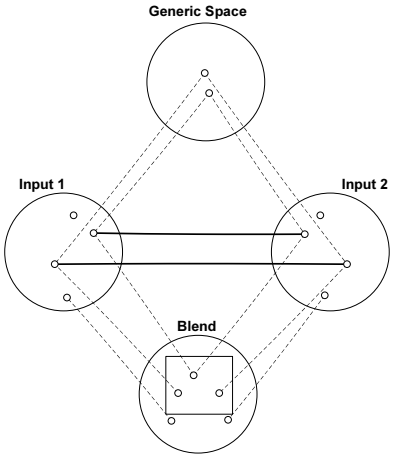



Figure 1: The original four-space conceptual blending network (Fauconnier and Turner 2002).

## Integration

*Integration* of input elements in the blend space results from three operations: *composition*, *completion*, and *elaboration* (Fauconnier and Turner 2002). Composition occurs when the elements from the input spaces are projected into the blend and new relations become available in the blended space. This implies projecting into the blend not only the matched elements but also other surrounding elements. Completion occurs when existing knowledge in long-term memory, i.e., knowledge from background frames, is used to generate 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. Elaboration is also known as "running the blend". There is not a pre-established order for these operations and several iterations may occur.

## Optimality principles

Integration is guided by *optimality principles*, which are responsible for generating consistent blends which in turn are more easily interpreted. Fauconnier and Turner (1998) provided a list of these principles:

**OP1** *Integration*: the blend must constitute a tightly integrated scene that can be manipulated as a unit. More generally, every space on the blend structure should have integration. In other words, the integration principle dictates that the blend must be recognized as a whole and as a new concept that is coherent.

**OP2** *Intensifying Vital Relations*: compress what is diffuse by scaling a single vital conceptual relation or transforming vital conceptual relations into others.

**OP3** *Maximizing Vital Relations*: create human scale in the blend by maximizing vital relations.

**OP4** *Topology*: 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. Put differently, the topology principle dictates that every element projected into the blend should maintain the same neighborhood relations as in the input space. This principle can be disregarded without having a major impact in the value of the blend, especially if we are dealing with free combinations, such as an imaginary object with a given goal (Pereira 2005).

**OP5** *Web*: manipulating the blend as a unit must maintain the web of appropriate connections to the input spaces easily and without additional surveillance or computation.

**OP6** *Unpacking*: the blend alone must enable the blend reader/observer to unpack the blend to reconstruct the inputs, the cross-space mapping, the generic space, and the network of connections between all these spaces. Unlike other principles, unpacking takes the perspective of the blend reader, i.e., someone who is not acquainted with the blend generation process.

**OP7** *Relevance*: all things being equal, if an element appears in the blend, there will be pressure to find significance for this element. Significance will include relevant links to other spaces and relevant functions in running the blend. In short, the relevance principle requires the existence of a reason for the blend to occur.

## Blends and creative thought

The theory built around conceptual blending inevitably deals with the phenomenon of creative thinking. The ability of producing new, surprising, and valuable ideas or artifacts comes frequently in advanced forms of conceptual blending (Turner 2014). Due to the "intuitive" nature of creative thinking, the construction of a comprehensive theory of such phenomenon is quite challenging. Conceptual blending theory, without being an exception, is sometimes vague and less prone to formalization when dealing with crucial aspects of creative thought. In particular, the framework does not explicitly deal with novelty and the optimality principles do not clearly dictate whether a blend is creative or not. However, it is a common assumption that novelty can result from the application of these principles.

Despite all these limitations, the conceptual blending framework provides not only a set of sound principles but

also a consistent terminology that can be used in creativity modeling. This has been a major motivation to consider the design of artificial creative systems based on computational approaches to conceptual blending.

## Looking for good blends

By understanding what humans perceive as a good blend, we hope to dissipate some of the vagueness surrounding the explanation of parts of the blending mechanism that are not fully described or even ambiguous. We are primarly interested in analyzing the relevance of the optimality principles, the selection of input spaces as well as the projection of elements. Our goal is not to establish solid rules to the blending process – as it would be incongruous with the theory – but to provide some hints to questions such as "How 'semantically far' should the input spaces be to produce a good blend?","Is there a correlation between the quality of blends and the number of elements for projection?", or "Are all the optimality principles required to produce good blends?". In the case of artificial creative systems, we also expect to find clues to questions such as "Are the typical relationships found in concept maps sufficient to infer the quality of a blend?", "How important is to include common sense knowledge (sensorial and subjective elements) in concept maps to achieve better blends?", or "To what extent is required to have a goal-driven blending to obtain better blends?".

It should be noted that we do not use any a priori definition of what is considered to be a good blend. Constructing such a definition is actually the goal of this study. Nonetheless, our work relies on the premise that good blends are creative to some extent, whereas the reciprocal is not necessarily true.

In this paper, we focus on visual blends. More accurately, we work with images depicting fictional hybrid animals. Examples of hybrid animals, such as *Pegasus* or *the lion man*, are often presented in the literature as well-known and/or ancient blends. There are also several experiments in the field of computational creativity involving the creation of hybrid animals (Pereira and Cardoso 2003; Neahus et al. 2014). In our case, we opted to analyze this particular type of blend due to the fact that hybrid animals tend to be easily perceived as a blend, i. e., the blend reader can recognize the input spaces and simultaneously identify a novel creature. Nevertheless, we will try to make generalizations from our observations rather than drawing conclusions that only hold for this type of blends.

### The survey

To assess the quality of blends, we conducted an online questionnaire survey in which approximately 100 participants judged 15 novel animals which are the result of blending anatomies from two different animals.[1] Each hybrid animal was depicted in one image/scene (see Figure 2). The author of all images but two is Arne Olaf (`http://gyyp.imgur.com/`). He uses *Adobe® Photoshop®* to create the hybrid creatures. The input images are put in two layers, adjusted in terms of size and unnecessary regions are removed. After that, he applies some common image processing techniques to make the transitions smoother.

Note that our focus is on blending at the conceptual level, overlooking aspects related to technical perfection. However, we are aware that technical perfection of a picture plays an important role in the perception of visual blends. This is why we decided to use blends with a similar level of quality in this respect - all chosen blends could be perceived as "good" as far as visual presentation is concerned. Moreover, the pictures share a similar rendering style. This enabled us to investigate other influential factors with more certainty as we ruled out rendering or poor presentation as a reason for bad human perception of a blend. This is particularly important in looking for findings that would hold also for other types of blends, not just the visual ones.

One may comment that there are no obviously bad blends in the dataset. This decision was based on a preliminary test done by ourselves, in which we noticed that there were big individual differences in the acceptance of blends, although the blends were all looking "nice" and the dataset presented good candidates for being well accepted. Our voting on blends was almost never unanimous, and this is why we wanted to investigate more thoroughly what could be expected on a bigger and more heterogenous population of subjects. With this in mind, "the good" and "the bad" blends from the title should be understood as "well accepted" and "not so well accepted" blends, showing the way towards creation of blends that will be well accepted by humans.

The criteria used in the survey cover some of the optimality principles criteria (e.g., by asking about coherence and consistency we are checking if a blended creature is perceived as having its own identity and corresponds to the optimality criterion of integration) as well as some criteria that define creativity, i.e., novelty, surprise, and value (Boden 1991). Thus, for each image depicting a hybrid animal, we asked the participants to rate the following criteria in a integer scale from 1 (the worst) to 5 (the best):

**OI** *Overall impression*;
**N/S** *Novelty/Surprise*;
**I** *Interestingness*;
**AA** *Aesthetic appeal*;
**C/H** *Comicality/Humor*;
**C/C** *Coherence/Consistency*;
**PF** *Evoques positive feelings*;
**NF** *Evoques negative feelings*;
**CIP** *Creative industries potential*.

The participants were also asked to provide a name to the hybrid creature as well us to inform us if they could easily recognize two distinct animals in the image. The latter question evokes the unpacking principle, i.e., the ability of the participant to reconstruct the input spaces.

### Survey results and discussion

Figure 3 depicts the median overall impression for each of the hybrid animals in the dataset. According to these results,

[1] Available at `http://animals.janez.me`.

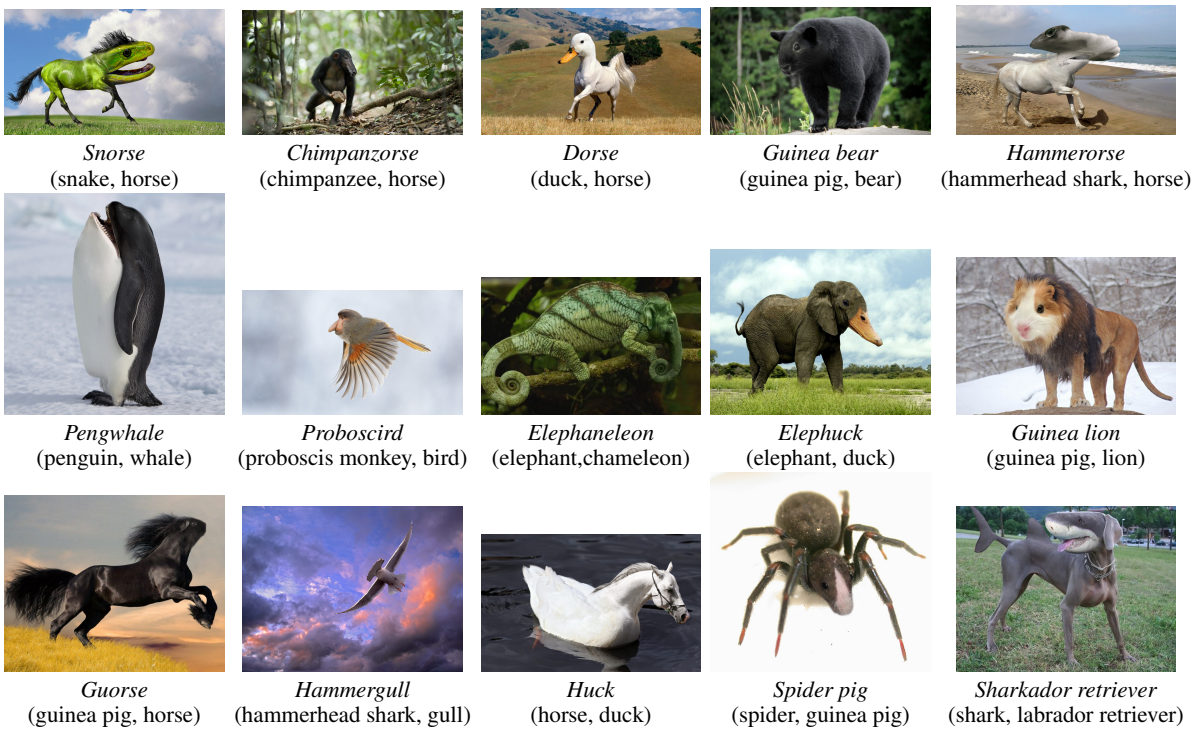



Figure 2: Hybrid animals dataset used in the online questionnaire. Each sub-caption contains the corresponding name of the blend as well as the input spaces. Names were coined by the authors of this paper or by the authors of the images and were not visible to survey participants. All blends were created by Arne Olaf, with the exception of *Sharkador retriever* and *Elephaneleon*, whose authorship is unknown. For a better visualization, some images were slightly cropped.

the top six best blends are *Guinea lion*, *Pengwhale*, *Guinea bear*, *Elephaneleon*, *Proboscird*, and *Dorse*, while *Spider pig*, *Hammerorse*, and *Guorse* were the least favorite blends.

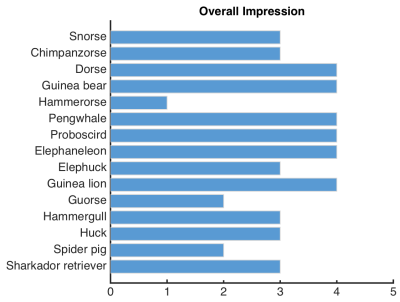



Figure 3: Overall impression (median) for each hybrid animal.

Figure 4 depicts a more detailed central tendency analysis of the survey results by including the median score for each criterion. Among the six best blends in terms of overall impression, *Guinea lion* and *Pengwhale* achieve the best overall scores. Out of the six best blends, five could be characterized by having a relatively big difference in the size of the original animal. It is also worth mentioning that *Guinea lion*, *Elephaneleon*, and *Pengwhale* are all in the best group with regard to the following criteria: novelty/surprise, interestingness, and coherence/consistency.

Regarding novelty, most blends achieved a median score of 4. The exceptions are *Chimpanzorse*, *Guinea bear* , *Elephuck*, *Guorse*, *Hammergull*, and *Huck*, which achieved a median score of 3. As it can be observed, high novelty does not necessarily lead to high overall impression. For instance, *Guinea bear* is a top-rated blend in terms of overall impression; however, its score in terms of novelty is among the lowest in the whole group. Conversely, *Hammerorse* has a high novelty score but a low overall impression score.

As pointed out by some respondents, novelty became more difficult to judge after a few images, as there were similar blends either in terms of input spaces or in terms of the elements for projection. This repetition and the fact that images were shown in fixed order might partially explain the lower scores obtained by *Elephuck*, *Hammergull*, or *Huck*.

Blends with a high overall impression score tend to have a high interestingness score. In fact, among the animals with the highest overall impression scores, *Dorse* is the only animal with an interestingness score of 3.

As for aesthetic appeal, we observe that blends with a low aesthetic appeal have a low overall impression score, whereas the most aesthetically appealing ones tend to have

high overall impression scores.

Coherence/consistency scores tend to be well aligned with the overall impression. The animals with the lowest overall impression scores – *Hammerorse*, *Guorse*, and *Spider pig* – have a consistency score of 2. For the remaining blends, with the exception of *Dorse*, the median coherence score coincides with the median overall impression score. We also observe that animals with higher overall impression scores tend to evoke more positive feelings, while animals with lower overall impression scores tend to evoke more negative sentiment.

Creative industries potential scores are not always in concordance with overall impression results. Similar results can be observed for the criterion comicality/humor. However, blends with the lowest overall impression scores are seen as having a low creative industries potential.

These results clearly show that the novelty alone does not guarantee the overall rating nor creative industry potential nor how interesting the blends are. The one considered to be one of the most novel ones is *Hammerorse*, but its has the lowest overall rating of all. Similarly, *Smorse* and *Sharkador retriever* are among the most novel ones; however, this is not reflected in their overall impression scores.

Another statistic analysis is given in Figure 5, which contains the correlation among pairs of criteria. As it can be readily seen, aesthetic appeal is strongly correlated with the overall impression ($\rho$= 0.8). There is also a strong positive association between overall impression and coherence ($\rho$=0.76). This result reflects the importance of the optimality principles, as they are responsible for defining coherent blends. The correlation between novelty and overall impression ($\rho$=0.47) corroborates our previous remarks: it is difficult to establish a straightforward association between these two scores.

We received also more than 20 comments related to the questionnaire. The majority of them were expressing satisfaction (having fun, enjoying the survey, etc.). The negative comments related especially to the fact that the survey was too long and that it got monotonous after a while. Specific points were commented, such as that coherence was difficult to judge and that novelty and humor were not applicable after the first few images. It was also proposed that comparing more animals at the same time would be better. A few people also explained which were their favorites, with *Pengwhale* being mentioned a few times. Some people provided more original explanations, e.g., "The horse duck was boring, because they are both vegetarian".

## The interview

We also conducted an interview with 4 people who took part in the survey. The main goal of the interview was to try to understand and discuss some of the ratings given by these participants. There was a general consensus that aesthetic appeal was an important requirement. For example, blending animals with similar types of coat – in terms of color, texture, or pattern – tends to result in aesthetically appealing blends. *Guinea bear* and *Guinea lion* were given as an examples of aesthetically appealing blends. *Snorse* was mentioned by one of the interviewees as another example of an aesthetically appealing blend, as there were no major differences between the snakeskin in the "snake part" and the coat in the "horse part" of the animal. *Pengwhale* was also a favorite among these participants. They enjoyed the fact that it was very difficult to establish a clear separation between "the whale part" and the "penguin part".

Participants took into account proportions when evaluating the aesthetic appeal. They presented *Guorse* as an example of a badly-proportioned blend: the proportions in the body of the horse require a head more elongated than the one of a guinea pig. In *Hammerorse*, the participants observed another instance of badly-proportioned parts. In this case, the head was seen as being too wide for the rest of the body.

One of the interviewees said to prefer the blend *Dorse* over the creature *Huck* because the head of *Dorse* has more resemblance with the head of a horse than the head of *Huck* has with the head of a duck. The interviewees also shared the opinion that surprise was required, but only to a certain extent. *Hammerorse* and *Spider pig* were given as examples of "too much surprise", which has a negative effect on the overall evaluation, whereas *Guinea bear* was presented as a blend with a minimal level of surprise.

Some participants suggested *Guinea lion* as a good example of comicality/humor due to the contrasting personalities of the animals given as input spaces. Although these mental spaces correspond to animals with similar coat and not so different anatomies, one is seen as a fierce creature, while the other one is a small harmless rodent

The participants emphasized the importance of recognizing the input spaces. However, there was the general idea that they enjoyed more when unpacking took time to occur.

## Good blends: input spaces, projection, and optimality principles

The level of novelty or surprise in a blend is partially dictated by the selection of input spaces and the choice of elements for projection. While the results from the survey do not show a direct association between novelty/surprise and the overall impression of the blend, it is somehow clear that both novelty and surprise are required to some extent. Selecting seemingly unrelated input spaces seems a good option only if the choice of elements for projection and subsequent tasks are able to deconstruct the idea that both concepts are unrelated. In this particular case, projection should be able to highlight various links between the two mental spaces that are less obvious instead of establishing a reduced number of more obvious connections.

Figure 6 depicts the concept similarity between different concepts used in the survey. Instead of using Linnaean taxonomy to compute the similarity between two animals, we opted for a more generic and elaborate measure that is able to generate more fine-grained results.[2] The concept similarity was calculated by applying the Personalized PageRank (PPR) (Haveliwala 2003) to ConceptNet. PPR is a variation of the standard PageRank algorithm used to rank nodes in a

[2]In our experiments with Linnaean taxonomy, the distances between animals were 5, 6, or 7.

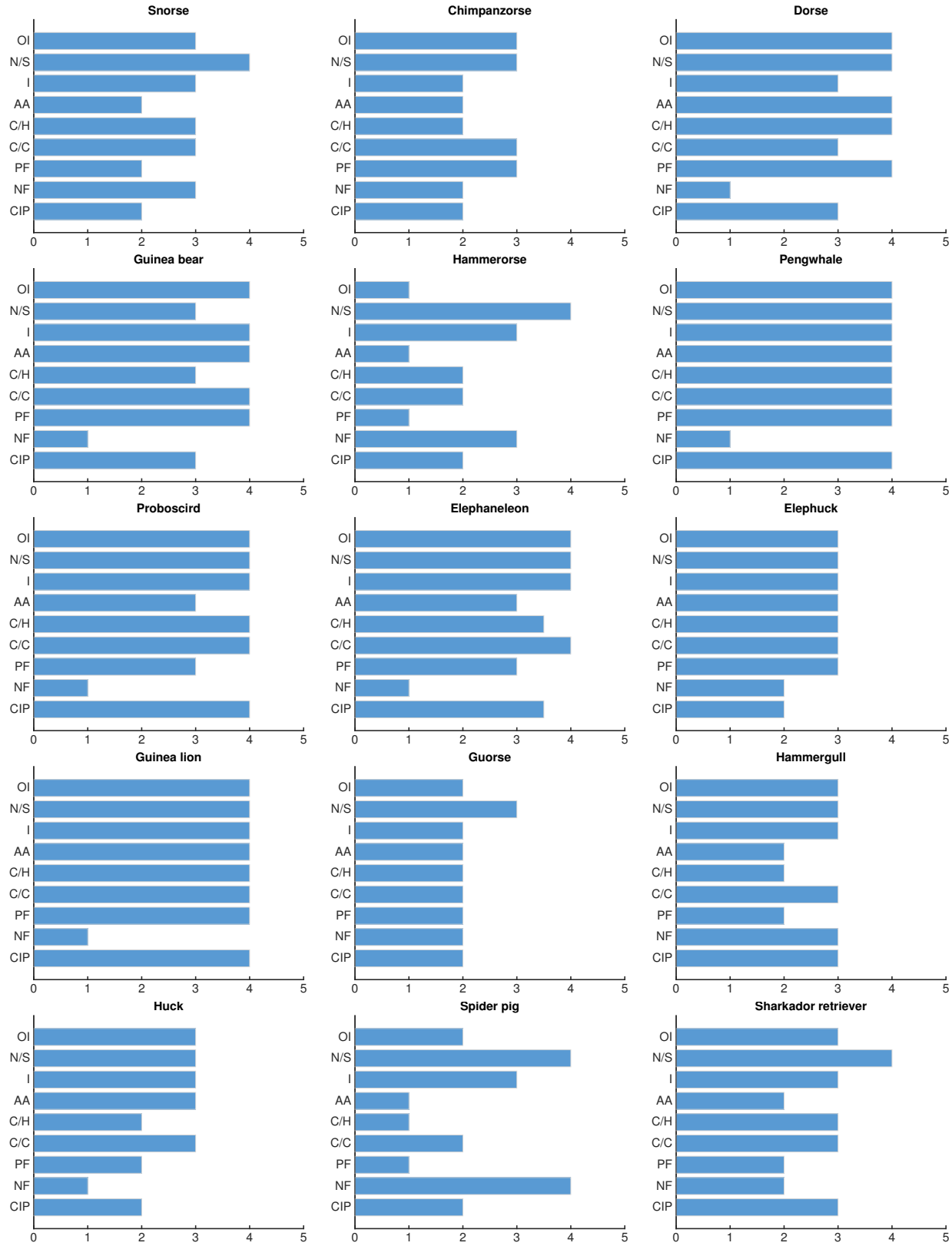



Figure 4: Survey results for each one the 15 hybrid animals. The bars represent the median score for each of the criteria.

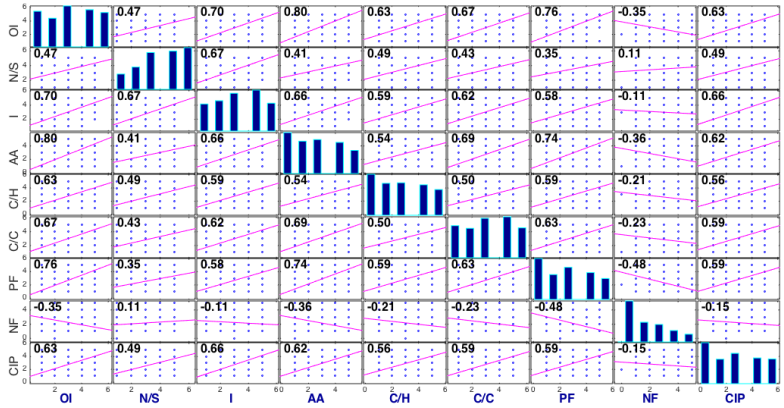



Figure 5: Matrix depicting correlations among pairs of criteria used in the survey. Non-diagonal elements contain scatter plots of the variable pairs. Diagonal elements contain histograms of the variables. The slopes of the least-squares lines in the scatter plots correspond to the displayed correlation coefficients.

network (Page et al. 1999). The PPR of a node $v$ in a network ($PPR\_v$) is a vector which, for each other node $w$ in the network, tells how simple it is to randomly walk from $v$ to $w$. It is calculated as a stationary distribution of the position of a random walker which starts its walk on node $v$ and at each step either (with probability $p$) randomly selects one of the connections leading out of its current node and travels along it or (with probability $1-p$) travels back to its starting location. In our experiment, $p$ was set to 0.85.

If the PPR of node $w$ according to node $v$ ($PPR\_v(w)$) is high, this means that the node $w$ is easy to reach from the departing node $v$. However, the path from $v$ to $w$ is not symmetric to the one from $w$ to $v$. Therefore, the similarity measure $s$ is proposed, where $s(v,w) = PPR\_v(w) + PPR\_w(v)$. In short, the higher the score the stronger the connection between the nodes and the higher the similarity between concepts.

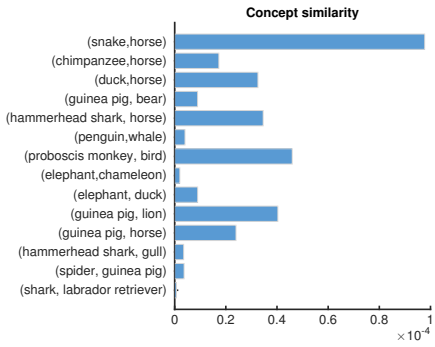



Figure 6: Similarity between the input spaces used in the survey.

For our work, we used the ConceptNet graph to calculate the similarity between two animals. We ran the PPR algorithm on the network to obtain the personalized PageRank vector for each of the animals in question. The personalized PageRank of a vertex is calculated iteratively by spreading the rank of the original vertex along its connections until the rank is no longer substantially changing.

This metric cannot be straightforwardly associated with the overall impression or novelty scores, as it does not faithfully reflect how semantically far the concepts are for a given observer. However, it suggests that sometimes seemingly unrelated input spaces (e.g, a horse (mammal) and a snake (reptile)) are sometimes more similar than two mammals (e.g., guinea pig and bear). We believe that exploring these less obvious similarities is a good starting point for the construction of high-quality blends.

Not unexpectedly, the results from the survey support the idea that all optimality principles are relevant, with the exception of topology (as already explained in the previous section). Integration is arguably the most important one and it should not be overlooked. It is necessary (but not sufficient) to dictate the coherence of the blend.

Figure 7 shows the percentage of affirmative answers to the unpacking question: "Can you easily recognize two distinct animals in the image?" for each one of the blends. In general terms, input spaces were easily recognized, although this task became more difficult when unpacking *Guinea bear* and *Guorse*, as the differences between the animal that provides the body and the blend are minimal. We believe that unpacking is a relevant principle, but it should not be given priority over other principles such as integration. On one hand, it allows the blend reader/observer to build his own interpretation of the blending process, which is fundamental to preform assessments from the perspective of the reader/observer. On the other hand, an immediate unpacking sometimes means a lack of surprise or novelty.

## Conclusions and Future Work

We presented and discussed an evaluation based on the human perception of visual blends. This research is part of a wider study which is oriented towards two major aims: (i)

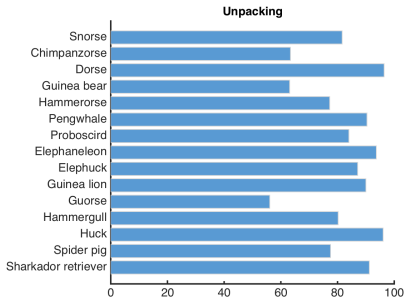



Figure 7: Unpacking: percentage of participants who responded affirmatively to the question "Can you easily recognize two distinct animals in the image?".

to help clarify some less clear and less described aspects of the blending mechanism which play a fundamental role in creative thinking; (ii) to improve the creation and curation processes in artificial creative systems.

Although we have only dealt with visual blends depicting hybrid animals, some of our observations can be applied to other types of blends. For instance, surprise and novelty are necessary but not sufficient to guarantee a high-quality blend. In fact, too much surprise is unfavorable if it affects the consistency of the emerging structure. The survey results also reflect the importance of having coherent blends, which emphasizes the importance of the optimality principles.

In this first experiment, we inevitably dealt with the specificities of visual blends, all being of similar technical quality, depicting hybrid animals. The results demonstrated that aesthetic appeal is an important criterion. Besides the quality of rendering, there are other aspects, namely symmetry and proportions, that influence aesthetic appeal. This may not be a relevant criterion when analyzing non-visual blends. However, since aesthetic appeal is related to symmetry and proportions, we argue that this criterion should be considered even when we are not working in the visual domain. For this reason choosing blends of similar technical quality, even at the cost of lower variety on the scale of all possible blends, seems to be the right decision, if we want to gain more insight into the conceptual level of blending. An interesting question remaining for future work is whether the results would be different if only textual descriptions or concept maps were given to the test subjects.

While the correlation of overall scores with other criteria in our experiment helps to identify the blends perceived as good or bad, the AHA! effect is correlated to the level of novelty, surprise, unpacking and creative industry potential. This will be further investigated with the analysis of the names given to the blends by the test subjects. Some of these names were very creative and reflected new qualities, existing in the blend while not being present in the input spaces. This will help us to understand the role of the emergent new structure reflected in such names, and might uncover the potential of blends to trigger the highly individual AHA! effect and human creativity.

## Acknowledgements

The authors acknowledge the financial support from the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the ConCreTe FET-Open project (grant number 611733) and the PROSECCO FET-Proactive project (grant number 600653). The authors thank also Jan Kralj for designing the concept similarity measure, and Janez Kranjc for the survey platform, which was developed within the WHIM FET-Open project (grant number 611560). We would also like to thank the survey participants for their time and their valuable comments.

## References

Boden, M. A. 1991. *The Creative Mind: Myths and Mechanisms*. Basic Books, Inc.

Colton, S.; Pease, A.; and Ritchie, G. 2001. The effect of input knowledge on creativity. In *Proceedings of the First Workshop on Creative Systems, International Conference of Case-Based Reasoning (ICCBR'01)*.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symp. on Creative Intelligent Systems*.

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.

Haveliwala, T. H. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. Technical Report 2003-29, Stanford InfoLab.

Jordanous, A. 2012. *Evaluating computational creativity: a standardised procedure for evaluating creative systems and its application*. Ph.D. Dissertation, University of Sussex.

Joy, A.; F. Sherry Jr., J.; and Deschenes, J. 2009. Conceptual blending in advertising. *Journal of Business Research* 62(1):39 – 49.

Neahus, F.; Kutz, O.; Codescu, M.; and Mossakowski, T. 2014. Fabricating monsters is hard - towards the automation of conceptual blending. In *Proceedings of the Workshop "Computational Creativity, Concept Invention, and General Intelligence"*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Pereira, F., and Cardoso, A. 2003. The horse-bird creature generation experiment. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour(AISBJ )* 1(3):257–280.

Pereira, F. 2005. *Creativity and AI: A Conceptual Blending approach*. Ph.D. Dissertation, University of Coimbra.

Ritchie, G. 2001. Assessing creativity. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science*, 3–11.

Turner, M. 2014. *The Origin of Ideas*. Oxford University Press.

Wiggins, G. 2001. Categorising creative systems. In *Proceedings of the Third Workshop on Creative Systems (IJCAI'03)*.

# Using Argumentation to Evaluate Concept Blends in Combinatorial Creativity

**Roberto Confalonieri[1], Joseph Corneli[2], Alison Pease[3], Enric Plaza[1] and Marco Schorlemmer[1]**

[1]Artificial Intelligence Research Institute, IIIA-CSIC, Spain
[2]Computational Creativity Research Group, Department of Computing, Goldsmiths, University of London, UK
[3]Centre for Argument Technology, School of Computing, University of Dundee, UK

**Abstract**

This paper motivates the use of computational argumentation for evaluating 'concept blends' and other forms of combinatorial creativity. We exemplify our approach in the domain of computer icon design, where icons are understood as creative artefacts generated through concept blending. We present a semiotic system for representing icons, showing how they can be described in terms of interpretations and how they are related by sign patterns. The interpretation of a sign pattern conveys an intended meaning for an icon. This intended meaning is subjective, and depends on the way concept blending for creating the icon is realised. We show how the intended meaning of icons can be discussed in an explicit and social argumentation process modeled as a dialogue game, and show examples of these following the style of Lakatos (1976). In this way, we are able to evaluate concept blends through an open-ended and dynamic discussion in which concept blends can be improved and the reasons behind a specific evaluation are made explicit. In the closing section, we explore argumentation and the potential roles that can play at different stages of the concept blending process.

## Introduction

A proposal by (Fauconnier and Turner, 1998) called *concept blending* has reinvigorated studies trying to unravel the general cognitive principles operating during creative thought. According to (Fauconnier and Turner, 1998), concept blending is a cognitive process that serves a variety of cognitive purposes, including creativity. In this way of thinking, human creativity can be modeled as a blending process that takes different mental spaces as input and blends them into a new mental space called a *blend*. This is a form of *combinatorial creativity*, one of the three forms of creativity identified by Boden (2003). A blend is constructed by taking the existing commonalities among the input mental spaces (called the generic space) into account, and by projecting the structure of the input spaces in a selective way. In general the outcome can have an emergent structure arising from a non-trivial combination of the projected parts. Different projections lead to different blends and different generic spaces constrain the possible projections.

This poses challenges from a computational perspective: large number of possible combinations exhibiting vastly different properties can be constructed by choosing different input spaces, using different ways to compute the generic space, and selecting projections. Within the Concept Invention Theory project[1] (COINVENT), we are currently developing a computational account of concept blending based on insights from psychology, Artificial Intelligence (AI), and cognitive modelling (Schorlemmer et al., 2014). One of our goals is to address this combinatorial nature. One potential outcome of this work is a deeper understanding of the way combinatorial creativity works in general.

The formal and computational model for concept blending under development in COINVENT (Bou et al., 2014) is closely related to the notion of *amalgam* (Ontañón and Plaza, 2010). Amalgamation has its root in case-based reasoning and focuses on the issue of combining solutions coming from multiple cases. Assuming the solution space can be characterised as a generalisation space, the amalgam operation combines input solutions into a new solution that contains as much information from the two inputs solutions as possible. When input solutions cannot be combined, amalgamation generalises them by dropping some of their properties. This process of generalisation and combination can be expensive from a computational point of view, depending on the search space to be explored.

The amalgam-based approach for computing blends makes explicit the combinatorial nature of concept blending, which raises the issue of evaluating and selecting novel and valuable blends as opposed to those combinations that lack interest or significance. Although Fauconnier and Turner (1998) suggest a number of qualitative criteria that can be used for evaluating concept blends, it is not straightforward to chararacterise them in a computational model.

In this paper, we propose to explore an argumentative approach to understanding and evaluating the meaning, interest, and significance of concept blends. Specifically, we propose to view evaluating blends as a process of *argumentation*, in which the specifics of a blend are pinpointed and opened up as issues of discussion. Our intuition is that in the context of new ideas, proposals, or artworks, people use critical discussion and argumentation to understand, absorb and evaluate. We also consider the constructive roles that argumentation can play in concept blending.

Computational argumentation models have recently appeared in AI applications (Bench-Capon and Dunne, 2007;

[1]See `http://www.coinvent-project.eu` for details.

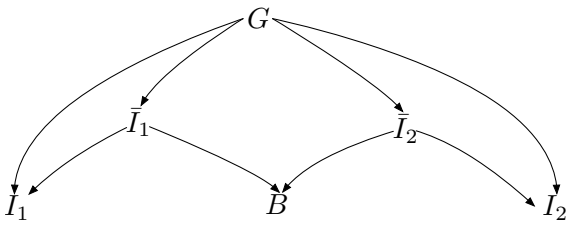



Figure 1: An amalgam diagram with inputs $I_1$ and $I_2$ and blend $B$ obtained by combining $\bar{I}_1$ and $\bar{I}_2$. The arrows indicate generalisation.

Rahwan and Simari, 2009), and we believe that incorporating argumentation can foster the development of a fuller computational account of combinatorial creativity. The current paper develops these themes at the level of (meta-) design; implementation is saved for future work.

### Roles of Argumentation in Concept Blending

Consider the *amalgam* diagram modeling the concept blending process (Figure 1): two input spaces $I_1$, $I_2$, two of their possible generalisations $\bar{I}_1$, $\bar{I}_2$, which have a generic space $G$ and blend $B$. When two input spaces cannot be combined because they do not satisfy certain criteria, the inputs have to be generalised for omitting some of their specifics. The combination of each specific pair $\bar{I}_1$, $\bar{I}_2$ yields a blend.

Informally, we can imagine argumentation taking place at various points in the amalgam diagram. In general this would happen in response to indeterminacy, that is, when some features of the diagram are underdetermined. We foresee that argumentation can be used:

**a.** to express opinions or points of view that can be used for guiding the selection/omission of specific parts of the input spaces; in particular, to select a specific pair of generalisation $\bar{I}_1$, $\bar{I}_2$ of the input spaces in the blending process;

**b.** to provide a computational setting for modeling discussions around the quality of a creative artefact, with the aim of evaluating and refining the generated blends.

In the first case, arguments would be about generalisation, i.e. which features should be preserved from $I_1$ and which features should be preserved from $I_2$. More complex inferences could be involved, for example in a case where $I_1$ is fixed, and constraints and various optimality criteria on the blend are imposed, which then yield various constraints on what the other input $I_2$ should be. We return to this point in the discussion section, and we focus for the most part on the second case.

In the second case, argumentation would be used to evaluate a range of blends, and the evaluation is carried out *post hoc*, by a variation of try-it-and-see. A range of blends are trialled, each one bringing out different (un)intended meanings. The evaluation is modeled as an argument, or dialogue in which the specifics of a blend are pinpointed and opened up as issues of discussion. This dialogue can be considered as an introspective evaluation, although it usually takes place among several parties as a means for the social development and understanding of creative artefacts. In this paper, we focus on this role.

### Our Approach

To exemplify our approach, we take the domain of computer icons into account. We assume that concept blending is the implicit process which governs the creative behavior of icon designers who *create* new icons by blending existing icons and signs. To this end, we propose a simple semiotic system for modeling computer icons. We consider computer icons as combinations of signs (e.g. document, magnifying glass, arrow etc.) that are described in terms of *interpretations*. Interpretations convey *actions-in-the-world* or *concepts* and are associated with shapes. Signs are related by sign-patterns modeled as qualitative spatial relations such as *above*, *behind*, etc. Since sign-patterns are used to combine signs, and each sign can have multiple interpretations, a sign-pattern used to generate a computer icon can convey multiple intended meanings to the icon. These are subjective interpretations of designers when they have to decide what is the best interpretation an icon can have in the real world. In this paper, we show how the intended meaning of new designed (blended) icons can be evaluated and refined by means of Lakatosian reasoning.

## Background

### Computational argumentation

Computational argumentation in AI aims at modeling the constitutive elements of argumentation, that are i) arguments, ii) attack relations modeling conflicts, and iii) acceptibility semantics for selecting valid arguments (Bench-Capon and Dunne, 2007; Rahwan and Simari, 2009).

The most well-known computational argumentation framework is due to Dung (1995). Dung defines an abstract framework to represent arguments and binary attack relations, modeling conflicts, by means of a graph. He defines different acceptibility semantics to decide which arguments are valid and, consequently, how conflicts can be resolved (Figure 2).

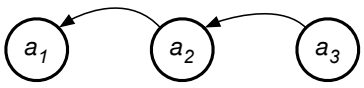



Figure 2: Dung framework example: Nodes represent arguments and edges (binary) attack relations. Argument $a_1$ is attacked by $a_2$ which is attacked by $a_3$. Thus, $a_2$ is defeated and $a_1$ can be accepted. $a_3$ is also accepted.

Abstract argumentation frameworks do not deal with how arguments are generated and exchanged. They merely focus on attack relations between arguments and acceptibility semantics. However, the intrinsic dialectical nature of argumentation is fully explored when an explicit argumentation process is considered. Then, the purpose of a dialogue becomes essential to determine how arguments should be generated and exchanged, and how a dialogue should be structured (Walton and Krabbe, 1995).

### Lakatosian argument and dialogue

Lakatos (1976) was a philosopher of mathematics who developed a model of argument, presented as a dialogue, to

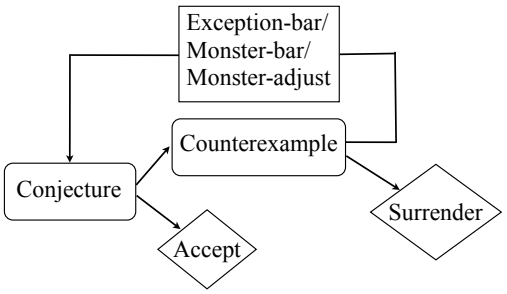



Figure 3: Our interpretation of Lakatos's game patterns.

describe ways in which mathematicians explore and develop new areas of mathematics. In particular, he looked at the role that conflict plays in such explorations, presenting a rational reconstruction of a dialogue in which claims are made and counterexamples are presented and responded to in various different ways. His resulting model describes conceptual continuity and change in the growth of knowledge, and contains dialogue moves, or methods, which suggest ways in which concepts, conjectures and proofs are fluid and open to negotiation, and gradually evolve via an organic process of interaction and argument between mathematicians. These dialogue moves are:

**Surrender** consists of abandoning a conjecture in the light of a counterexample.

**Piecemeal exclusion** is an exception-barring method that deals with exceptions by excluding a class of counterexamples, i.e., by generalising from a counterexample to a class of counterexamples which have certain properties.

**Strategic withdrawal** is an exception-barring method that uses positive examples of a conjecture and generalises from these to a class of object, and then limits the domain of the conjecture to this class.

**Monster-barring/monster-adjusting** is a way of excluding an unwanted counterexample. This method starts with the argument that a 'counterexample' can be ignored because it is *not* a counterexample, as it is not within the claimed concept definition. Rather, the object is seen as a monster which should not be allowed to disrupt a harmonious conjecture. Using this method, the original conjecture is unchanged, but the meaning of the terms in it may change. Monster-adjusting is similar, in that one reinterprets an object in such a way that it is no longer a counterexample, although in this case the object is still seen as belonging to the domain of the conjecture.

The moves above are not independent processes; much of Lakatos's work stressed the interdependence of creation and justification. These moves describe the evolution of both arguments and conclusions in mathematics, and as such constitute argument patterns, or schemes. However, they are a rational representation of exchanges between mathematicians and describe dynamic, rather than static arguments, presented as a dialogue. Thus, they also have temporal structure, and can be seen as a dialogue game, in which at any point various dialogue moves are applicable (see (Pease et al., 2014) for a description of Lakatos's methods in these terms). The fact that we include negotiations over definitions and changes in the conclusions being argued means that it is difficult to apply traditional abstract argumentation frameworks, which assume that such aspects are stable. However, we can see some of the moves in terms of Dung's framework: for instance if an initial argument for a conjecture forms $a_1$ in Figure 2, then $a_2$ might be a counterexample to the conjecture, and $a_3$ might be the monster-barring move.

The Lakatosian way of conceiving the reasoning as an open-ended discussion about a problem suggests that we can exploit Lakatos's moves for structuring dialogues for the evaluation of creative artefacts. Evaluation in creativity is not a static and rigid process, and the discussion should flow in a dynamic way. As such, in this paper, we propose to use Lakatosian reasoning to model the negotiation about the intended meaning of generated blends (icons). Figure 3 shows the dialogue game we will adopt to model these dialogues. For another formal framework of dialogue games for argumentation see (Prakken, 2005).

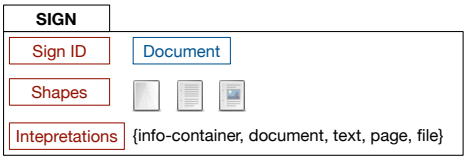
## Icons and Signs

We follow a semiotic approach to specify the intended meaning of computer icons. Semiotics is a transdisciplinary approach that studies meaning-making with signs and symbols (Chandler, 2004). Although it is clearly related to linguistics, semiotics also studies other forms of non-linguistic sign systems and how they may convey meaning; this includes not only designation, but also analogy, and metaphor. Although some people may regard Peirce's Sign Theory as the origin of semiotics, Saussure founded his semiotics (semiology) in the social sciences. Currently, cognitive semiotics and computational semiotics take their own perspectives on the relation between sign and meaning-making. In this paper, we take a semiotic approach to describe computer icons in the sense that icons, as a spatial pattern of shapes, are viewed as signs, and compositions of signs are interpreted to convey a meaning, as when we say 'this icon means the download is still active'.

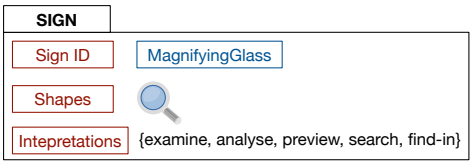
The shapes recurrently used in icons are interpreted as *signs*; screens, magnifying glasses and folders are examples of signs. A magnifying glass sign can be used in different icons in such a way that its meaning is *context-dependent*, that is, it depends on other signs related to it in different icons. We associate to each sign a set of *interpretations*, that encode the kinds of intended meaning associated to that sign as *actions-in-the-world* or *concepts*.

An icon is represented as a pattern defined by a collection of signs and qualitative spatial relations like *above*, *behind*, etc. We can find patterns of meaning that are shared among different icons by analysing recurrent patterns of signs and their spatial relation. We call them *sign patterns*. A sign pattern has an associated collection of *interpretations* that encode the intended meanings associated to that sign pattern.

Signs, sign patterns, and interpretations, which we will use in the paper, can be built by analysing and annotating existing libraries of computer icons. As we shall see, the inherent polysemy of signs, sign patterns and icons opens the way to use arguments for evaluating the quality or adequacy of new icons created by concept blending.

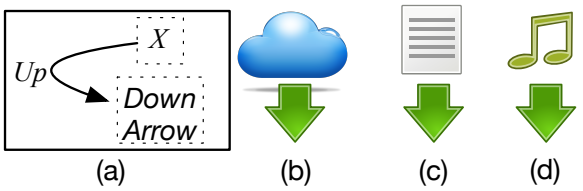



(I) The structure of the DOCUMENT sign, including associated shapes and interpretations.

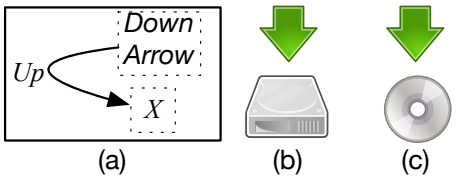



(II) The structure of the MAGNIFYINGGLASS sign, including associated shapes and interpretations.

Figure 4: Example of signs




(I) (a) the sign pattern FROM-DOWNARROW with three examples of the pattern where X is a sign for (b) cloud content, (c) document content, and (d) audio content.




(II) (a) the sign pattern DOWNARROW-TOWARD with two examples of the pattern where X is a sign for (b) a hard disk storage, and (c) an optical disk storage.

Figure 5: Example of different sign patterns used with the same sign DOWNARROW

## A semiotic system for icons

In this section, we will formalise the notions presented above. A *sign* $S$ is a tuple $\langle id, \mathcal{F}, \mathcal{A} \rangle$ where $id$ is a sign identifier, $\mathcal{F}$ is a set of shapes embodying the sign $S$ and $\mathcal{A}$ is a set of interpretations. We use $\mathcal{S}$ to denote the available set of signs. Figure 4 provides two examples. Figure 4I shows the structure of the DOCUMENT sign, with several shapes embodying the sign, and a list of interpretations that express how this sign is used in different ways to convey meanings such as *info-container, document, text, page, file*. Intuitively, this means that the shapes used in the icons are sometimes interpreted as a document and other times as a page, etc. Moreover, the specific shapes can be used interchangeably to embody a DOCUMENT, i.e. there is no clear distinction, regarding the shapes, between document vs. page vs. file. Another example of a sign is the MAGNIFYINGGLASS shown in Figure 4II, with interpretations *examine, analyse, preview, search*, and *find-in*.

We will also describe a library of annotated *icons* $\mathcal{I}$, where each icon $I \in \mathcal{I}$ consists of two parts: (1) a spatial configuration of signs and (2) the intended meaning of that icon. For instance, in Figure 5I, the icon (b) has the spatial configuration of a 'cloud on top of a downward-arrow' and its meaning is 'downloading content from the cloud'.

## Sign patterns

In our framework, sign patterns relate signs in icons using spatial qualitative relationships such as *above*, *behind*, *up*, *down*, *left*, etc. We assume that these relationships are represented as binary predicates, *Above*(X,Y), *Up*(X,Y), etc., where X and Y are variables ranging over signs in $\mathcal{S}$. For our current purposes, we use the qualitative spatial relationships defined in (Falomir et al., 2012).

Let us consider two examples of sign patterns that include the DOWNARROW sign. DOWNARROW has a vertical downward-pointing arrow shape and is associated with the interpretations {*down*, *downward*, *downloading*, *download-from* and *download-to*}. The sign pattern called FROM-DOWNARROW (shown in the schema labelled (a) in Figure 5I) uses the qualitative spatial relationship *up* between a variable X and the sign DOWNARROW. Examples (b), (c) and (d) in Figure 5I illustrate the intuitive meaning of the sign pattern FROM-DOWNARROW: 'downloading X'. Thus, example (b) refers to downloading cloud content, (c) document content, and (d) audio content.

The inherent asymmetry of arrows in general, and arrow signs particularly, can be appreciated when considering the opposite spatial relation, when the sign DOWNARROW is "up" from another sign (Figure 5II). Then, the sign pattern DOWNARROW-TOWARD is used to mean that X is the destination of the downloading. Example icons (b) and (c) are intended to mean that the data being downloaded (whose type or origin is now elided) is to be stored in a destination such as a hard disk or an optical disk.

## Evaluating Blends using Argumentation

As briefly described previously, the amalgam-based computation of concept blending amounts to combine different input spaces into a new space, called blend, by taking the commonalities of the inputs into account, by generalising some of their specifics and by projecting other elements. In the following, we describe how concept blending can account for modeling the creative process of a designer of computer icons.

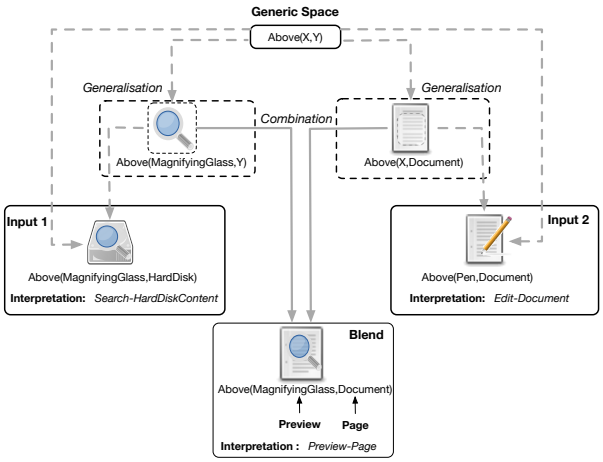



Figure 6: Generating an icon interpreted as *Preview-Page* through amalgam-based concept blending.

## A design scenario

Assume a designer is looking for creating a new icon with the intended meaning of previewing a document or a page. The creation of such icon can be achieved by the following amalgam-based concept blending process (Figure 6). In addition to the DOCUMENT and MAGNIFYINGGLASS signs, we assume we have available a HARDDISK sign and a PEN sign which have already been used to make icons.

**The input mental spaces.** The input mental spaces of the designer are an icon of a hard-disk with a magnifying glass hovering above it, whose meaning is *Search-HardDiskContent*, and an icon of a document with a pen above it, whose meaning is *Edit-Document*.

**The generic space.** The sign pattern *Above*(X,Y) is used in both icons. The first icon contains the relation *Above*(MAGNIFYINGGLASS, HARDDISK) between the MAGNIFYINGGLASS and the HARDDISK, and the second contains the relation *Above* (PEN,DOCUMENT) between the PEN and the DOCUMENT.

**Further generalisation.** Two generalisation steps are needed: *Above*(MAGNIFYINGGLASS,HARDDISK) $\rightarrow$ *Above(*MAGNIFYINGGLASS,Y*)*; correspondingly, *Above*(PEN,DOCUMENT) $\rightarrow$ *Above*(X,DOCUMENT).

**Combination via variable substitution.** We combine the schemas *Above*(MAGNIFYINGGLASS, Y) and *Above*(X, DOCUMENT) via [X/MAGNIFYINGGLASS, Y/DOCUMENT]. The icon of a page with a magnifying glass hovering above it is generated.

**The intended meaning.** The designer associates to the icon the intended meaning of *Preview-Page*, by selecting the interpretations (*Preview*, *Page*) for the MAGNIFYINGGLASS and DOCUMENT signs.

In this case, the designer decided that the intended meaning of *Above*(MAGNIFYINGGLASS, DOCUMENT) is *Preview-Page*, that is, a page can be examined without opening it. However, during the creative process, the designer could have generated other blends, not only by combining other signs, but also by selecting different interpretations associated to the MAGNIFYINGGLASS and DOCUMENT signs. For instance, the icon in Figure 7 still represents a page with a magnifying glass hovering above it, but it has been given a different intended meaning.

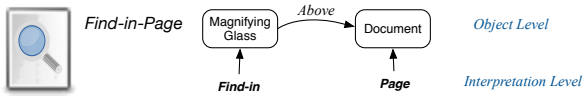



Figure 7: An example of interpreting the sign pattern of an icon as *Find-in-Page*.

The meaning of a blended icon cannot simply be considered right or wrong: interpretation depends on different points of view. Thus the evaluation of whether it is useful or valid for a specific purpose can be the object of a discussion.

## Arguments about intended meanings

In the icon domain, arguments may include a clear interpretation of any constituent signs in the icon if it is a composition of signs, or a good fit with other icons in the icon set.

For example, we can consider a counter-argument, i.e. an argument that *attacks* the interpretation $a_1$ "magnifying glass above document means *Preview-Page*" in Figure 6, to be phrased as follows:

$a_2$ : "However, the icon in Figure 6 can also be interpreted to mean *Find-in-Page*."

The rationale is that the MAGNIFYINGGLASS sign can often be understood as *finding* or *searching* for something. Thus, the icon can be also interpreted as *Find-in-Page* by associating the interpretation *find-in* instead of *preview* for the same sign MAGNIFYINGGLASS (Figure 7).

This attacking argument can be made at an abstract/conceptual level, for instance, by taking other possible blends of the DOCUMENT and MAGNIFYINGGLASS signs related by the sign pattern *Above*(X,Y) into account. Or, alternatively, if there is an icon library that contains an icon that 'satisfies' the argument above, then this attacking argument can be supported by a specific counterexample. Any of these two forms of attack evaluates negatively the icon in Figure 6. Therefore, if there are several alternative designs for a new icon, this attacking argument diminishes the degree of optimality/adequacy of that design with respect to alternative designs.

The original interpretation can be defended, as usually done in computational argumentation models, by a new argument that attacks the attacking argument $a_2$. For instance, the designer may say:

$a_3$ : "The icon in Figure 6 can only be interpreted differently if MAGNIFYINGGLASS is understood to mean *find-in* instead of *preview*. However, the other icons in

my library use MAGNIFYINGGLASS to mean *preview*, not *find-in*."

Argumentation semantics can then be used, once a network of arguments is built, to determine the outcome. For instance whether argument $a_1$, the original interpretation, is defeated or not can be determined as follows (Figure 2): in this example $a_3$ has no attack, so it is undefeated, which means it defeats $a_2$; since $a_2$ is defeated, the attack against $a_1$ is invalid and $a_1$ is undefeated (i.e. is accepted).

Arguments about the intended meanings of an icon can be embedded in a dialogue modeled in terms of Lakatos's moves and the dialogue pattern shown in Figure 3.

## Lakatosian reasoning for blend evaluation

Here we present a Lakatos-style dialogue between two players, a proponent $P$ and an opponent $O$. The goal of each player is to persuade the other player of a point of view, in this paper, the intended meaning of a new blended icon. In such a setting, we expect to see negotiations over the meaning of an icon take place between experts and novices, or between people designing icons and people using (interpreting) them, or various combinations.

To discuss a given icon using Lakatosian reasoning, we assume that an initial conjecture is about the interpretation of an icon usually being an *action-in-the-world* or a *concept*, together with an example of a particular icon and a particular interpretation. The conjecture could be constructed by inductive generalisation.

**Example 1.** In this example, Lakatosian reasoning is used for discussing the intended meaning of a new icon generated by concept blending:

**$P_1$:** "An icon with a magnifying glass over a page means *Preview-Page*" (Conjecture)
**$O_1$:** "I disagree, this icon (Figure 7) means *Find-in-page*." (Counterexample)
**$P_2$:** "No, this is a different case because the magnifying glass must be over pages with text on them to magnify (it shows what we're about to magnify)." (Monster-barring)

After this dialogue, it is agreed that the intended meaning of the icon is *Preview-Page* and the icon itself has been clarified. Alternatively, the proponent and the opponent could make a different evaluation by following different moves. For instance, if the proponent accepts the counterexample, then the intended meaning of the icon can be refined due to piecemeal exclusion:

**$P_1$:** "An icon with a magnifying glass over a page mean *Preview-Page*" (Conjecture)
**$O_1$:** "I disagree, this icon (Figure 7) means *Find-in-page*." (Counterexample)
**$P_3$:** "Ok, only icons with a magnifying glass over a page with text mean *Preview-Page*". (Piecemeal exclusion)

After this dialogue the intended meaning about the new icon has been changed by modifying the conjecture and taking the counterexample into account.

Sometimes players have different points of view due to the sign patterns they have used in their concept blending.

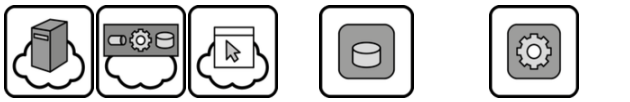



Figure 8: Interpreting the design of cloud icons[2]

**Example 2.** Let us imagine that the proponent has generated an intended meaning for an icon using the FROM-DOWNARROW sign pattern, whereas the opponent has used the DOWNARROW-TOWARD pattern (Figure 5 illustrates these cases). The two players can engage in the following dialogue:

**$P_1$:** "Look at icons in Figure 5I, icons with a DOWNARROW relate to content." (Initial Conjecture)
**$O_1$:** "The icon in Figure 5IIb has a DOWNARROW but doesn't relate to content." (Counterexample)
**$O_2$:** "The icon in Figure 5IIc also has a DOWNARROW but doesn't relate to content." (Counterexample)
**$P_2$:** "The conjecture is right because the two examples actually do relate to content as they are to do with storage and content is part of storage." (Monster-adjusting)

In this case, the proponent excludes the counterexamples using monster-adjusting, and reinterpreting them in a way that they are not counterexamples anymore.

A conjecture might even be at a higher level, for asserting that a particular metaphor is appropriate or inappropriate.

**Example 3.** For example, someone who is familiar with the 'gear means adjust setting' metaphor in one program may be comfortable with it in another program:

**$P_1$:** " An icon containing the 'gear' sign is a good one for Settings, because it invokes the idea of a gear change on a bicycle" (Initial Conjecture)
**$O_1$:** "The 'gear' sign does not invoke the idea of a gear change on a bicycle *from my point of view*." (Counterexample)
**$P_2$:** "Ok, you're right, it does not invoke the idea of a gear change on a bicycle, but it is often used for Settings." (Monster-adjusting)

**Example 4.** Argumentation may also consider the role a given abstract design plays within a given icon set.

**$P_1$:** "Even without knowing what the first or third icon in Figure 8I stands for, I can make a conjecture that it has to do with a server or a user interface accessed via the cloud. However, with the second icon, I'm not sure what it means. It is composed of various signs that I don't understand. It's probably badly designed." (Conjecture)
**$O_1$:** "Did you notice that icons in Figure 8II and Figure 8III are both defined as part of the same icon set? They mean 'Stateful component' and 'Processing component' respectively. Therefore, the second icon is actually well designed, because it uses signs appearing in other icons of the same icon set." (Counterexample)

[2]From `http://cloudcomputingpatterns.org`.

**$P_2$:** "But the second icon contains a pipe sign that is not used anywhere within the icon set, so I still don't know what the second icon means. If there were an icon with a pipe sign with a clear meaning, then I could understand the second icon better. " (Strategic withdrawal)

The main characteristic of employing Lakatosian reasoning is that it allows a dynamic and social development of the intended meaning of blended icons. This cannot be achieved by using only abstract argumentation frameworks, since they assume that the object of discussion does not evolve. Therefore, having an argumentation process of this kind has several advantages: it promotes not only open-discussions around the meaning of an icon, but also the construction of a discourse about how an intended meaning is obtained.

This is a desirable characteristic in computational creativity when evaluating creative outcomes such as concept blends. In this way, the evaluation evolves into a refinement process of an initial created concept, giving much more flexibility at the moment of deciding whether a blend is suitable.

## Discussion

We have illustrated the use of argumentation to evaluate completed blends. We alluded earlier to the role argumentation can play in the *generation* of blends, for instance by suggesting different ways to generalise the input spaces. Indeed, successive statements may serve to carry out the steps in the blending process iteratively, relaxing or refining as needed. These steps can be modelled using Lakatos's moves. From a conjectural candidate solution, to additional criteria that reveal this blend to be a 'monster' (i.e. which identify features of the candidate solution that cannot be allowed in the final solution for one reason or another), to adjustments that yield a more complete description of the problem and point the way toward a more satisfactory solution. An example of using argumentation for deciding which generalisations to use for creating a new icon is the following:

**A:** "We can create a different blend icon starting from the same icons of before." (see Figure 6)

**B:** "We could use the HARDDISK sign from the first icon and the DOCUMENT sign from the second icon."

**A:** "But putting the DOCUMENT sign above the HARDDISK does not make sense from my point of view."

**B:** "You're right, let's use the HARDDISK sign from the first icon and the PEN sign from the second icon."

**A:** "Sounds good, now we have a *Write-HardDisk* icon."

From this discussion and the previous sections, we think that it is feasible to bring the framework of argumentation inside the concept blending process. Moreover, this appears to work in a symmetric direction: the steps in an argumentation process can be carried out through blending. For instance, concept blending could be seen as the process behind the creation of rational arguments (Coulson and Pascual, 2006).

One area closely akin to the icon domain is the domain of *sentences* in a natural or artificial language. These can be evaluated for their coherence, succinctness, and fitness-to-purpose from a semantic standpoint (including relationship to other sentences), among other criteria; cf. (Abramsky and Sadrzadeh, 2014) for a category-theoretic view.

Since people have different standards for evaluation, they frequently disagree about what constitutes a satisfactory result, be it a final outcome or a design decision that is only a step the way to developing an artefact. They may also disagree at a more fundamental level about what can be considered a valid point of view or an appropriate manner of conducting an argument. For example, "Godwin's law" states that an online discussion ends when someone compares one of the discussants to Hitler and whoever made the comparison automatically loses the debate. Naturally, the validity of this principle is itself debatable. During the course of argumentation, the goalposts may shift, as new information is revealed about the domain under discussion, and about the discussants themselves. The relationship between argumentation and decision-making has been explored (Ouerdane, 2009), including the case of updating models of preferences (Ouerdane et al., 2014); the latter is quite similar to our previous work on Lakatos's games (Pease et al., 2014).

## Conclusion and Future Work

Computational models of combinatorial creativity faces the daunting issue of evaluating a large number of possible novel combinations. Particularly, Fauconnier and Turner (1998) propose a model that includes a collection of optimality principles to guide the construction of a 'well-formed integration network'. Our computational model, based on generalisations of input spaces and amalgams, makes this combinatorial nature more explicit. The heuristic criteria called 'optimality principles' are too underspecified to be used as computational measures to evaluate and select possible blends. Moreover, alternative numeric measures may be not enough to evaluate the quality or novelty of creative artefacts. Our intuition is that in the context of creative outcomes, people use argumentation to understand, criticise, modify and evaluate them, and that computational argumentation is a useful tool for computational creativity.

The domain of computer icons generated by blending, where the evaluation of new icons is focused on their intended meaning, shows that symbolic argumentation is a process that is adequate to distinguish well-formed icons from mix-and-match combinations, unambiguous and clear icons from ambiguous or incomprehensible icons. This domain supports our claim that numeric heuristic evaluation measures are insufficient to recognise good blends, and shows the usefulness of an argumentation-based process for identifying good blends, detecting their critical problems, and refining them in an evolving, open-ended process.

We have shown how Lakatosian reasoning can be used in evaluating concept blending for icon design. Our approach offers two main advantages. Firstly, the evaluation process can *improve* the blend, since the dialogue about it refines resulting blends. Secondly, the *reasons behind* a particular evaluation are made explicit. This is crucial given recent work on the importance of context in creativity judgments (Charnley, Pease, and Colton, 2012; Colton, Pease, and Charnley, 2011). Argumentation offers a framing story that

shows how and why a particular artefact was constructed, which can be presented alongside the artefact itself.

We envision several future works. First, we intend to specify an ontology for modelling the semiotic system presented and to build a library of icons. Having a domain knowledge will allow us to generate arguments by induction, for instance, by analysing icons cases. Moreover, it will also open the possibility to explore the use of value-based argumentation (Bench-Capon, Doutre, and Dunne, 2002) for selecting the input icons to be used in the concept blending process. This latter point is important, since usually the inputs of a blending process are assumed to be already provided. Second, as far as the interpretation of icons is concerned, we are thinking to take advantage of existing approaches to natural language processing and understanding, especially Construction Grammars (CxG). In CxG, the grammatical construction is a pairing of form and content. In our semiotic system, sign patterns seem equivalent to the form, while interpretations would be akin to the content. Working with a grammar would make evaluation more explicit, e.g. we could use quantitative measures of ambiguity; and this would open many other domains for application.

Finally, we plan to implement Lakatosian reasoning by employing existing computational tools for argumentation (Devereux and Reed, 2010; Wells and Reed, 2012). Our goal is to provide a computational argumentation framework and to integrate it into the framework for computational creativity we are developing in the COINVENT project.

## Acknowledgements

This work is partially supported by the COINVENT project (FET-Open grant number: 611553).

## References

Abramsky, S., and Sadrzadeh, M. 2014. Semantic unification – A sheaf theoretic approach to natural language. In *Categories and Types in Logic, Language, and Physics*, volume 8222 of *LNCS*, 1–13. Springer.

Bench-Capon, T. J. M., and Dunne, P. E. 2007. Argumentation in artificial intelligence. *Artificial Intelligence* 171(10-15):619–641.

Bench-Capon, T. J. M.; Doutre, S.; and Dunne, P. E. 2002. Value-based argumentation frameworks. In *Artificial Intelligence*, 444–453.

Boden, M. A. 2003. *The Creative Mind - Myths and Mechanisms (2nd ed.)*. Routledge.

Bou, F.; Eppe, M.; Plaza, E.; and Schorlemmer, M. 2014. D2.1: Reasoning with Amalgams. Technical report, COINVENT Project. http://www.coinvent-project.eu/fileadmin/publications/D2.1.pdf.

Chandler, D. 2004. *Semiotics: The Basics*. Routledge.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativiy. In *Proc. of the 3rd Int. Conf. on Computational Creativity*, 77–81.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *2nd Int. Conf. on Computational Creativity*.

Coulson, S., and Pascual, E. 2006. For the sake of argument: Mourning the unborn and reviving the dead through conceptual blending. *Ann. Rev. of Cognitive Linguistics* 4:153–181.

Devereux, J., and Reed, C. 2010. Strategic argumentation in rigorous persuasion dialogue. In *ArgMAS*, volume 6057 of *LNCS*. Springer Berlin Heidelberg. 94–113.

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.

Falomir, Z.; Cabedo, L. M.; Abril, L. G.; Escrig, M. T.; and Ortega, J. A. 2012. A model for the qualitative description of images based on visual and spatial features. *Computer Vision and Image Understanding* 116(6):698–714.

Fauconnier, G., and Turner, M. 1998. Principles of conceptual integration. In Koenig, J. P., ed., *Discourse and Cognition: Bridging the Gap*. Center for the Study of Language and Information. 269–283.

Lakatos, I. 1976. *Proofs and refutations: the logic of mathematical discovery*. Cambridge University Press.

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 *LNCS*, 257–271. Springer.

Ouerdane, W.; Labreuche, C.; Maudet, N.; and Parsons, S. 2014. A dialogue game for recommendation with adaptive preference models. Technical report, Ecole Centrale Paris. Cahiers de recherche 2014-02.

Ouerdane, W. 2009. *Multiple Criteria Decision Aiding: a Dialectical Perspective*. Ph.D. Dissertation, University Paris-Dauphine, Paris, France.

Pease, A.; Budzynska, K.; Lawrence, J.; and Reed, C. 2014. Lakatos Games for Mathematical Argument. In *Proc. of COMMA*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, 59–66. IOS Press.

Prakken, H. 2005. Coherence and flexibility in dialogue games for argumentation. *J. Log. and Comput.* 15(6):1009–1040.

Rahwan, I., and Simari, G. R. 2009. *Argumentation in Artificial Intelligence*. Springer Publishing Company.

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 *5th Int. Conf. on Computational Creativity*.

Walton, D., and Krabbe, E. C. W. 1995. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press.

Wells, S., and Reed, C. 2012. A domain specific language for describing diverse systems of dialogue. *Journal of Applied Logic* 10(4):309 – 329.

# Visual Information Vases: Towards a Framework for Transmedia Creative Inspiration

**Britton Horn**
PLAIT Research Group
Northeastern University
Boston, MA USA
bhorn@ccs.neu.edu

**Gillian Smith**
PLAIT Research Group
Northeastern University
Boston, MA USA
gillian@ccs.neu.edu

**Rania Masri**
College of Arts, Media and Design
Northeastern University
Boston, MA USA
rania@raniamasri.com

**Janos Stone**
College of Arts, Media and Design
Northeastern University
Boston, MA USA
j.stone@neu.edu

**Abstract**

Inspiration is an important aspect of human creativity and one that creative systems are only recently implementing. In this research, we describe and implement a transmedia creative inspiration model for generative art systems. Our implementation of this model is Visual Information Vases (VIV), an artificially intelligent ceramicist that creates 3D-printable vases using inspiration from a user-supplied image. VIV scores an image along four aesthetic measures—activity, warmth, weight, and hardness—by evaluating the image's color palette. VIV then attempts to create a vase with similar aesthetic measures through evolution. The resulting vases are diverse and functional creations. We hope that this model will allow future generative systems to create inspired artifacts from a wide variety of sources.

## Introduction

In current models of creative AI systems, one underexplored aspect of creativity is inspiration: interpreting concepts from one medium and translating them into another. The analogical mapping of perceptions and concepts (Hofstadter and Mitchell 1994) is a critical step in human creativity since it allows people access to solutions or creations outside their current mental state through influence by some external source (Hadamard 1996). This method of inspiration is common in many areas and can produce novel results. Composers interviewed by McCutchan conveyed their inspiration came through channels including music, nature, and poetry. More technical fields also involve creative inspiration, including examples of animals and insects inspiring work in robotics (McCutchan 2003).

In Thrash and Elliot's conceptualization of inspiration, three commonalities arose from their readings on previous literature: Inspiration is *evoked*, involves *transcendence* and implies *motivation* (2003). In this paper, we describe a system that focuses on evocation of inspiration from a source domain and transcendence of that inspiration to create an artifact in an entirely different domain. We also believe motivation is a critical step in creativity, however one which is outside the scope of this paper.

A full computational model of inspiration is still a long way off, however we attempt to close this gap by modeling one piece of inspiration: cross-domain analogy mapping. In order to show this is a feasible construct for creative systems, we developed a framework for transmedia analogy mapping from color palettes of images to 3D printable vases using the four aesthetic measures activity, warmth, weight and hardness. These measures were chosen primarily because of their importance among sculptors we interviewed, as well as their history within color science (Eysenck 1941; Granger 1955; Ou et al. 2004). Our inspiration framework was derived to mimic a common creative process performed by many sculptors and artists—choosing a color from an image to be the basis of inspiration for a new piece. The artist must transfer her feelings about the color onto a completely different domain. The essence of the inspiration source is not lost, but expressed in the new domain using techniques available in the target domain.

Visual Information Vases (VIV) is an AI-based generative art system which uses our model of inspiration to produce 3D-printable vases with inspiration from 2D images uploaded by a user. Users interact with VIV online by uploading images, viewing results, and printing vases for everyday use. Our proof-of-concept implementation presented in this paper produces vases through evolution using the four aesthetic measures stated above as primary components of the fitness function. To our knowledge, this is the first instance of a system using evolution to create content optimized on aesthetic characteristics from an entirely different domain.

VIV analyzes the colors of a user's image to create a color palette from salient and dominant colors. Color palette analysis is performed to create an aesthetic profile for the image. VIV then uses an evolutionary algorithm to produce a vase with a similar profile to that of the user supplied image. The resulting vase can be printed from a myriad of materials and printers to produce a functional, decorative vase. Vases are

described in a manner similar to that of Reed's while researching beauty as an aesthetic measure for evolutionary vase creation (2013).

The main contribution of this research is the implementation of a novel cross-domain inspiration framework which translates aesthetic qualities from color to vases. This framework resembles a small part of methods used by human artists to create content with external inspiration sources. While humans have successfully used this technique perhaps for centuries (Thrash and Elliot 2003), our goal is to show this is a viable form of inspiration in generative art systems through its implementation in VIV and the creation of usable, decorative vases.

## Related Work

### Generative Art Systems

Existing generative art systems use a wide range of techniques. Some create content based solely on preprogrammed rules (Cope 1996; Krzeczkowska et al. 2010; McCorduck 1990; Norton, Heath, and Ventura 2013) while others use user input (Clune and Lipson 2011; Draves 2005; Machado and Cardoso 2000; Secretan et al. 2008) or external sources (Cook and Colton 2011; Smith et al. 2006). Systems that use external inputs could be seen as receiving inspiration from outside stimuli. However, existing systems using external inspiration directly map stimuli to generation rules (Cook and Colton 2011; Smith et al. 2006). Also, these systems gain their inspiration from a pre-defined domain and so their inspiration model is non-extensible. Our model of inspiration allows an artifact from a wider range of domains to be used as inspiration for another domain since it is the high-level aesthetic measures which translate knowledge rather than a direct mapping.

A popular fitness function in generative art systems is to have either an individual or larger audience choose their favorite artifact from a set of produced artwork. The system then generates future content using responses from users. This method can be seen in *Endless Forms* (Clune and Lipson 2011) and *Pic Breeder* (Secretan et al. 2008) where users choose their favorite item from a given set of produced content. These systems create the next generation of candidates which are variants of a user's choices. On a larger scale, *Electric Sheep* (Draves 2005) produces abstract art work to please a more global audience. When a computer goes to sleep, the Electric Sheep come on to create morphing abstract animations that can be voted up by users. More popular sheep live longer and thus allow the system to evolve its creations to the favorability of a large audience. VIV was not created with the intent of personalization. Rather than have a human intervene in each generation step, VIV generates vases using aesthetic metrics found to be important by subjects of a preliminary survey.

In the domain of vase generation, one previous system has created printable vases through evolution with aesthetic measures as fitness functions (Reed 2013). Reed's generation of vases based on Birkhoff's beauty metric (Birkhoff 2003) produced many interesting vases rated highly by viewers. This research differed from previous generative art research which focused on 2D abstract art by expanding the application of aesthetic measures to a functional and decorative 3D object. Birkhoff's metric was adapted to be suitable as a fitness function in an evolutionary algorithm to great success. VIV, in contrast, does not have one aesthetic score which she is trying to maximize each time. Instead, VIV generates vases using four aesthetic measures with scores varying between evolutionary runs based on user input.

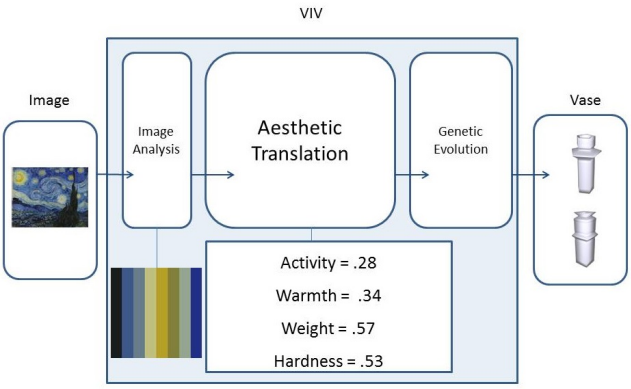



Figure 1: Design of the VIV system. The input image is analyzed and scored along the four aesthetic measures of activity, warmth, weight, and hardness. VIV's evolution component then evolves a vase to match the given aesthetic scores.

### Cross-Domain Inspiration

Cross-domain knowledge transfer as inspiration is a concept creative people have put to great use throughout history. Artists, scientists, and social leaders have gained inspiration from supernatural, internal (intrapsychic), and external (environmental) sources (Thrash and Elliot 2003). Creative computer systems, on the other hand, are only beginning to have the concept of inspiration incorporated into their makeup.

Research by Ranjan et al. (2013) had expert artists create paintings that were the artist's interpretation of one of a small set of instrumental music pieces. Results showed that people were able to correctly identify which painting went with a particular piece of music. The painters in this research did not convey which aspects of the music they were inspired by and they also did not state how they would manifest that inspiration into their painting. Similarly, viewers gave no indication of the features they found correlated between the two artistic mediums.

Similar research was conducted in the opposite direction– composers were asked to create music pieces using a square, lightning bolt, curved shape and an edgy shape as creative stimuli (Willmann 1944). This research showed composers are capable of interpreting an image, creating abstract concepts based on that image, then constructing those concepts within the domain of music. This is a complicated set of events which have yet to be implemented in computational systems. Our research attempts to close this gap by using consistent and limited aesthetic measures to demonstrate a

Figure 2: Two examples of VIV's color palette extraction and the resulting vases which correspond to a similar aesthetic profile. The left example is a warm, soft vase and the right is a cool, hard vase.

system can gather abstract characteristics from one domain and produce those concepts in a different domain with techniques unique to that domain.

One of the few generative systems that uses transmedia inspiration to create its content is *Game Blender* (Lopes and Yannakakis 2014). *Game Blender* uses conceptual blending as its means of cross-domain inspiration to create games. This crowdsourced, mixed-initiative system blends audio, narrative, ludus, and level architecture facets into a playable game. Blended creations consist of one artifact from each facet and can be controlled by the user through a number of parameters. Rather than a direct conceptual blending approach, VIV utilizes a mediation layer which performs analogy mapping from one domain to another. This is an attempt to move away from domain-specific blending approaches and towards an a more abstract methodology.

## VIV

A diagram showing an overview of VIV's process is shown in Fig. 1. This section will detail the image analysis and vase generation portions of the system.

### Image Analysis

VIV extracts a color palette of dominant and salient colors from the source image in the CIELAB color space. Dominant colors are chosen by selecting the average color from the most common bins in the image's color histogram. Colors are determined to be salient if they are at least two standard deviations from the mean color of an image (Huang, Liu, and Yu 2011). VIV then ranks salient colors according to dominance preventing tiny areas of a few pixels from making it into the color palette. Duplicates are removed using the current CIELAB distance function (Sharma, Wu, and Dalal 2005) and a final color palette is produced with a maximum of 8 dominant and salient colors each. An example color palette obtained from an image can be seen in Fig. 2.

Previous research by Ou et al. developed formulas to model single color emotion by having Chinese and English viewers rate individual colors along the aesthetic dimensions of activity, weight, warmth, and hardness (2004). We use these equations to determine scores for each color in an extracted color palette along the same four aesthetic dimensions. VIV then applies a weighted average of all colors from the dominant and salient color groups. The highest ranked colors from the dominant and salient groups are weighted at 75%. The remaining percentage is progressively halved until all colors are evaluated in the color palette. We use a weighted average rather than equal weights to allow for a more distinct aesthetic profile. We still use all colors in the color palette, although at reduced levels, since each color is a prominent color in the image and should have some effect on the overall analysis. The final aesthetic profile is then passed to an evolutionary algorithm which will use these scores in its fitness function. We acknowledge colors are a very small subset of information processed by human viewers of images and our color weighting is not necessarily human-like, however we feel this information is sufficient to demonstrate transmedia analogy mapping.

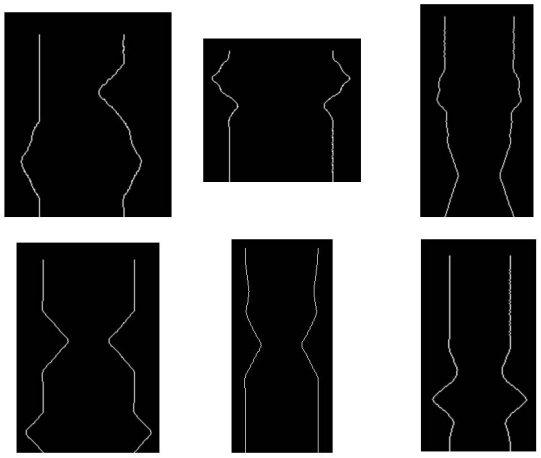


Figure 3: Example silhouettes of a variety of vases. Beziér curves for each side can be identical or unique. These curves are then interpolated around the center axis with a variable sampling rate.

### Vase Depiction

Similar to Reed's work with vases, our vases are described as two Beziér curves interpolated around a center axis. The distance from each curve to the center axis may vary and be unique between curves. Also, the interpolation can be performed with a variable sampling rate, producing vases with triangular, square, or round bases and anything between. Each vase begins as a cylinder (two straight lines of equal distance to the center axis). Vase genetic data corresponds to a set of initial parameters (e.g. starting height, width, interpolation points, number of points per line) and a list of vase manipulations.

Vase manipulations in our initial implementation are only squeeze/pull and shorten. Each of these operations can be done on one or both sides of the vase. Even with these limited and simple manipulations, definite variation can be seen (see Fig. 3). The squeeze and pull manipulations are described using two numbers: size and depth. The size determines how drastic of an alteration occurs and the depth

determines how many neighboring points are affected. This produces manipulations which can be smooth or jagged. Some constraints were placed on these alterations in order to maintain a functional and printable vase. For example, due to 3D printer constraints, a minimum wall width needed to be enforced so that the vase wouldn't break during the printing process. Vases with a minimum distance between curves below this threshold were considered non-viable and thrown out during evolution.

## Data Collection

In order to determine which vase metrics contribute to each of our four aesthetic measures, we administered a web survey to both trained artists and novices. Recruitment was done through campus e-mails, social media posts and leveraging existing professional and artist contacts. There were 50 respondents of which 27 described themselves as artists with at least three years of experience. The remaining respondents labeled themselves as "hobbyists", "no experience" or did not complete demographic information. The survey was administered anonymously therefore no background verification was done on self-reported artistic experience. All demographic information was collected at the completion of the survey. This questionnaire design was modeled after previous research which attempted to model player preference in generated Mario levels (Shaker, Yannakakis, and Togelius 2013). We applied similar techniques replacing players' level preference along fun, challenge and frustration with respondent's assessment of vase activity, weight, warmth, and hardness in order to determine features associated with each dimension.

Survey respondents were given a series of randomly generated paired vases and asked to compare them along the four previously mentioned aesthetic dimensions in a four-alternative forced choice questionnaire. Responses included "both" and "neither". An example comparison can be seen in Fig. 4. Subjects were allowed to do as many comparisons as they desired before completing the survey and filling out demographic information. The least number of comparisons performed by a single respondent was 1 and the greatest was 30 (mean=8.58). Data is still being collected, but at the time of writing this paper, 430 comparisons had been obtained. Using these 430 comparisons, vases were ranked along each aesthetic dimension by number of votes using existing pairwise comparison techniques (Shaker, Yannakakis, and Togelius 2013). A winning vote garnered one point, each vase received half a point for a vote of "both", and losing or "neither" resulted in no points awarded. Once rankings had been determined, we then used Principal Component Analysis and Multiple Linear Regression to determine which vase metrics contributed to each aesthetic measure (Freedman 2009). One big advantage of using Multiple Linear Regression is that it creates a function which is human-readable and easily implemented in a computer system.

## Feature Selection

We identified several metrics for evaluating the vases. Many more are possible but using previously applied vase metrics as a starting point, we compiled the following list:

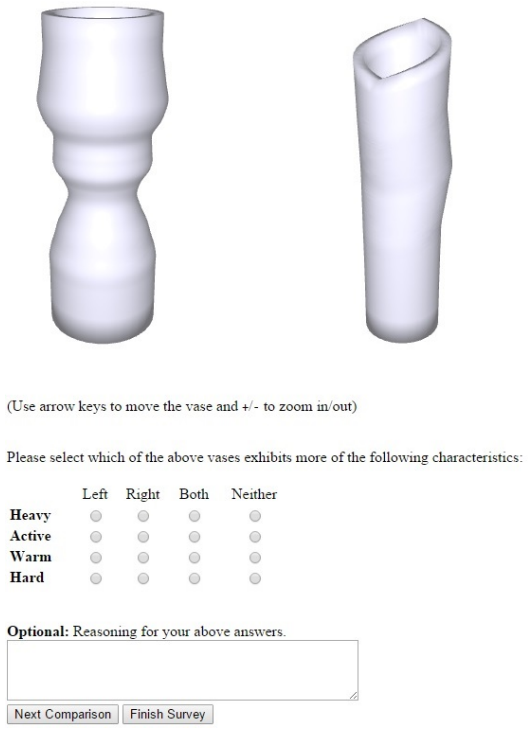



Figure 4: Example comparison from our four-alternative forced choice questionnaire.

- $H$ — Height: In vases with a height difference between sides, the greater of the two is selected.
- $W_{max}$ — Maximum width: Greatest total distance perpendicular from the center axis.
- $W_{min}$ — Minimum width: Least total distance perpendicular from the center axis.
- $I$ — Inflection points: Number of changes in slope along each side of the vase. Inflection points from each curve of the vase silhouette are added to obtain the total inflection points.
- Center of mass: $x$ and $y$ location of the center of mass of the 3D rendered vase. $CoM_x$ denotes the $x$ location and $CoM_y$ denotes the $y$ location.
- Linearity: Variance from a straight line between inflection points averaged along each side of the vase. A cylinder would have a linearity of 1.0 as the base and lip location count as inflection points and there is no variation between the two in the vertical direction.
- $S$ — Sampling Rate: number of equidistant points around the unit circle which are used during interpolation. Can also be viewed as the number of points in the base.

We also included additional relational metrics which are the result of combining these:

- $A$ — Asymmetry. Evaluated as $\frac{CoM_x}{W_{max}}$
- $R_{min}$ — Minimum width to height ratio. $\frac{W_{min}}{Height}$

Figure 5: Printed vases created by VIV. The left and center vases were created with inspiration from the artwork in fig. 2 (first example) and the vase on the right is an attempt by VIV to make her most active vase.

Figure 6: Depiction of the complete vase generation process using inspiration from one version of the famous *Scream* works by Edvard Munch.

- $R_{max}$ — Maximum width to height ratio. $\frac{W_{max}}{Height}$

Principal Component Analysis of our survey data for activity yielded important vase metrics to be the number of inflection points, lateral asymmetry and a low number of interpolation points. Warmth was influenced by lateral symmetry and a higher number of interpolation points. The ratio of the location of the minimum and maximum widths to the vase height correlated with weight. Hardness was determined by a high ratio of maximum width to height, high center of gravity, and less interpolation points. Each of the vase metrics used are not direct inputs to the vase generation algorithm. Instead, they are tools for expression of aesthetic qualities interpreted from another domain.

### Vase Generation

Each generated vase is given an aesthetic profile by the four equations below which was determined through Multiple Linear Regression of our survey data.

$$Activity = -0.2 * I + 2.3 * A - 0.002 * S + 0.5 \quad (1)$$

$$Warmth = -2.0 * A + 0.001 * S + 0.41 \quad (2)$$

$$Weight = 0.06 * R_{width} - 0.06 * R_{max} + 0.6 \quad (3)$$

$$Hardness = 0.2 * R_{max} - 1.8 * CoM_y - 0.2 * R_{min} - 0.01 * S + 1.6 \quad (4)$$

The fitness function used during evolution is the Euclidian distance between an image's aesthetic profile and the generated vase's profile where evolution is trying to minimize this score.

Vase creation is done through genetic evolution of a population of 100 vases over 100 generations. We used 100 generations because this is the point where additional generations produced results which were no closer to an aesthetic profile than the current population. For each generation, there is a 10% elitism rate where vases are kept without change, 40% crossover rate, and 50% mutation rate.

Recall that vase representation is comprised of initial parameters including starting height, width, sampling rate, and points per line as well as a list of vase manipulations. Our crossover implementation involved choosing initial parameters from one parent or the other and combining manipulation lists. Manipulations lists could be combined in a couple of different ways. Most simply, the manipulations from the second parent could be appended to the first parent's list. Alternatively, for each list index, one manipulation was randomly chosen from a parent's list at that same index.

Mutations involved re-assigning one of the initial parameters to a different value, adding a manipulation to the manipulation list at a random index, randomly removing a mutation, or altering the size of a manipulation.

## Results

The examples given in this paper show input from a variety of famous artworks (see Fig. 6) and the diverse vases created by VIV using each of these artworks as inspiration. There is great variety in input and output to the system yet VIV consistently creates vases with an aesthetic profile which reflects that of the inspiring work. Fig. 8 demonstrates a set of vases produced from the amateur photo in Fig. 7. VIV determined this image to be a light and soft image so the vases produced tended to be round with a high center of gravity.

Using a generative art system such as VIV coupled with modern 3D printing techniques, vases can be produced in a matter of hours which previously took expert artists weeks, if at all. Fig. 9 is an example which ceramicists we corresponded with stated would be extremely difficult for them to replicate because of the sharp edges throughout the internals of the vase.

## Discussion

In order to prove our inspiration model, we set out to create a working generative art system with this model at its core. VIV has been used to create numerous distinct vases with various aesthetic profiles inspired by images ranging from some of the most famous paintings to amateur photos. While many may argue VIV is not truly creative since she neither possesses any type of novelty search nor a true understanding of her creations, we can see that cross-domain analogical inspiration is a viable model for generative art systems.

Our initial implementation uses the four aesthetic measures of activity, warmth, weight, and hardness as the inspiration channels between two dimensional images and 3D-printable vases. This model is not confined to our proof-

Figure 7: An example image and color palette extracted from an amateur photo.

Figure 8: A set of vases created with the image from Fig. 7 as inspiration. The image was viewed by VIV as soft and light therefore the vases produced had a high center of gravity and a round base.

of-concept, but extensible to other analogy mappings and domains. Our implementation has shown how a system can interpret aesthetic measures from one domain using techniques specific to that domain, create an analogous mapping to another domain, and produce content within the target domain using techniques separate from those of the source. Fig. 5 contains examples of VIV's final printed output.

## Future Work

Color analysis is just one piece of information people take in when viewing art. In future implementations, a more robust image analysis which includes line, angle, feature, and object detection would be desirable as well as the extension of our single color affect analysis to color combinations. Just as human viewers take in a range of stimuli from artwork, we want VIV to mirror this in her analysis of images with a more in-depth interpretation.

We plan to conduct user studies in order to quantitatively determine if our resulting vases fit within acceptable bounds of the previously stated aesthetic measures for a large portion of human viewers rather than our initial face-value assessment. We envision this proceeding in two phases. For the first validation phase, we will give subjects a pool of vases with varying pre-defined aesthetic profiles and ask them to group together the vases they feel are most similar. If our vase profile equations are adequate, subjects should be able to organize vases by aesthetic profile. The second validation phase would extend this method to grouping vases by image. Because our inspiration model only uses an image's color palette rather than the image as a whole, this validation may be better suited to grouping by color palette rather than by original image.

Also, extension of these aesthetic measures to other researched methods would be beneficial. Birkhoff's beauty metric is an abstract aesthetic measure which could be incorporated since it perhaps is more easily studied in a broad range of domains rather than something such as warmth or hardness. As this measure has already been studied in both the domains of evolutionary vase creation and color science, its addition to our initial implementation would be rather straightforward. However, its use in domains where more granular aesthetic principles are hard to assess could be useful for future applications.

## Conclusion

We have presented the detailed design of VIV and her use of a novel cross-domain inspiration framework. We demonstrated how VIV uses this framework to create vases with an aesthetic profile interpreted from a different domain. In this way, abstract artistic concepts can be gathered from one domain and manifested in another mirroring creative methods utilized by people. Generative art systems in parallel with new media technologies, allow for a wider range of artistic content to be produced by both humans and computers. Our hope is that this model of inspiration can be used to provide creative systems with the ability to translate high level knowledge between new domains and expand their expressive range as well as broaden people's creative potential.

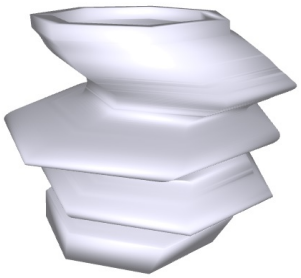


Figure 9: Example vase from VIV obtained when she tries to max out the activity measure. This vase was considered to be difficult to replicate by some ceramicists.

## References


Birkhoff, G. D. 2003. *Aesthetic Measure 1933*. Harvard University Press.

Clune, J., and Lipson, H. 2011. Evolving 3d Objects with a Generative Encoding Inspired by Developmental Biology. *SIGEVOlution* 5(4):2–12.

Cook, M., and Colton, S. 2011. Multi-faceted evolution of simple arcade games. In *CIG*, 289–296.

Cope, D. 1996. *Experiments in musical intelligence*, volume 12. AR editions Madison, WI.

Draves, S. 2005. The Electric Sheep Screen-Saver: A Case Study in Aesthetic Evolution. In Rothlauf, F.; Branke, J.; Cagnoni, S.; Corne, D. W.; Drechsler, R.; Jin, Y.; Machado, P.; Marchiori, E.; Romero, J.; Smith, G. D.; and Squillero, G., eds., *Applications of Evolutionary Computing*, number 3449 in Lecture Notes in Computer Science. Springer Berlin Heidelberg. 458–467.

Eysenck, H. J. 1941. A Critical and Experimental Study of Colour Preferences. *The American Journal of Psychology* 54(3):385–394.

Freedman, D. A. 2009. *Statistical models: theory and practice*. cambridge university press.

Granger, G. W. 1955. An Experimental Study of Colour Preferences. *The Journal of General Psychology* 52(1):3–20.

Hadamard, J. 1996. *The Mathematician's Mind: The Psychology of Invention in the Mathematical Field*. Princeton University Press.

Hofstadter, D. R., and Mitchell, M. 1994. The copycat project: A model of mental fluidity and analogy-making. *Advances in connectionist and neural computation theory* 2(31-112):29–30.

Huang, C.; Liu, Q.; and Yu, S. 2011. Regions of interest extraction from color image based on visual saliency. *The Journal of Supercomputing* 58(1):20–33.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation-with intent. In *Proceedings of the 1st international conference on computational creativity*, 20.

Lopes, P., and Yannakakis, G. N. 2014. Investigating Collaborative Creativity via Machine-Mediated Game Blending. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Machado, P., and Cardoso, A. 2000. NEvArthe assessment of an evolutionary art tool. In *Proceedings of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, Birmingham, UK*, volume 456.

McCorduck, P. 1990. *Aaron's Code: Meta-Art, Artificial Intelligence and the Work of Harold Cohen*. New York: W H Freeman & Co.

McCutchan, A. 2003. *The Muse that Sings: Composers Speak about the Creative Process*. Oxford; New York: Oxford University Press.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding Creativity in an Artificial Artist. *The Journal of Creative Behavior* 47(2):106–124.

Ou, L.-C.; Luo, M. R.; Woodcock, A.; and Wright, A. 2004. A study of colour emotion and colour preference. Part I: Colour emotions for single colours. *Color Research & Application* 29(3):232–240.

Ranjan, A.; Gabora, L.; and OConnor, B. 2013. The Cross-Domain Re-interpretation of Artistic Ideas. *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*. arXiv: 1308.4706.

Reed, K. 2013. Aesthetic Measures for Evolutionary Vase Design. In Machado, P.; McDermott, J.; and Carballal, A., eds., *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, number 7834 in Lecture Notes in Computer Science. Springer Berlin Heidelberg. 59–71.

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*, CHI '08, 1759–1768. New York, NY, USA: ACM.

Shaker, N.; Yannakakis, G.; and Togelius, J. 2013. Crowdsourcing the Aesthetics of Platform Games. *IEEE Transactions on Computational Intelligence and AI in Games* 5(3).

Sharma, G.; Wu, W.; and Dalal, E. N. 2005. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application* 30(1):21–30.

Smith, A.; Romero, M.; Pousman, Z.; and Mateas, M. 2006. Tableau machine: an alien presence in the home. New York, NY, USA: ACM.

Thrash, T. M., and Elliot, A. J. 2003. Inspiration as a psychological construct. *Journal of Personality and Social Psychology* 84(4):871.

Willmann, R. R. 1944. An experimental investigation of the creative process in music: The transposability of visual design stimuli to musical themes. *Psychological Monographs* 57(1):i–76.

# The Painting Fool Sees! New Projects with the Automated Painter

**Simon Colton**[1,2], **Jakob Halskov**[3], **Dan Ventura**[4],
**Ian Gouldstone**[2], **Michael Cook**[2] and **Blanca Pérez-Ferrer**[1]
[1]The MetaMakers Institute, Academy for Innovation and Research, Falmouth University, UK
[2]Computational Creativity Group, Department of Computing, Goldsmiths, University of London, UK
[3]UBIC Inc., Tokyo, Japan
[4]Computer Science Department, Brigham Young University, USA

**Abstract**

We report the most recent advances in The Painting Fool project, where we have integrated machine vision capabilities from the DARCI system into the automated painter, to enhance its abilities before, during and after the painting process. This has enabled new art projects, including a commission from an Artificial Intelligence company, and we report on this collaboration, which is one of the first instances in Computational Creativity research where creative software has been commissioned directly. The new projects have advanced The Painting Fool as an independent artist able to produce more diverse styles which break away from simulating natural media. The projects have also raised a philosophical question about whether software artists need to see in the same way as people, which we discuss briefly.

## Introduction

The Painting Fool (`thepaintingfool.com`) is software that we hope will be taken seriously as a creative artist in its own right, one day. It is a well established project, with an emphasis on implementing processes which could be described as artistic and/or creative, rather than merely producing images which look like they may have been painted by a person, as with many graphics packages, as per (Strothotte and Schlechtweg 2002). Many technical details of the project and discussions of the outreach activities performed with The Painting Fool are given in (Colton 2012b). Progress in the project is usually both technical and/or societal, and the work presented here addresses both aspects.

On the technical side, we have enabled The Painting Fool to use machine vision techniques before, during and after painting, to take on more creative responsibility, produce more interesting pieces and provide better framing information. This has involved integrating aspects of the machine vision abilities of the DARCI system (Norton, Heath, and Ventura 2013; Heath, Norton, and Ventura 2014). In addition to being used in art generation itself (Norton, Heath, and Ventura 2011), DARCI has been used as an artificial art critic (Norton, Heath, and Ventura 2010), which makes it the perfect complement to The Painting Fool. Implementing such synergies is rare in Computational Creativity research, with a few notable exceptions, such as the combination of parts of the MEXICA, Curveship and GRIOT programs into the Slant storytelling system (Montfort et al. 2013).

On the societal side, to get The Painting Fool accepted as an artist, we engage the public, journalists and members of the art world (artists, art students, art educators, critics, curators, gallery owners, etc.), as natural stakeholders in the question of whether software can be creative or not. Exploration of some of the stakeholders issues in Computational Creativity is given in (Colton et al. 2015), where The Painting Fool is a case study. This, along with a philosophical underpinning given in (Colton et al. 2014) provide a general grounding for the design decisions presented here, in terms of why they represent significant progress towards the long-term aim of public acceptance of The Painting Fool as a creative artist. In this context, we describe here three new art projects where The Painting Fool has used its new visual capabilities with increasing sophistication, to produce interesting art and experiences for audiences via more autonomous behaviours in the software. These projects include a mood-based portraiture demonstration, where the visual processing was used to express intent; The Painting Fool's first art commission for a third party; and a private art project.

The collaborative projects with DARCI have progressed The Painting Fool project along a number of axes. With machine vision abilities, it can now analyse its output, albeit simplistically: new functionality with potential to make it more appreciative in motivating and assessing projects, and via analysis during sketching activities. Also, choosing rendering styles can now be done by the software itself, rather than a person. This adds much autonomy, increases impressions of creative responsibility in the software, and has led to surprising results, as the paintings no longer only resemble those produced in traditional ways by people. The Painting Fool uses the digital medium more fully in interesting new styles difficult for people to achieve, which again increases the impression of independence and creative responsibility.

This paper is organised as follows. In the next section, we describe aspects of The Painting Fool and DARCI used in the collaboration, followed by a discussion of how association networks from DARCI were used by The Painting Fool in increasing levels of sophistication. We then present the three new art projects enabled by this collaboration, and put these into the context of related work. We conclude with a discussion of the advances made in The Painting Fool project, and we briefly question whether software artists need to see in the same way as people.

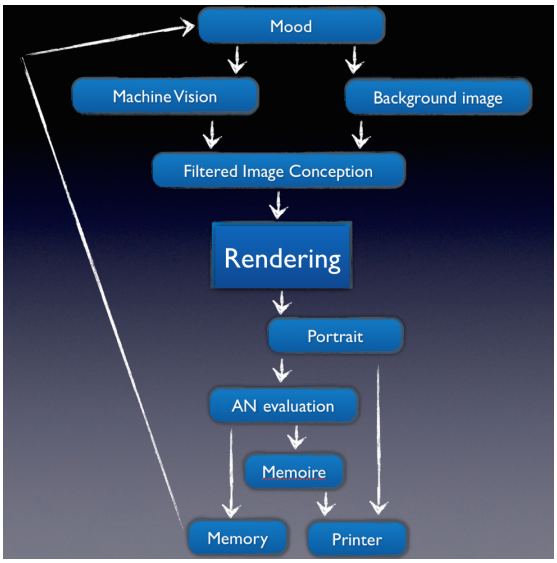



Figure 1: A workflow representation of The Painting Fool's processing for the *You Can't Know My Mind* exhibit.

## Background

### The Painting Fool: Workflows

There is no single way in which The Painting Fool produces artworks, but rather a set of tasks it can achieve through performing certain behaviours, and workflows which combine these into art-producing processes. The behaviours make use of various AI techniques including natural language processing (Krzeczkowska et al. 2010), constraint solving (Colton 2008b), evolutionary search (Colton 2008a), design grammars (Colton and Pérez-Ferrer 2012) and machine learning (Colton 2012a). The workflows are constructed through a teaching interface currently consisting of 24 screens. An example workflow, for the *You Can't Know My Mind* exhibit (described below) is given in figure 1. This highlights that the vision system is used both at the start of the process and towards the end (the 'AN evaluation' node).

Before the work described here, The Painting Fool had a very rudimentary visual analysis system that was able to evaluate features of an image such as texture, colour variance and symmetry. It is also able to segment a given digital photograph into a set of *colour regions*, using a threshold-based neighbourhood construction method, path-finding for edge rationalisation and edge abstraction methods. A waypoint in every workflow is the construction of such a set of colour regions, which can be achieved using this segmentation process, via design grammars, variation of hand-drawn scenes and/or a constraint solver placing rectangles onto the canvas. The colour regions direct the rendering process, whereby each region is either filled-in or outlined via the simulation of natural media such as paints and implements such as paintbrushes. The rendering of each region can include multiple fill/outline passes, and the rendering of the entire segmentation of colour regions can be done repeatedly, building up a layered image.

The segmentation and rendering methods are highly parameterised, requiring 14 and 57 parameters to be set respectively, as described in (Colton 2012b). Choosing from the space of possible segmentation and rendering methods constitutes a large part of the creative responsibility taken on in an art project, along with choosing and arranging subject matter, etc. We show below how the software now takes on the responsibility of choosing the rendering settings.

### DARCI: Association Networks

One way for The Painting Fool to have an increased appreciation of the artefacts it produces and some level of intentionality (both desirable qualities), is for it to employ a perceptually grounded cognitive model that can associate visual stimuli with linguistic concepts. That ability was realized by borrowing a piece of the DARCI system, a visuo-linguistic association approach, which consists of a set of neural networks that perform a mapping from low-level computer vision features to adjectival linguistic concepts, learned from a corpus of human-labeled images.

These images come from a continuously growing dataset obtained via a public facing website (`darci.cs.byu.edu`) that solicits volunteer labeling of random images. Volunteers are allowed to label images with any and all adjectives they think describe the image, and as a result, images can be described by their emotional effects, most of their aesthetic qualities, many of their possible associations and meanings, and even, to some extent, by their subject. Furthermore, through additional labeling exercises, volunteers can specify labels that explicitly *do not* describe the image, allowing the collection of explicit negative labels as well as positive ones. The result is a rich, challenging, dynamic dataset. A recent snapshot of the data reveals 17,004 positive labels and 16,125 negative labels using 2,463 unique adjectives associated with 2,562 unique images, an average of approximately 12 unique labels per image, and 110 adjectives with at least 30 positive and 30 negative image associations.

Images are perceived by the system as a vector of 102 low-level computer vision features extracted from the image using the DISCOVIR system[1]. This level of image perception does not admit significant semantic understanding, but it does allow appreciation of concepts that can be adequately expressed with global, abstract features dealing with characteristics of the image's color, lighting, texture, and shape. Given training data in the form of (image feature vector, adjectival label) pairs, a mapping is learned using a set of artificial neural networks that we call *association networks*. Since learning image-to-concept associations is a multi-label classification problem, and we cannot assume implicit negativity, the only appreciation networks trained for a particular image are those explicitly labeled with (positive or negative examples of) the associated concept. Each adjectival concept is learned by a unique association network, which is trained using standard backpropagation and outputs a single real value, between 0 and 1, indicating the degree to which an input image can be described by the network's associated adjectival concept.

[1] `appsrv.cse.cuhk.edu.hk/~miplab/discovir`

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| layers: | ABCab | ABCab | ABCab | aB | Ba | Ba | Aab | ABCab | aB | ABCab | ABCab | aB | aB | ABCab | Ba | BA | ABCab |
| blockish: | 0.272 | 0.214 | 0.233 | 0.204 | 0.226 | 0.229 | 0.200 | 0.251 | 0.244 | 0.305 | 0.237 | 0.218 | 0.222 | 0.249 | 0.225 | 0.245 | 0.233 |
| colorful: | 0.458 | 0.474 | 0.484 | 0.535 | 0.453 | 0.423 | 0.471 | 0.574 | 0.470 | 0.490 | 0.472 | 0.483 | 0.434 | 0.449 | 0.456 | 0.516 | 0.490 |
| glazed: | 0.386 | 0.298 | 0.386 | 0.315 | 0.429 | 0.380 | 0.303 | 0.362 | 0.356 | 0.358 | 0.313 | 0.327 | 0.305 | 0.373 | 0.296 | 0.325 | 0.354 |
| pale: | 0.389 | 0.459 | 0.435 | 0.498 | 0.407 | 0.451 | 0.506 | 0.428 | 0.431 | 0.390 | 0.430 | 0.495 | 0.468 | 0.431 | 0.452 | 0.416 | 0.438 |
| textured: | 0.083 | 0.179 | 0.107 | 0.226 | 0.107 | 0.138 | 0.189 | 0.125 | 0.121 | 0.110 | 0.170 | 0.161 | 0.182 | 0.101 | 0.162 | 0.105 | 0.094 |

Figure 2: Seventeen painting styles along with layering scheme and partial visual profile.

## Implementing Vision-Enhanced Painting

From the DARCI system, The Painting Fool inherited a set of 236 association networks (ANs), and a method of turning a given image $I$ into the numerical inputs to the ANs. Each AN corresponds to a particular adjective, i.e., the higher the output from the AN for adjective $A$ when given input values for $I$, the more likely (the AN predicts) that a viewer will use $A$ to describe $I$. We first determined which of the adjectival ANs were suitable for dealing with The Painting Fool's output. To do this, we ran each AN over hundreds of painterly images from The Painting Fool and recorded the range of the numerical outputs. We found that for the majority of the ANs, the output range was so low that we couldn't meaningfully claim that it was differentiating between images based on visual properties. We selected all ANs where the range of outputs was 0.05 or greater, and then performed a sanity check on those remaining, removing any which described images in a particularly counter-intuitive way, e.g., the AN for 'red' outputting a higher score for a patently green image than for a patently red image.

This left a selection of 65 usable ANs, to which we implemented an interface in The Painting Fool. For each selected AN, we recorded the highest and lowest outputs over the hundreds of images mentioned above, and when output from a new image is calculated, it is normalised between these extremes. As described below, the ANs have been used in a number of new workflow behaviours for The Painting Fool. The simplest of these is to allow the software to frame it's output (Charnley, Pease, and Colton 2012) by describing it visually. It can also compare and contrast images in terms of a particular adjective, or in terms of a profile of multiple adjectives. It can also employ the ANs during the painting process, as described in the following subsections.

### A Space of Simulated Visual Art Implements/Styles

Given an image segmentation of colour regions as described above, The Painting Fool produces a non-photorealistic rendering of it in a series of whole-segmentation layers, during which each region itself is rendered in multiple layers. During the rendering of each layer, which can either be filled in, or outlined, the software simulates natural media such as paints, and the usage of implements such as brushes in outline/fill styles such as hatching, as described in (Colton 2012b). The rendering of a layer is determined by a set of 57 parameters, which cover the simulation of the media itself (e.g., wetness of paint), the implement (e.g., brush size) the support (e.g., canvas roughness) and the style (e.g., number of times to draw an outline).

We defined a space of painting styles by fixing the rendering to a single whole-segmentation layer during which a scheme of up to five rendering layers per region was allowed. The region layering scheme was represented as a string with letters $A, B, C, a, b$ or $c$. Upper case letters represent a fill layer with lower case letters representing outline layers. Where upper and lower case letters correspond (e.g., $A = a$), all the other settings are the same, hence they represent the simulation of the same natural media in roughly the same way, but one produces an outline, the other produces a fill. For instance, $ABCab$ represents three fill layers and two outline layers, with all the settings of the first two fill layers exactly as for the two outline layers. We found that it increased visual coherence if the fill layers corresponded to the outline layers in this way. After initial experimentation, we constrained the space to include five layering schemes: $aB, Ba, ABab, Aab$ and $ABCab$, which we found produced a suitably large variety of visual styles.

We generated 1,200 painting styles by randomly sampling the space of rendering styles with each of the 57 parameters set randomly to an appropriate value in its range, and then mapping a set of these styles onto one of the five layering styles above, also chosen randomly. For each style, we used The Painting Fool to render a given segmentation of an abstract flower. Then, for each of the 65 selected ANs described above, we calculated the normalised output for each of the 1,200 flower paintings, thus creating a visual profile for each style. Example painting styles, along with the layering scheme and part of their visual profile are given in figure 2. The seventeen pictures demonstrate somewhat the diversity in the painting styles within this space. The partial profiles indicate that while the AN outputs have a relatively small range, it is sufficient for a choice of painting style based on these values to be meaningful.

### Employing Vision During Painting

To recap, we supplied The Painting Fool with 1,200 different painting styles, each with a visual profile derived from applying association networks. As described above, there are various workflows for producing images with The Painting Fool. When the workflow starts with a digital photograph, images are segmented into a certain number of colour regions, with more regions usually leading to more photorealism in the final paintings. Each colour region corresponds, therefore, to a region of the original photograph, and this photo-region can be interrogated in order to choose an appropriate painting style. To do this, The Painting Fool extracts the photo region onto a transparent image, then applies

all 65 adjectival ANs to the extract, to compile a profile. The Euclidean distance of this photo-extract profile from the visual profile of the 1,200 painting styles is used to order the styles in increasing distance. The distance can be interpreted as an *appropriateness* of the painting style to the underlying photo extract. That is, the style with least distance will render the region in a way that is most similar in nature to the original photograph (according to the ANs).

The new workflow for The Painting Fool uses machine vision during painting as follows: it takes a photograph and segments it into colour regions. For each colour region, a photo-extract profile is produced using the ANs, and this is used to order the painting styles in The Painting Fool's database, in terms of how appropriate they are to the photo extract. From the top ten most appropriate styles, one is chosen randomly to paint the region in question. Choosing from the top ten in this fashion means that each time the same photograph is painted, it produces a different image, yet each time, each painting style is appropriate to the region it is used to paint. We have enhanced this workflow by enabling a sketching mechanism. That is, The Painting Fool tries all of the ten most appropriate sketching styles in situ, then produces a visual profile of the resulting region of the painting, and chooses the one where this profile is closest to the photo-region profile. This reduces the reliance on the initial flower experiments somewhat, as The Painting Fool can see what each style looks like actually in the painting, before committing to one in particular. It also opens up the potential for The Painting Fool to produce a sketchbook to accompany each painting, as framing information.

## Cultural Applications

In the subsections below, we describe new cultural application projects with The Painting Fool which have been enabled by its access to a vision system. These span the kinds of public, private and commissioned art projects that an artist might expect to undertake as part of their general activities.

### The 'You Can't Know My Mind' Exhibition

For the *You Can't Know My Mind* exhibit reported in (Colton and Ventura 2014), we focused on the question of intentionality in creative software. As software is programmed directly, it is fair criticism to highlight that in most Computational Creativity projects, the intention for the production of artefacts comes from the software's author and/or user. For the *You Can't Know My Mind* project, we raised our intentions to the meta-level, i.e., we intended for the software to produce portraits and entertain sitters in order to learn about its own painting styles. However, the aim of each artefact production session was determined by The Painting Fool itself, in order for it to exhibit behaviours that unbiased observers might project the word 'intentionality' onto.

An 8-point description of how The Painting Fool operated in this project is given in (Colton and Ventura 2014). Of note here, we used the machine vision system from DARCI offline, to prepare the software for portraiture sessions. That is, for each of 1,000 abstract art images produced by the Elvira sub-module (Colton, Cook, and Raad 2011), and for each of 1,000 image filters produced by the Filter Feast sub-module (Torres, Colton, and Rueger 2008), the output of all of the adjective ANs were calculated. Hence the software could choose from the most appropriate abstract backdrops and the most appropriate filters for an adjective, $A$, chosen to fit a mood, to produce a *sketch conception* to aim for with each portrait. The 'background image' and 'filtered image conception' nodes in figure 1 correspond to these.


Figure 3: Example comparisons of sketch conceptions (left) and associated portraits (right). The percentages portray the range of the output from the adjective AN for the second image in terms of the AN output value for the first image.

Under the assumption that the sketch will invoke people to project certain adjectives onto the image upon viewing, the sketch conception has aspects which The Painting Fool aspires to achieve in its painting. The conception image is segmented into colour regions, and a simulation of various painting media (paints, pastels and pencils) are used in one of eight styles, to produce a portrait. At the end of each portraiture session, The Painting Fool uses the vision system to compare the level of adjective projection in the portrait to that of the sketch. To do this (indicated by the 'AN evaluation' node in figure 1), it applies the adjectival AN for $A$ to the sketch conception and to the final portrait, and compares the output. If the AN output for the portrait, $O_p$, is within 95% to 105% of the AN output for the conception, $O_c$, i.e., $0.95 \times O_c \leq O_p \leq 1.05 \times O_c$, this is recorded as satisfactory. If it is higher than 105%, this is recorded as a success, and if it is higher than 110%, this is recorded as a great achievement, with failures similarly recorded. Three examples comparisons of conception and portrait are given in figure 3. The level of achievement/failure is used to update a probability distribution that The Painting Fool can use to choose painting styles later to (attempt to) achieve an image with maximal output respect to a given adjectival AN.

Figure 4: Front page and excerpt from the Japanese version of the essay for the 'I Can See Unclearly Now' commission. Third image: an early photograph of artwork hung in the Behaviour Informatics Laboratories of UBIC.

## The 'I Can See Unclearly Now' Commission

UBIC[2] is a behavioural information data analysis company based in Tokyo. In early August of 2014, UBIC's CTO, Mr. Hideki Takeda came across The Painting Fool's website while exploring recent advances in Artificial Intelligence research on the web. At that time, UBIC's Behavior Informatics Laboratories (BIL) in Shinagawa, Tokyo, was implementing a complete office renovation scheme reflecting the company's reorientation from eDiscovery vendor to supplier of in-house Big Data Analytics solutions powered by an AI engine called the Virtual Data Scientist. The new office concept of the BIL can be summed up as: "Shaking the boundaries between the virtual and the real so as to stimulate the senses and promote intelligence and creativity". For example, the new office features both real bamboo and bamboo imprinted on a glass wall. The choice of bamboo is not arbitrary, but motivated by the fact that this plant plays a prominent role in traditional Japanese culture. It is highly symbolic and associated with, for example, Noh theatre[3] in which the protagonists are often ghosts from another plane of existence but appearing in the real world.

Mr. Takeda decided to commission artworks from The Painting Fool, as this would fit very well with the blurring of virtual and real spaces in the BIL The first author of this paper – who is the lead researcher in The Painting Fool project – was contacted by the second author acting on behalf of UBIC, and ultimately three series of images were commissioned, along with an essay highlighting how the machine vision system was used in increasingly sophisticated ways from the first to the third series. Constraints were put on the commission: (i) to include a portrait from a live sitting, and (ii) to include a piece involving Alan Turing, as an AI pioneer. Moreover, it was agreed that the commission would involve an element of research and implementation, driving The Painting Fool project forward. Example images (with details) from the three series are given in figure 6, and details from the essay, along with an early photograph of one of the pieces hung in the BIL are given in figure 4. The title of the commission was chosen to highlight The Painting Fool's new usage of machine vision techniques, while indicating that the system is far from perfect.

To tie the three series of images together, the same style of backdrop was used, consisting of 10,000 adjectives rendered in a handwritten way in varying shades of greyscale pencil, onto dark backgrounds. In all the pieces, the mass of adjectives open up in multiple places into which red handwritten adjectives are strategically placed. For the first series, *StarFlowers*, paintings of the abstract flowers used for assessing painting styles were placed using a constraint solver to avoid overlap, as per (Colton 2008b), with slightly differing sizes. Before placement, each flower image was assessed by the 65 adjective ANs, and from the top ten highest scoring adjectives, two were chosen to appear alongside the flower in the piece, in red handwriting. The pairs were automatically chosen so that no flower had the same two adjectives next to it. For instance, in the detail of figure 6, the first flower is annotated with 'peaceful' and 'warm'.

In the second series, *Good Day, Bad Day*, two photographs of the second author seated, posing firstly in a good mood, and secondly in a bad mood were used. The 65 adjectives were split into positive, neutral and negative valence categories, e.g., happy, glazed, bleary respectively. The painting style with the highest average AN output over the positive adjectives was chosen to paint the first pose, and the most negative style was similarly chosen to paint the second pose. Each portrait was annotated at its edges with red handwritten adjectives appropriate to the painting at that edge point. In the third series, *Dynamic Portraits: Alan Turing*, a photograph of Turing was hand annotated with lines to pick out his features. We then used the method of arbitrarily choosing from the top ten most appropriate painting styles for each colour region described above, to produce a number of portraits, with the annotated lines being painted on at the end, to gain a likeness. The rendered painting was analysed with the 65 ANs and the 17 most appropriate adjectives were scattered around the backdrop of the image, in a non-overlapping way, as usual in red handwriting.

Dozens of images from the three series were sent to UBIC to choose from for the BIL, with very little curation from the first author. UBIC representatives confirmed that the commission achieved the brief of producing pieces which blur the line between the real (i.e., painted by a person) and the virtual (i.e., painted by a computer), and were very happy with the commission. They produced a translated version of the essay for visitors to the lab, and hung an example from each series in the BIL.

[2] www.ubicna.com

[3] en.wikipedia.org/wiki/Noh

Figure 5: Portrait of Geraint Wiggins.

### The Portrait of Geraint Wiggins

In (rather belated) celebration of a milestone birthday, we used the vision-based sketching approach described above, to produce a portrait. Given an original image, hand-annotated with lines picking out facial features, The Painting Fool segmented it into 150 colour regions/lines, and for each, chose the top ten most appropriate painting styles, as described above. For each of the ten, it painted the region, calculated the visual profile of the region of the painting that resulted, and finally chose the style with minimal distance between its visual profile and that of the original photo-extract. In this way, the painting process was deterministic, but not predictable, and produced a striking portrait with painterly and distinctly non-painterly effects. To add a physical uniqueness, the image was printed onto 300 4cm by 4cm squares which were composed into the final piece in an overlapping formation, as per the Dancing Salesman Problem piece described in (Colton and Pérez-Ferrer 2012). The portrait is shown in figure 5.

## Related Work

It is commonplace for an artist to be commissioned to work with a bespoke piece of software, or even to develop new code, to produce artwork, with the person using the software as a tool, and this tool may be generative. However, it is much less common for a commission to be made specifically because the software will take on some of the creative, not merely generative, responsibilities.

The ANGELINA system (Cook and Colton 2014) has been commissioned to produce games for the New Scientist, Wired and PC Gamer Magazines. In the former, ANGELINA designed a game as normal, but its designer provided custom visual theming, drawing new sprites and creating sound effects for *Space Station Invaders*, since ANGELINA was not capable of this. The commissions for Wired and PC Gamer came much later, when ANGELINA had more independence and could produce full games, given just an initial theme of a short phrase, proposed by the journalist. For the PC Gamer game, *NBA Mesquite Volume 2*, ANGELINA used a database of labelled textures compiled from social media mining, for the first time in a released game. This happened because the theme chosen, 'avocado', matched a label in the database for the first time since the database had been added. This created an additional talking point for the article, and in general the games were well received and drove up online viewing figures.

The *Paul* drawing robot by Patrick Tresset (Tresset and Fol Leymarie 2012) has much in common with The Painting Fool, in that it uses a camera and machine vision techniques to capture an image, then automatically draws a portrait: in this case, physically, using a robotic arm and a pen. It also simulates looking while it draws, but this is only for entertainment purposes, i.e., after the initial photograph is taken, the vision system is not used again. Paul has been commissioned on a number of occasions, most notably for a week-long workshop at the Centre Pompidou in late 2013. Tresset has also found success in selling versions of the robot painter to art museums. Another robotic painter, which does use machine vision during painting and has also been commissioned for art is the eDavid system, as described by (Lindemeier, Pirk, and Deussen 2013). Here, a camera is used to photograph the canvas after a series of paint strokes have been applied, with a vision system employed to optimise the placement of future strokes based on the visual feedback.

It is beyond the scope of this paper to perform a survey of commissions where software creators rather than artists controlling software have produced artworks. However, we can tentatively introduce some metrics for comparing projects/software/programmers to begin to characterise such commissions. For instance, one could compare the domain specific training of the programmer, e.g., comparing the commissions of artist Harold Cohen (who represented the UK in the Venice Biennale) and his AARON system (McCorduck 1991) with Oliver Deussen (who has no artistic training) and his eDavid system mentioned above, as this may indicate more autonomy in the software (but doesn't necessarily). Other measures could include how much curation takes place, i.e., how much of the software's output is usable; what amount of hand-finishing of output takes place; and how much extra coding is required for each project.

## Conclusions and Future Work

Through the above projects, The Painting Fool has advanced as an artist in three major ways. Firstly, the creative responsibility of choosing a painting style has been handed to the software. With the *You Can't Know My Mind* project, it learned a probability distribution which can choose between one of eight painterly rendering styles, to produce an image which people will probably describe using an adjective, chosen intentionally to express a mood. With the *I Can See Unclearly Now* project, the software gained the ability to choose between 1,200 painting styles for each colour region dynamically during painting. With the *Portrait of Geraint Wiggins* project, it went further: performing in situ sketches

to test painting styles in the context of the painting at hand. Hence, the decision making involved in determining rendering styles is now undertaken by the software, which is a major advance in autonomy, and potentially towards its acceptance as an artist in its own right.

Secondly, on close inspection of the pieces in figures 5 and 6, while the images produced retain a painterly style somewhat, there are aspects which couldn't be produced with natural media simulation. This is because the painting styles in its database include ones which simulate the ground in-between natural media such as paints and pastels, and others which have no analogue in the physical world. This means that – for the first time – The Painting Fool can produce images using a much broader range of pixel manipulations, producing styles which have little grounding in traditional painting, We also see this as a major advance, as it extends the variety of images the software can produce, and potentially increases perceptions of autonomy.

The third advance will be expressed more in future work than in the projects presented here. Through the mapping of visual stimuli to linguistic concepts, The Painting Fool is able to project adjectives onto images, and we plan to enhance this with the ability to similarly project nouns. This will increase its capacity to appreciate its own work and that of others, enabling it to provide more sophisticated commentaries about what it has produced, and we touched on this with the output in the *You Can't Know My Mind* project, where the conceived and rendered images are compared visually. We plan to take this framing further, with The Painting Fool keeping a sketchbook for each project, adding value, and helping audiences to understand its processes.

It is clear from figure 2 that the visio-linguistic system does not yet match that of people perfectly. Moreover, we acknowledge that – as pointed out by a reviewer – we have not provided data to verify that our strategy to match the visual profile of an image with appropriate painting styles for regions is a good strategy, nor have we yet compared and contrasted alternatives or tested people's reactions to the artworks produced. We aim to experiment with this approach and explore alternatives in future work. However, before undertaking much further work, we wish to raise, discuss and be guided by responses to a philosophical question for the Computational Creativity community: is it important that an automated artist has a visual system similar to that of people? For communication/framing value, it might be preferable for the software's visual judgements to match ours closely. However, as illustrated by a recent internet storm about colours in a dress (Rogers 2015), we all have different visual perception systems, and notions of beauty differ from generation to generation and person to person. As art is driven forward by such differences, it may be more interesting and important artistically for us to learn The Painting Fool's visual system, rather than it learning ours.

## Acknowledgements

We wish to thank the anonymous reviewers for their very helpful and insightful comments. This work was funded by EPSRC project EP/J004049, EC funding from the FP7 ERA Chair scheme (project number 621403) and by UBIC Inc.

## References

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. *Proc. 3rd ICCC*.

Colton, S., and Pérez-Ferrer, B. 2012. No photos harmed/growing paths from seed – an exhibition. In *Proceedings of Non-Photorealistic Animation and Rendering*.

Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On Acid Drops and Teardrops: Observer Issues in Computational Creativity. In *Proceedings of the AISB symposium on AI and Philosophy.*

Colton, S., and Ventura, D. 2014. You Can't Know My Mind: A festival of computational creativity. *Proc. 5th ICCC*.

Colton, S.; Pease, A.; Corneli, J.; Cook, M.; Hepworth, R.; and Ventura, D. 2015. Stakeholder groups in computational creativity research and practice. In Besold, T.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity Research: Towards Creative Machines*. Springer.

Colton, S.; Cook, M.; and Raad, A. 2011. Ludic considerations of tablet-based evo-art. In *Proceedings of EvoMusArt*.

Colton, S. 2008a. Automatic invention of fitness functions with application to scene generation. In *Proceedings of EvoMusArt*.

Colton, S. 2008b. Experiments in constraint based automated scene generation. In *Proceedings of the fifth international workshop on Computational Creativity*.

Colton, S. 2012a. Evolving a library of artistic scene descriptors. In *Proceedings of EvoMusArt*.

Colton, S. 2012b. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*, 3–38. Springer.

Cook, M., and Colton, S. 2014. Ludus ex machina: Building a 3D game designer that competes alongside humans. *Proc. 5th ICCC*.

Heath, D.; Norton, D.; and Ventura, D. 2014. Conveying semantics through visual metaphor. *ACM Transactions on Intelligent Systems and Technology* 5:31.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation – with intent. *Proc. 1st ICCC*.

Lindemeier, T.; Pirk, S.; and Deussen, O. 2013. Image stylization with a painting machine using semantic hints. *Computers and Graphics* 37(5):293–301.

McCorduck, P. 1991. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. W. H. Freeman & Co.

Montfort, N.; Pérez y Pérez, R.; Harrell, F.; and Campana, A. 2013. Slant: A blackboard system to generate plot, figuration, and narrative discourse aspects of stories. *Proc. 4th ICCC*.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. *Proc. 1st ICCC*.

Norton, D.; Heath, D.; and Ventura, D. 2011. Autonomously creating quality images. *Proc. 2nd ICCC*.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2).

Rogers, A. 2015. The science of why no one agrees on the colour of this dress. *Wired, Science Section, 26th Feb*.

Strothotte, T., and Schlechtweg, S. 2002. *Non-Photorealistic Computer Graphics*. Morgan Kaufmann.

Torres, P.; Colton, S.; and Rueger, S. 2008. Experiments in example based image filter retrieval. In *Proceedings of the Workshop on Cross-Media Information Analysis, Extraction and Management*.

Tresset, P., and Fol Leymarie, F. 2012. Sketches by Paul the robot. In *Proceedings of the 8th Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*.

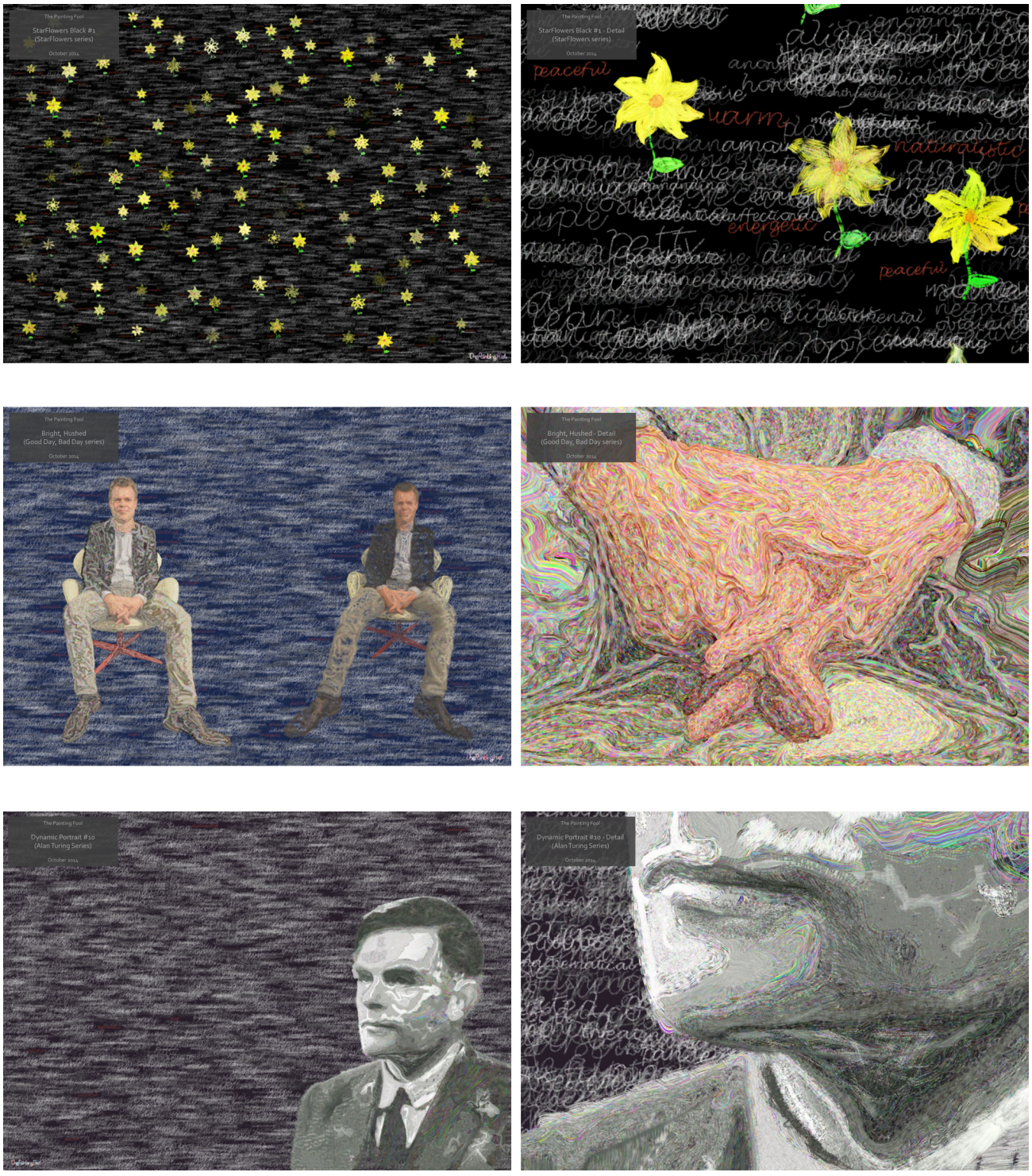



Figure 6: Example images, each with detail, from the 'I Can See Unclearly Now' commission. First pair: from the *Star Flowers* series. Second pair: from the *Good Day, Bad Day* series. Third pair: from the *Dynamic Portraits: Alan Turing* series.

# Make Something That Makes Something: A Report On The First Procedural Generation Jam

**Michael Cook**
Computational Creativity Group
Goldsmiths, University of London
ccg.doc.gold.ac.uk

### Abstract

We report on the first procedural generation jam, PROCJAM, an event designed to bring together artists, researchers and game developers to experiment with new techniques and applications for generating content for videogames. Much of the event's resulting work has applications beyond videogames, however, and we believe the event may be a strong platform for engaging creators and programmers in Computational Creativity in the future. We discuss the structure of the event, the results it yielded, and the potential future impact of such events on the Computational Creativity community.

## Introduction

Procedural content generation (PCG) is a crucial and rapidly developing area of videogames technology (Togelius et al. 2011). PCG is a rich area of games culture – it has been used as a supplement for human effort (Interactive Data Visualization 2002), a source of wonder and unpredictability (Toy et al. 1980), a tool for artistic expression (Betts 2014), and a unique mechanical design tool (Yu 2009). Its increase in popularity and its growing importance in the culture of videogames has also been mirrored by a surge in the aesthetic of generative software in art, web culture (such as Twitterbots) and other creative media. These are all areas which have considerable overlap with Computational Creativity in terms of the techniques they use to generate artefacts, and represent a great opportunity to share the field's philosophy and theory with a vibrant, active community of people.

Game jams are increasingly common events within the game development community where people develop games under the constraints of both a time limit and some kind of common theme (which might be a technical constraint such as containing the game within 4 kilobytes of Java[1] or a creative constraint such as incorporating a theme like *fishing*[2]). Entrants to game jams typically fall within one of two categories: novices looking to use the event to create their first game, and experienced developers looking to experiment and innovate (Zook and Riedl 2013). In both cases the short timescale helps encourage entrants to set themselves projects which are small enough to be easily completed.

Popular game jam formats are repeated at regular intervals throughout the year. Ludum Dare,[3] one of the most popular, runs every four months. Entrants make a game in 48 hours from scratch, including game design, code, music and visual art, following a theme voted on by entrants in the week prior to the game jam. In December 2014, Ludum Dare 31 received over 2000 entries for the theme *Entire Game On One Screen*. By running repeatedly, these regular game jams build communities of creators who meet to create together, share ideas, give feedback on games (there is an extensive period of reviews and ratings after the jam) and often form collaborations or extend their jam entries into full commercial releases (Zucconi 2014). They form strong global communities who share ideas, draw in new practitioners, and push forward the state of the art (Gray et al. 2005).

In this paper we present a report on the first procedural generation jam, or PROCJAM, an event held in November 2014. The jam ran over nine days, starting with a streamed day of talks about procedural generation and ending with 138 entries being submitted in the form of games, tools, experimental prototypes and artworks. Although styled as a game jam, PROCJAM deviated from the traditional format in several important ways, which helped expand the appeal of the event beyond game developers and draw in people interested in generative techniques in general. We will go into these changes to the format in depth later in this paper, as we believe they are crucial to the success of PROCJAM and point to a format for generative events that could form the basis of Computational Creativity outreach in the future. We also will outline how PROCJAM itself is fostering work related to Computational Creativity and how this can grow in coming years.

We believe that the community-building and experimental aspects of game jams are extremely valuable, and make the game jam format ideal for engaging communities of programmers such as Twitterbot writers, programmer-artists and game developers with Computational Creativity, as well as being rich sources of inspiration and code which would be of benefit to everyone working in and around this field. Additionally, events like PROCJAM can also be valuable ways

[1] http://www.java4k.com
[2] http://www.fishingjam.com
[3] http://www.ludumdare.com/compo

to expose people to Computational Creativity for the first time, in much the same way that game jams encourage people to try out making a game, and may serve as a useful model for student workshops and similar activities.

In this paper, we will outline the format for PROCJAM and explain how and why we deviated from several common elements of typical game jam organisation to create a better community for creating and sharing ideas. We then give several specific examples of entries to the jam and discuss their relevance to Computational Creativity. We follow this with a general analysis of entries, identifying issues related to Computational Creativity that arose in them, and also areas where our research could contribute to future entries to the event. Finally, we discuss the jam format as a model for outreach and engagement, and look ahead to the future of PROCJAM.

## PROCJAM Format and Organisation

PROCJAM took place from November 8th to the 17th 2014, co-ordinated across the web using Twitter hashtags and a central website where people could submit their entries.[4] Subsequently, the jam has registered its own website to coordinate the community and future events.[5]

The most common format for a game jam is as follows: at the beginning of the jam a theme is announced, normally on the jam's website so that people can take part from around the world. Participants then have 48 hours to develop a game from scratch, including art and music assets, that somehow incorporates the jam's theme. Entries are then submitted at the end of 48 hours. A review phase then takes place in which people vote for their favourite entries, with the voting pool consisting either of the general public, other entrants to the jam, or a select panel of judges. Prizes may be awarded to the winners.

This format for a game jam is very popular and is replicated hundreds of times a year from large-scale jams with hundreds of entrants down to small-scale local jams run between small groups of friends. With PROCJAM we made several changes to the standard game jam format with the express aim of increasing participation, particularly with those who had relevant experience writing generative software but had not interacted with game developers before. A secondary aim for the jam's format was to encourage experimentation and allow people to prototype unusual projects that stretched the state of the art in generative techniques for games.

Unlike most other game jams, making a game was not the only way to enter PROCJAM. Entrants could alternatively submit a piece of software that simply generated something (the jam's slogan was *Make Something That Makes Something*). Developing a game is a highly specific skill that people are unlikely to have unless they already work in games, and developing a game in the timeframe of a game jam is even more difficult. By relaxing this constraint, people who have interesting ideas, knowledge or skills can contribute generative systems to the jam that might spur on projects or inspire developers to integrate new kinds of system in their future games. As a result, the jam received many entries in the form of complete games, but equally saw systems which generated dungeons and planets, weapon ideas and fabric designs, music loops and more. Bringing in people from different backgrounds helped make PROCJAM feel more like a melting pot of ideas and less like a competition.

We removed the requirement to produce original artwork and music for the jam, too. Since the primary focus of the jam was on new ideas in procedural generation, rather than testing game development skills, it didn't make sense to require people to put effort into elements of a game that were unrelated to their main contribution. This encourages people to enter the jam by relieving pressures on them to take on more work. We also removed a similar requirement that all code should be written from scratch. Game jam games tend to be very simplistic in nature because of their short development cycles, which works well for the goals the jams often have. However, in order to allow people to spend the week focusing on procedural generation, it made sense to allow them to use existing codebases or even entire games. One group of developers took a game they had been working on and added a procedural generation system to it as part of the jam. This doesn't just make the jam more appealing to outsiders, it can also allow deeper work to be done that builds on existing efforts (Hecker 2012).

By allowing entrants to make anything from a small script to a full game, and removing the restrictions on existing code and art assets, the process of evaluation becomes an issue. This raises the question of how to rate and compare entries when they are so varied in their origins. PROCJAM circumvents this simply by removing the ratings process - people are encouraged to comment on each others' entries and share them among one another, but there is no numerical rating system and no winners are declared. This solves the issue of comparing, say, a script which generates quilt patterns with a full murder mystery game. However it simultaneously encourages people to try out more experimental ideas without the intimidation of being judged and ranked by someone else's idea of what a good jam entry should be.

All of these changes have the same ultimate aims: to encourage people to take part, particularly those who are not game developers by trade, and to encourage experimentation and the sharing of new ideas.

We supplemented PROCJAM with a day of talks which we livestreamed on the web on the first day of the jam[6]. 80 people turned up to attend the day of talks, with 200 viewers tuning in to each talk throughout the day, and many hundreds more have viewed the recordings of the talks online since. The talks provided inspiration to jam entrants, with many citing the talks during the development of their jam entry, but they also provided an opportunity to be exposed to new views on generative systems – the speakers included an academic researcher, an artist and a creative director at an indie games studio. One of the aims of the event was to elevate procedural generation in games beyond "random levels", and having a variety of speakers giving talks was a

[4] http://itch.io/jam/procjam

[5] http://www.procjam.com

[6] http://www.procjam.com/talks/2014


Figure 1: A screenshot from Dreamer of Electric Sheep.


Figure 2: A screenshot from Inquisitor showing the conclusion to a case. The player has correctly guessed the motive but not the murderer or the weapon.

good way of doing this. We hope to have a speaker at next year's PROCJAM event to promote Computational Creativity as a new philosophy for procedural generation in games.

## Selected Entries

In this section we will briefly describe and discuss three entries to the jam. We look at their most interesting features and the responses some of them received from the games community. We selected three projects that we believed would be of most interest to the Computational Creativity community, either because of their philosophy, innovative concepts, or the relationship between the techniques used and ideas within the field of Computational Creativity. We give details of where to find these entries, as well as all other jam entries, in the next section.

### Dreamer of Electric Sheep

Dreamer of Electric Sheep is a text adventure submitted to PROCJAM by Tom Coxon, who also gave a talk at the jam's opening event about his procedurally-generated adventure game, *Lenna's Inception* (Bytten Games 2014). Like most text adventures, players are presented with descriptions of their surroundings and can manipulate the world by inputting simple commands for their character to execute. Dreamer attempts to procedurally generate the game world using a combination of ConceptNet (Liu and Singh 2004), a commonsense knowledge database, and Spritely[7], a tool which generates game art from web searches. ConceptNet stores its data in the form of concepts, series of facts that are all about a similar topic. These facts are linked to one another through triplets (such as {*magazine, AtLocation, bookstore*}) which can be explored through an API which Dreamer uses. ConceptNet has seen use in academic Computational Creativity research, such as (Llano et al. 2014).

Dreamer searches the ConceptNet API for concepts which other concepts are linked to via the relationship *AtLocation*. It then populates those places with objects and characters that ConceptNet says should be found in them, and uses Spritely to generate an illustration of the location. Spritely queries online image databases such as Google Images and Wikimedia Commons, searching for images that can be cleanly shrunk down in size with their backgrounds extracted, to make relatively clear sprites for use in games.

The player can perform common text adventure commands such as moving in the cardinal compass directions to travel between places, as well as picking up objects. However, because the game lacks deeper knowledge about the objects it places in each location, this can result in surreal interactions like picking up shop assistants and taking them with you. The game gets around this somewhat by being set inside a dream world, thereby allowing unusual things to take place without the game's sense of reality breaking.

### The Inquisitor

The Inquisitor is a murder mystery game by Malcolm Brown. The game tasks the player with solving a murder by investigating the crime scene, discovering evidence, questioning witnesses and identifying the murderer, murder weapon and motive. The crime is procedurally generated, generating a cast of characters and relationships between them, simulating the movement of the characters before and after the murder (so that evidence such as blood trails and witnesses are realistic and consistent) and then leaving the player to put together the details within a time limit.

Although the individual generative techniques within The Inquisitor are not new *per se*, the way it uses them to generate murder mysteries is novel and quite effective. In particular, interviewing witnesses yields partial and sometimes conflicting information, forcing the player to take notes and draw up potential scenarios in which certain characters are lying, and information is procedurally redacted from certain kinds of evidence, leaving out the contents of a letter but revealing its author, for example, or smudging the name of its author but revealing unrequited love. The Inquisitor also

[7] http://www.github.com/gamesbyangelina/spritely

Figure 3: A screenshot from Secret Habitat showing a generated gallery in the generative game landscape. The player can walk inside and view the pieces, as well as exploring outside to find other galleries.

adds little additional touches on top of the game, such as a procedural system for applying accents to the pre-written dialogue. This takes dialogue written in plain English and then adds affectations to it to simulate a character who is drunk or has a particular speech impediment. To our knowledge this is a completely novel idea for content generation in games.

### Secret Habitat

Secret Habitat is an 'art gallery simulator' and ambient exploration game by Strangethink. The game has no obvious end state. Instead, the player is encouraged to enjoy walking around the game's generated world, entering the various buildings, viewing artworks and listening to audio recordings, all of which are also procedurally generated. The game seems to appeal to a particular aesthetic of wonder and mystery associated with generative software. One journalist wrote about the game:

> *[The paintings] seem to use a similar algorithm, or similar parts, or similar something, as colours, patterns, and other motifs repeat across them; you can recognise they're part of a series. Seeing different spins on common themes can be delightful, and it's awfully exciting when you discover one painting very different to the rest of its set. (O'Connor 2014)*

Strangethink's biography simply reads *I make strange computer worlds*[8], and Secret Habitat leans towards digital, interactive art as much as it does the traditional ideas of videogames as systems of rules and objectives. Procedural generation has much overlap with both game development and interactive art, and generative software has a unique value in being able to present extremely large or infinite scales to a user (Betts 2014).

## Analysis of Entries

PROCJAM received 138 entries in total, although more entries may exist that were not officially submitted, since we are aware that the jam was set as a class assignment in at least two universities and not all students submitted their entries to the site. All of the jam entries are available online[9]. The entries include full games, prototypes, tech demos, tools and libraries designed for developer use, as well as standalone generators and art pieces. We encourage the reader to visit the site and browse the entries themselves.

By way of a brief survey, we categorised the entries to the jam into two categories: *game*, or *tool*. The categories are defined loosely as follows: a **game** is any software designed to be interactive, but not for the purposes of producing something; a **tool** is any software designed to facilitate the generation of content as part of a larger creative activity. These definitions are not strict, but we offer them here as a rough partition of the entrants to the jam. Overall there were 79 game submissions and 59 tool submissions.

The games typically involved some kind of generative element in how they set up their game world, such as generating the 3D galleries in *Secret Habitat* or using simulation to create murder mystery scenarios in *The Inquisitor*. Most tools fell into one of two categories: some generated common kinds of content in accessible ways, such as world map generators of which multiple were submitted to the jam. This is partly because procedural generation lacks a cohesive community and established baseline software that solves common problems such as world generation - instead, developers tend to reinvent solutions to common problems repeatedly. We believe this is a key problem PROCJAM can target to benefit the generative software community in coming years. Other tools generated unusual kinds of content which are not commonly seen in games, like *GlyphGenerator*'s alphabets or *Bootleg*'s 3D shoe models. These are exciting because they break new ground in generative techniques and offer new applications for games, similar to those submitted to the jam.

### Computational Creativity Issues

PROCJAM's aim was primarily to produce generative tools and games, and to bring together both novices and experts to try out new ideas and learn more about the field. The theory and practice of Computational Creativity is gaining awareness in generative communities, but we believe that many developers are not confident about how to concretely use these ideas in the software they are building. That said, many entries to PROCJAM touch upon issues in the field, and others show clear areas where they could be extended to take advantage of results from Computational Creativity research.

Many of the tools explore co-creation, in which the software either creates alongside the user or tries to assist the user in achieving a particular goal, such as (Liapis, Yannakakis, and Togelius 2012). *Synthetic Poetry* allows the user to write poetry on alternating lines, along with one of three poet models based on Keats, Shakespeare and YouTube Comments. Other entries were more straightforward tools, such as *Nodemancer*, which allows users to specify the components of an item, such as a sword, and then lets the system design the specific details autonomously. Most

[8] https://twitter.com/strangethink23

[9] http://itch.io/jam/procjam#entries

tools focused on the user retaining control, however: *SPARTAN* proudly announces that 'the user has complete control over every step of the generation' – encouraging people to explore ideas that break ideas like complete user control is something that will need to be emphasised in future years of the jam. We hope to expand the jam's resource pool to include tutorials on basic Computational Creativity techniques and perhaps an invited talk from a Computational Creativity practitioner in a future event.

Issues relating to framing and context, as in (Charnley, Pease, and Colton 2012), arise in several entries including games like *The Inquisitor* which generate text as part of gameplay (for example, to provide dialogue and scene-setting for the murder mystery). This text is partly game content but also acts as contextual information that justifies decisions made by the generative system in producing other content. We have argued in the past that framing for game content generation is a broader concept than simply being 'wall text', and can extend to text that appears in-game to help the player understand and contextualise generated scenarios and systems (Cook 2014). Many of these games are beginning to explore these ideas, and we hope to see this trend continue in the future of the event.

Also related to framing, several entries play with the problem of communicating the logic, internal representation or behaviour of the generative system. Both *Diversitizer* and *Meadows* present the player with a natural environment populated with various flora. The locations of each plant, as well as its properties, are governed by procedural systems and vary each time the game is started. The player can gain an understanding of these parameters and the expressive range of the generator by observing the environment and repeatedly generating new worlds, even though the software does not communicate any information to the player through text. In this way, discovering the decisions made by the software become part of the purpose of interacting with the artefact, which is an interesting kind of *implicit* framing that is not often discussed in Computational Creativity discourse.

Many entries to the jam have clear ways in which they could be extended using techniques from Computational Creativity if the developer wished. Identifying simple ways in which common game ideas can be extended is important both in planning 'code camp' events for Computational Creativity, and for giving compelling examples at events like PROCJAM to show developers steps they can take to begin exploring the field. Many generative systems use parameters selected by the developer, such as *Infinity Explorer* which generates 3D worlds for the player to fly around in an airship. Encouraging developers to build their systems such that they can select parameters either based on external, contextual factors as in (Colton, Goodwin, and Veale 2012) or by evaluating its own output as in (Smith and Mateas 2011) is a good way to begin to move some of these generative systems in new directions.

The idea of the software evaluating its own work seems to be one of the most accessible ideas from Computational Creativity that generative software developers can start experimenting with. Generative systems tend to be developed in such a way that they are guaranteed never to produce bad output: in other words, they rely on reorganising hand-made elements that the developer knows in advance will produce reliable content. This is an effective method for game development as it ensures the player will not be disappointed, however the culture of experimentation that we tried to encourage makes PROCJAM an ideal place for people to try out ideas that are less robust but perhaps more interesting and experimental.

## Discussion

PROCJAM has relevance to the Computational Creativity community because it represents the founding of an interdisciplinary community of generative programmers who we hope will, over time, be introduced to and begin experimenting with ideas and approaches from Computational Creativity too. The results of the jam and the details about its organisation are important in their own right; that said, we believe that PROCJAM also holds interesting potential for future events that could strengthen and broaden the appeal and reach of our field.

Computational Creativity is a relatively young academic field that is still laying some of its foundations (Colton and Wiggins 2012). At the same time, many of the aims it has and the technologies and techniques it uses are highly relevant to movements in digital art, videogames and web culture as it stands today. In much the same way that outreach events must target academics in related fields, we should also look to engage these non-academic communities, to share solutions, and to encourage the adoption of our ideas. We often use the term 'mere generation' as a way of describing purely generative software, but we must also bear in mind that a lot of exciting and interesting work is being done in generative software communities, and we should seek to engage with these communities, learn from them, and try to convince them that ideas from Computational Creativity are exciting and interesting, too.

The format of an event like PROCJAM, particularly with some of the changes we made that we discussed earlier, make it ideal for informally bringing together several communities at once, making new connections and allowing them to demonstrate their working practices and techniques to one another. It also serves as a small-scale and self-contained event to set for students who may be interested in the field; PROCJAM was a credit assignment for one university class in particular, and we understand that feedback from the students was extremely positive. With encouragement and additional resources about Computational Creativity, future versions of PROCJAM (or perhaps a separate Computational Creativity jam) could serve as informal, global workshops that introduce people to the area in a practical way.

Despite their short length and highly applied nature, jams can serve a similar purpose for researchers as they do for programmers. There already exist examples of published research work which started off as a jam submission, in which an idea was quickly prototyped and then later developed after the jam (Cook and Colton 2014). PROCJAM also played host to jam games which were implemented to demonstrate an existing research tool or technique in a more concrete

way (Cerny 2014). We hope that subsequent PROCJAMs will see more researchers from this community take part to produce games or tools that demonstrate their work to game developers.

PROCJAM also leaves a legacy of code and ideas that persists after the jam has ended and gives the event lasting value in the months when it is not running. PROCJAM's 138 submissions offer ideas and inspiration, and in some cases code samples and open source projects. Entrants to the jam have already collaborated with one another, expressed intentions to develop their entries into full games, and in one case an academic issued an open call to the PROCJAM community for PhD applications, which was taken up by one entrant. PROCJAM had multiple features written about it in industry magazines and several others on major websites[10][11], and many of the games created for the jam were individually featured and written about as well. The jam's slogan, *Make Something That Makes Something*, has reappeared in other events relating to procedural content generation too[12]. All of this shows that the jam is more than just the week in which it is held – it has a larger impact by creating a community of people that we hope will thrive.

To build upon this, we are planning for PROCJAM 2015 to have more resources ready online before the jam begins, aiming to encourage newcomers to writing generative games, tools and software. Through talking to entrants, we've identified several ways in which we can make the event easier to enter for people. We are hiring an artist to produce some public domain art assets for people to use, specifically designed to be easily recombined and mashed up in procedural systems. We're also talking to developers and researchers with the aim of producing some short tutorials that demonstrate simple generative techniques. These resources will persist beyond the jam itself, hopefully making it easier for people to begin learning about generative techniques at any time of year. We intend to include ideas from Computational Creativity among these resources, in the hope that it will encourage people to think of these ideas as being as essential as any algorithm for making things that make things.

Next year we hope to run some analysis on the entrants to the jam, primarily through optional surveys. This will help us get an idea of the jam's makeup, and people's motivations for entering the event. We are concerned that the lack of evaluation will make the jam complicated for people to curate and explore afterwards, and also acknowledge that some people will be interested in being rated by their peers. We are still reviewing our decision to remove rating altogether from the jam - we may make alterations in 2015 to improve filtering and curation, although it is still unlikely we will implement global ratings that declare overall winners, as we felt the lack of rating contributed a lot to the jam's informal atmosphere.

[10]http://tinyurl.com/procjampcgamer

[11]http://tinyurl.com/procjameurogamer

[12]http://tinyurl.com/aigameslecture

## Conclusions

In this paper we reported on the first procedural generation jam, or PROCJAM. The event was designed to create a new community around generative techniques for games and other software, with an emphasis experimentation, sharing of ideas and introducing new people to writing generative code. We described the changes we made to the classical game jam format to encourage more participants and make the event more accessible. We believe we were successful in this regard, but we also know that there is a lot of work left to be done in maximising the event's impact and accessibility, which we hope to address in future years. We then showed some illustrative examples from the 138 entries received, and discussed the potential for jams to impact communities close to Computational Creativity and potentially nurture relationships and collaborations between them.

## Acknowledgements

Thanks to the reviewers which helped improve the paper, particularly the introductory segment on procedural generation. We would like to thank Azalea Raad for her help organising the event as well as the speakers, press and entrants who supported the jam. We wish to thank PROSECCO for kindly sponsoring the day of talks that launched the jam. Thanks also to Sophie Houlden, whose Fishing Jam inspired several of the changes to PROCJAM's format. This work was sponsored by EPSRC grant EP/L00206X.

## References

Betts, T. 2014. *An Investigation of the Digital Sublime in Video Game Production*. Ph.D. Dissertation, University of Huddersfield.

Bytten Games. 2014. Lenna's inception. http://lennasinception.com/.

Cerny, M. 2014. Machine understanding for interactive storytelling. http://martin-cerny-ai.itch.io/amuse.

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*, 77–81.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI 2012 - 20th European Conference on Artificial Intelligence.*, 21–26.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Cook, M., and Colton, S. 2014. A rogue dream: Automatically generating meaningful content for games. In *Proceedings of the First Experimental AI in Games Workshop, AIIDE 2014*.

Cook, M. 2014. If only we could talk to the generators. Talk at the Procedural Content Generation Summit at ITU Copenhagen.

Gray, K.; Gabler, K.; Shodhan, S.; and Kucic, M. 2005. How to prototype a game in under 7 days. http://tinyurl.com/7dayprototype.

Hecker, C. 2012. The depth jam. http://chrishecker.com/The_Depth_Jam.

Interactive Data Visualization, I. 2002. Speedtree. http://www.speedtree.com/.

Liapis, A.; Yannakakis, G.; and Togelius, J. 2012. Co-creating game content using an adaptive model of user taste. In *Proceedings of the Third International Conference on Computational Creativity*.

Liu, H., and Singh, P. 2004. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal* 22:211–226.

Llano, M. T.; Hepworth, R.; Colton, S.; Charnley, J.; and Gow, J. 2014. Automating fictional ideation using conceptnet. *Proceedings of the AISB14 Symposium on Computational Creativity*.

O'Connor, A. 2014. Time to admire a secret habitat. http://www.rockpapershotgun.com/2014/12/17/secret-habitat-art-gallery-simulator.

Smith, A. M., and Mateas, M. 2011. Knowledge-level creativity in game design. 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*, 16–21.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Trans. Comput. Intellig. and AI in Games*.

Toy, M.; Wichman, G.; Arnold, K.; and Lane, J. 1980. Rogue.

Yu, D. 2009. Spelunky.

Zook, A., and Riedl, M. O. 2013. Game conceptualization and development processes in the global game jam. *Proceedings of the Foundations of Digital Games (FDG'13)*.

Zucconi, A. 2014. 0rbitalis. www.orbitalis.org.

# SMUG: Scientific Music Generator

**Marco Scirea, Gabriella A. B. Barros, Noor Shaker**
Center for Computer Games Research
IT University of Copenhagen, Denmark
{msci,gbar,nosh}@itu.dk

**Julian Togelius**
Department of Computer Science and Engineering
New York University, NY, USA
julian@togelius.com

**Abstract**

Music is based on the real world. Composers use their day-to-day lives as inspiration to create rhythm and lyrics. Procedural music generators are capable of creating good quality pieces, and while some already use the world as inspiration, there is still much to be explored in this. We describe a system to generate lyrics and melodies from real-world data, in particular from academic papers. Through this we want to create a playful experience and establish a novel way of generating content (textual and musical) that could be applied to other domains, in particular to games. For melody generation, we present an approach to Markov chains evolution and briefly discuss the advantages and disadvantages of this approach.

## Introduction

Some traditional works in music or lyrics generation already take into account real-world information. For instance, Colton et al.'s work creates a mood based on a newspaper article, and uses this to generate a poem (Colton, Goodwin, and Veale 2012). In general song composition process, the composer takes inspiration from his life experiences and perceptions of the world around him. This can enrich the final result, creating meaningful pieces of melody, harmony and/or stories.

Dynamic music generation in itself is not novel. Algorithmic music composition has been actively researched for the last several decades, using a large variety of approaches. Some examples include Mezzo's take into creating Renaissance style music through manipulation of *leitmotifs* (Brown 2012); the Cell-based approach (Houge 2012) used in Tom Clancy's EndWar; and the use of neural networks to create musical improvisations (Smith and Garnett 2012).

This work attempts to create lyrics from academic papers and appropriate melodies to go with them. We believe this system can also be modified to use different initial data sources, be it text sources for the lyrics or music sources for the music style. We chose academic papers as input due to their diversity and availability. Furthermore, due to their usual seriousness, it was our opinion that it would be amusing, not only for readers but also for authors, to see these works in a different light.

We believe that this system has value in being an interesting novel idea, and for creating a playful experience with something that, generally, very much lacks fun and playfulness.

We also see the proposed approach applicable in multiple areas. The most interesting for us would be in games: we think that our system (or a fork of it) could be used to improve player experience. For example, to create content for games where story is expressed through music (e.g. Karmaflow (Karmaflow ) or Brutal Legend (Studio 2009)). Or by increasing re-playability and personalized content creation in games where music plays an important part in, either as ambiance or gameplay. Some adventure games even use music in small game sections to remind the player of the game's story or to provide a little comic relief moment (e.g. Deponia (**?**)).

This paper is divided in six sections. The following section (2nd) will describe background theories that we have adopted and the state of the art of research in those particular areas. The third and fourth section will present our approach for music and lyrics generations respectively, giving special attention to our algorithms' behaviours. Then we will present our results and, finally, section six will discuss these and expose our conclusions.

## Background

### Lyrics generation

Natural Language Generation, a sub-field of natural languages processing, has been the focus of several studies across the years. It includes creating text which is contextual, grammatical and lexical coherent, and is strongly related to poetry and lyrics generation.

One of the most important works in poetry generation uses a grammar-driven approach to create poetry, out of a given subject, that is metrically constrained. This work define three evaluation criteria to poetry generation: grammaticality, meaningfulness and poeticness(Manurung 2004). Grammaticality means that the poetry/lyrics must follow linguistic conventions dictated by a grammar; meaningfulness states that the work must convey a context or message that is understandable; and poeticness involves poetic aspects, such as rhyme and rhythm.

A different approach uses a corpus-based approach to

write lyrics about an user-specified theme (Toivanen et al. 2012; 2013). It copies a piece of text (in this case, a poem) and iteratively alters it, changing the words one by one. These words are extracted from a graph and are morphologically similar to the original. The novelty of the final piece is evaluated by calculating how many words were changed.

Oliveira's "PoeTryMe"(Oliveira 2012; Oliveira et al. 2014) uses semantic networks, generation grammars and sets of relation instances to create sentences. Nguyen and Sa generate rap lyrics, by extracting words from a database of real rap songs, and a rhyming database produces words that rhyme with the extracted ones (Hieu Nguyen 2009). Finally, they combine them into a fixed song structure.

There has been a great amount of work dedicated to create Tamil lyrics. Tamil is an old language spoken mainly in Tamil Nadu and Sri Lanka, with literature that goes back two thousand years(Suriyah et al. 2011). Sridhar et al(Sridhar et al. 2014) use the ontological meaning of a scene and a N-gram based approach to generate verses in this language. It identifies syllable patterns for the lyrics, and then create sentences that match said patterns.

Case-based reasoning has also been applied by the COLIBRI poetry generator to generate poetry from text provided by the user (Díaz-Agudo, Gervás, and González-Calero 2002). The quality of this approach results rely heavily on the quality of the original user-given text.

It is also possible to find applications online for this purpose. Country Western Song Machine[1] randomly creates country musics using a templates, and can output a very large amount of possible combinations. The Romantic Love Poetry Generator[2] uses pre-defined templates and user inputs to create poems. The words simply replace specific spaces in the template. Similarly, the Song Lyrics Generator[3] allows the user to select a style (e.g. "Freestyle" or "Love song") or an artist (e.g. "The Beatles" or "Katy Perry"), and to fill a form, that varies according to the style/artist. The form answers replace words in real music.

Our method differs from previous work in the sense that we extract structures from real songs, unlike (Oliveira 2012; Oliveira et al. 2014) extraction of words or the use of templates. Thus, we believe our system can allow for more diversity and expressiveness. Also, none of the cited works use the same input as we do (scientific papers), and very few try to parse information about the real-world into lyrics.

## Music generation

Procedural generation of music content is an interesting field which has received much attention over the last decade. Examples of research on this topic range from creating simple sound effects, to avoid repeating the same clip over and over, to create even more complex harmonic and melodic structures (Shaker, Togelius, and Nelson 2014). While many games use some sort of procedural music structure, there are different approaches (or degrees), as suggested by Wooller *et al.*: *transformational* algorithms and *generative* algorithms (Wooller et al. 2005).

Transformational algorithms act upon an already prepared structure, for example by having the music recorded in layers that can be added or subtracted at a specific time to change the feel of the music (e.g., *The Legend of Zelda: Ocarina of Time* (Nintendo 1998) is one of the earliest games that used this approach). Note that this is only an example and there are a great number of transformational approaches (see GenJam (Biles 1994) and Music Sketcher (Abrams et al. 1999)), but we won't discuss them in this paper.

Generative algorithms instead create the musical structure themselves, which increases the difficulty in maintaining consistency between the music and the game events. This approach usually requires more computing power as the musical materials have to be created on the fly. An example of this approach can be found in *Spore* (Maxis 2008): the music written by Brian Eno was created with Pure Data, where many small samples created the soundtrack in real time. Also note that hybrid approaches are possible, see Experiments in Music Generation (Cope 1996)

In this project we adopt the generational approach, although limited to the generation of melodies. The motivation for us choosing this approach is that we believe we can create more novel content this way, instead of applying transformations to already existing content. Another pitfall of the generational approach is the amount of time necessary for generating the content; in our case, as the evolution of the Markov chains that will generate the melody is done *a priori*, we have a very fast (and inexpensive) generation of melodies.

# Lyrics generation

The lyric generation process used in this approach takes as input an academical paper in PDF format, and output a series of verses. It has two main steps: pre-processing and lyric generation.

## Pre-processing

Pre-processing involves populating databases of words (and their stems) and song structures. It needs to be executed only once, prior to the first lyric generation. Firstly, the word database was populated using Google searches for lists of word types (e.g. verbs, prepositions, pronouns). For each word in the database, its stem value was also extracted using SnowbalStemmer(Porter and Boulton 2001).

Afterwards, it was necessary to populate the structure database. By structure we define a group of word types in sequence that represent a sentence. For instance, the structure for "We see our big, blue sky" would be "Pronoun verb pronoun adjective comma adjective other". Possible values for the structure are: *verb*, *pronoun*, *preposition*, *adjective*, *adverb*, *conjunction*, *other*, *onomatopoeia*, *comma* and *dot*. "Other" represents both nouns and words that may not fall into other categories. We chose to use it, instead of "noun",

[1]Country Western Song Machine, 1998, http://www.outofservice.com/country/

[2]Romantic Love Poetry Generator: http://www.links2love.com/poem_generator.htm

[3]Song Lyrics Generator: http://www.song-lyrics-generator.org.uk/

because it allows a higher level of diversity while choosing the word. This way, not only can we choose a noun, but also any of the other categories previously mentioned as well. Onomatopoeia is an other value with less than three letters (e.g. "Po-po-poker face" would be represented as "onomatopoeia onomatopoeia other other"). These types are represented, in code, as integers.

To identify structures in real songs, a group of 50 songs were analysed. These songs were selected from famous artist (e.g. Rihanna, Michael Jackson), using as criteria that all songs need to be in English and there cannot be more than 3 songs per singer. For each sentence in the lyrics, the algorithm extracted its equivalent structure which is then inserted into the structure database.

### Lyric generation

The process for generating lyrics is divided further into three steps: parsing and analysis of paper, creation of song structure, and lyrics word generation.

In the first step, the algorithm receives a PDF file containing the paper and extracts its words using the PDFBox library[4]. Then, the text is processed, removing everything before the abstract and after the references. This aims at avoiding inputting data that will not significantly improve the user's understanding of the paper. If the system cannot identify the abstract or the introduction (in the absence of the abstract), it will start at the very beginning.

In order to evaluate the importance of each word in the text, a word count is performed. It uses the stem value of the word, and is calculated as the sum of all occurrences of words derived from this stem, in the text. For instance, assume that "wait" appears once and "waiting" appears twice in text. The count would be 3 for both of them, because they have the same stem "wait". Also, each word was added to a collection of values types present in paper, according to their value type (see Section Pre-processing).

Secondly, the algorithm randomly selects a number of structures from the database. They will represent the total structure of the music, i.e. each structure will represent the structure of a line in the final lyrics. For the purposes of this paper, all songs have a total of 24 structures, divided into 6 groups of 4 structures each.

Finally, for each structure chosen, a sentence is created according to type values in the structure. *Comma* and *dot* values are translated directly into "," and ".". Types *verb*, *pronoun*, *preposition*, *adjective*, *adverb* and *conjunction* trigger a roulette selection among all words from that type that appeared in text. This selection uses the word count as probability. *Onomatopoeia* inserts either "aah", "ooh" or a random word from text with its start repeated (e.g. "ta-ta-taxonomy"). Lastly, *other* trigger a roulette selection with all words in text.

## Music generation

To create a melody we decided to use two Markov chains. These are mathematical systems that undergo transitions from one state to another on a state space (Norris 1998). A Markov chain is a stochastic process with the Markov property: the next state to be selected only depends on the previous one.

Markov models can be trained using existing sequences of events (e.g., words in a book, or notes in a musical piece) and, once trained, be used to generate a new sequence of events statistically similar to the training data. It is highly unlikely for a Markov model to recreate an exact training sequence as it contains an intrinsic stochastic element. However this depends very much on the training data. An important limitation of Markov chains is that they capture statistical similarities only on a local scale, and not on a high level; this means that we lose information of structures like repetition of musical phrases in different part of the composition. Nevertheless, even with this disadvantages Markov chains have classically been extensively used for the purpose of melody generation, as they can be trained easily to create sequences of notes (Ames 1989).

Examples of research that use Markov chains and Evolutionary Algorithms are Manaris *et al.*'s *Monterey Mirror* (Manaris, Hughes, and Vassilandonakis 2011) and Bell's work (Bell 2011). Manaris' work focuses on evolving Markov chains to obtain the rare chains that will with high probability reproduce high-level structure (repetition of entire phrases or in general more structured music) while Bell's work uses interactive evolution to produce chains that create music pleasant to the listener. These are much more complex works that generate complete music and not just melody, as in our case.

There are some reasons why we have decided to approach the creation of these Markov chains through such an unorthodox method of using evolutionary algorithms (unorthodox only in this particular application of course). Using traditional (EM-based) training would have been faster and easier, yet it is in its nature to lead to an overfitting of the chain to the training set. What we hope to achieve through our approach is obtaining a chain that would reflect the characteristics of the training set while avoiding overfitting: in short obtaining a chain that reflects the characteristics of the training set while maintaining some diversity.

Another interesting feature that this approach gives us is introducing constraints through the fitness function. This gives us the option of tuning our chain in more interesting ways (of course this means in parts deviating from the training set, but that's exactly the point). These constraints could be musical rules, for example avoiding too large intervals between notes. We discuss these in the **fitness function** section.

### Markov chains and Representation

In our approach we decided to use two Markov chains: one to determine the notes of our melody and another one to select the duration of these notes. Markov chains can be expanded to include some memory of the previous states by considering as state not only the current one but some of the previous ones. The amount of previous states we "remember" is called *order* of the Markov chain; if we consider a chain of order 2, it means that every state is a couple con-

[4]PDFBox is a Java open-source PDF library: https://pdfbox.apache.org/

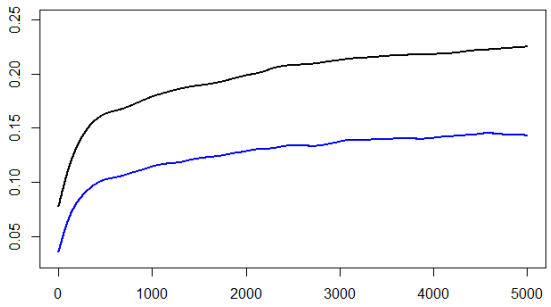



Figure 1: Fitness changes in the **notes** Markov chains population during 5000 generations. In black is represented the fitness of the best individual of the generation, while in blue is the average fitness of the population.

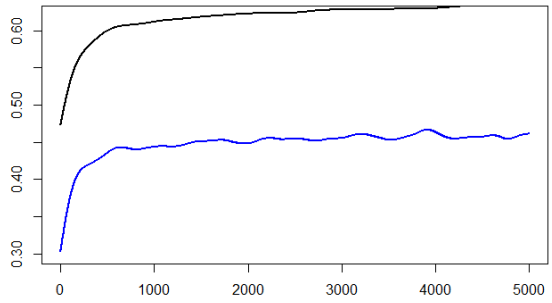



Figure 2: Fitness changes in the **durations** Markov chains population during 5000 generations. In black is represented the fitness of the best individual of the generation, while in blue is the average fitness of the population.

sisting of the previous note and the current one. In our final implementation we decided to use an order 2 chain.

We implemented a Markov chain as a hash-map. Labels (or keys) are the name of the state (the current note), and values are another hash-map containing the probabilities of choosing a note (transition) from the current state. We also adopt this hash-map as our **genotype**. You could visualize it as a labelled bi-dimensional matrix, with as labels states and transitions, the next state can be calculated as: (previous state - older note) + transition.

The state space can be calculated as $\frac{n!}{o!(n-o)!}$ where $n$ is the amount of notes we consider and $o$ is the order of the chain. In the case of our order 2 chain, where we consider 3 octaves (36 notes) it would be $\frac{36!}{2!(36-2)!} = 630$. To restrict this space we apply restrictions to remove states which we consider not to be good, in particular all the states that contain a transition between notes with intervals higher than an octave. To avoid leafs in our chains we do not allow for allowed states to have a 0 probability to move to any other (allowed) state.

To extract musical information without be restricted by the key of the song, we have our Markov chain for note generation work by *degrees*. In music *degree* is defined as the position of a note in a specific key's scale: for example a C can be considered differently depending what is the key of the song, in a C major song it will be a *I*st degree, while in a G major song it would be a *IV*th degree (as the scale of G would be [G A B C D E F♯]).

## Evolving Markov Chains

We evolved our Markov chains using a genetic algorithm. The parameters used for our final chains are:

- Population size = 200
- Generation number = 5000
- Elitist factor = 1/4 (this means that we keep the best 1/4th of the population in the next generation)
- Mutation chance = 10%

The procedure to create the new generation is to copy to the new one the best individuals of the previous, then we fill the rest of the population with offspring of randomly selected individuals from the previous generation. Finally each individual has a chance of mutating.

To create offspring we use a one-point crossover approach: we select a random index of the states in our Markov chain (hashmap) and we create two new chains, the fist will contain the values of the first parent until the index and the values of the second parent for the following ones, while the second one the opposite. Because of the way we are representing the chains all of them always have the same amount of states, so the only thing changing while doing crossover are transition probabilities between states.

We are aware that using crossover will sometimes lead to broken Markov chains, with some orphan sub-chains that will result unreachable. This is an inherent issue with crossover, but we assume that a broken chain with high fitness will still present the characteristics that we desire and through our elitist strategy we will be able to preserve the individuals that presents good gene combination, be they broken or not. Vice versa, a broken chain with low fitness has a higher chance to be replaced.

To mutate a chain we consider a chance of $\frac{1}{numberOfStates}$ for each state to randomize it's transitions; this way we will statistically only have one state changing when mutating the chain, but still allowing for bigger mutations (or no mutation) to happen.

## Fitness function

The fitness function we have chosen to apply for the evolution of the chains can be described as:

$$f = \sum_{S_i \in Songs} PredictRew(S_i) - ConstraintsPen$$

where $Songs$ is a set of melodies from existing songs, $PredictRew(S_i)$ is defined as the probability of the Markov

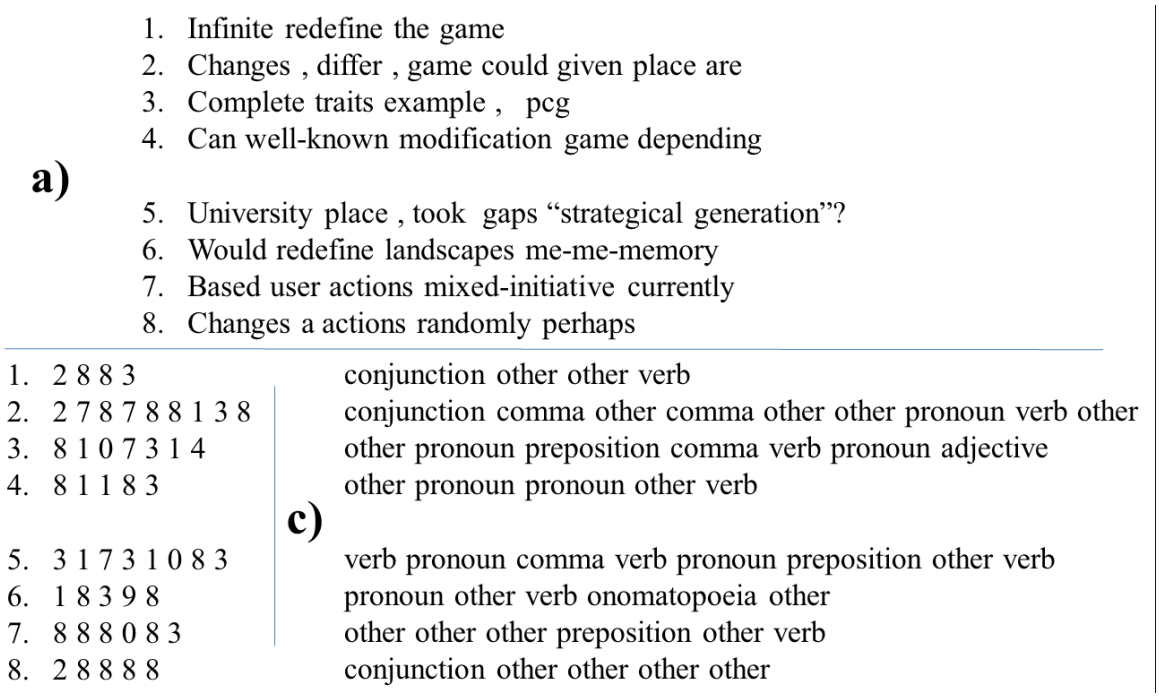



Figure 3: Output of program the program for a paper by Togelius et al.(Togelius et al. 2011). a) Actual lyric generated. b) Random structure used, represented in code. c) Same structure, in natural language.

chain we're currently evaluating to predict the melody in the song $S_i$ and $ConstraintsPen$ is the penalty assigned to the chain according to the constraints we want to apply to it.

By considering $S_i = \{n_0, n_1, ..., n_k\}$, where $n_i$ is the *i*-th note in the melody and $k + 1$ is the amount of notes in the melody, we calculate $PredictRew(S_i)$ as:

$$PredictRew(S_i) = \sum_{\{n_i, n_{i+1}, n_{i+2}\} \subset S_i} P(n_{i+2}|n_i, n_{i+1})$$

where $P(n_{i+2}|n_i, n_{i+1})$ is the probability that the Markov chain we are evaluating presents for the transition $n_{i+2}$ from the state $(n_i, n_{i+1})$. To make a practical example, if the song presents a sequence of the type $(C, D, E)$, the fitness of the chain will increase by the probability it has of making the transition $E$ from the state $(CD)$.

$ConstraintsPen$ is composed by two rules we introduced to eliminate cases we consider musically uninteresting:

$$ConstraintsPen = BigLeap + SameNoteLoop$$

where $BigLeap$ is defined as:

$$BigLeap = \sum_{n_k|(n_i,n_j)\in Chain} P(n_k|(n_i, n_j))$$
$$\text{if } |(n_k - n_j)| > 12$$

So it will increase for every transition that appears in the chain that presents a voice movement bigger than an octave (e.g. $(C1, C1) \rightarrow D2$ ). $SameNoteLoop$ is instead defined as:

$$SameNoteLoop = \sum_{n_i|(n_i,n_i)\in Chain} P(n_i|(n_i, n_i))$$

This way we will have a higher penalty for transitions that keep us in the same state when the state is comprised of a couple of identical note (e.g. $(C1, C1) \rightarrow C1$ ).

The fitness function for the chain that will determine the duration of the notes (instead than the notes themselves) is evaluated the same way, but without $ConstraintsPen$, as these constraints are pitch specific.

Our $Songs$ set consists of 20 songs taken from a list of most popular pop songs. It presents a variety of styles, but all the songs are in a major mode. This limits our generation to melodies in major mode, while for minor melodies we would have to evolve a new chain using a set of songs in minor key. This is necessary because the intervals between the notes in a major and minor scale differ, making us hypothesize that our chain will only be able to produce melodies appropriate for the key of the songs used to calculate the fitness.

The elements of the set are the voice track from the songs; we have isolated the voice melody and stored it in a MIDI file, from this file we extract the degrees of the notes of the melody (by considering in which key the song is) and the duration of these notes. We will then use these values in the evaluation of our chains (remember that to abstract the key our chain work by *degrees*).

## From text to melody

To create a melody to go with some particular lyrics we our method is:

1. Find the total amount of “syllables”. In this case we con-

sider a simplistic concept of syllable: we consider a syllable for every time we encounter a vowel (groups of vowels are considered as part of the same syllable).

2. Create as many notes as the syllables in the lyrics using the notes chain
3. Define the duration of the notes using the durations chain
4. Add rests after each word (with a 30% chance that there is going to be no rest)

Finally, for easy usage and visualization of the melody we produce a midi file representing our melody.

## Results

### Lyrics

Figure 3 shows two verses of lyrics generated using Togelius et al. paper (Togelius et al. 2011), and it's basic structure. It is possible to notice some degree of understanding in the sentences, and diversity in word choices.

Figure 4 shows a small part from lyrics generated by the system using Darwin's paper (Darwin 1991), with melody. Another verse from the same work goes as follows:

Natural who in, throw all selection
Re-re-related nature relations law on any
Cl-cl-class that false but it inhabitants generic
It natural origin its, species to sp-sp-special and on its
.
That more be all, – reflecting
Each relations on these – natural
Each dr reflecting gr-gr-grouping circumstances
Selection introduction

Figure 5 show some verses from a song generated with this paper. In a different iteration, the following verses were generated:

Parent chain mutating
Another should, we pre-processing musical be with pr-pr-pre-processing states
That songs structures that, sridhar, other possibility use
Im-im-improve, pr-pr-priori, ad-ad-add, im-im-improved, rh-rh-rhyming, on-on-on, ooh
,
Papers to music lyrics generation, figure
Generation that, consider, generate, correct, with we is using
Restrict input and note statistical final as music
Create it, approaches, create, be, into it chains generated

### Music

In this section we'll try to analyse some of the melodies our generator produced.

In figure 4 we can see an example of a melody generated by our system from Darwin's paper *On the origin of species by means of natural selection*, the generation of melodies is very fast, as the training is done *a priori*. Interesting to note is how our generator doesn't create melodies that strictly stick with the diatonic scale but introduces alterations.

In the figure we can see how in the fifth bar it lowers the VII degree to a B♭, and more interestingly how it presents the note again on a different octave. Looking at the other notes played in the chord we can recognize how the chord underlying the measure could well be a C7 with the omission of the V degree [C E B♭]. While this chord goes out of the normal key it is not uncommon to use it in this key and it doesn't necessarily signify a change of key.

Another example generated from this paper can be observed in figure 5. This score shows even more alterations than the other one with a more dissonant and almost jazz-like feel. Interesting to note how musical passages seem to emerge and be repeated with alterations: for example the succession E-D-C (bars 1, 2 and 3, with a rest in the latter) and the succession C-C-B[-C] (repeated two times in bar 4 and inverted and transposed just afterwards becoming C-C-D[-C].

Nonetheless, we haven't conducted an evaluation study on the melodies produced so we cannot make any statement on how interesting or musically pleasing the melodies are to the listener. Also we believe that to achieve a more interesting result we would need a harmonic framework to give more musical context to the produced melody; as we discussed in this section we can see some passages that seem to present some chord, but that is a purely emergent behaviour.

## Discussion

This section will discuss our main findings in this project, and final considerations about them and the work in general.

### Lyrics

Regarding lyric generation, although our approach may be perceived by some as simplistic, we believe it is capable of creating relatively fluid and interesting lyrics. The sentences structure seem somewhat sensible, although there are definitely space for improvement. Rhyming also happens in some moments, however it does occasionally, as the current version of the system cannot guarantee rhyming. We intend to correct it in further implementations, perhaps using a rhyming library or accessing a service online to check possible words. This would permit to create musical rhyme patterns (e.g. ABAB or AABB, where A and B represent rhyming endings of sentences). The size of sentences, too, varies, and a syllable measure constraint could help improve it.

Furthermore, it is possible to understand, to some extent, basic ideas transposed from the paper to lyrics. Some words that are clearly significant in the paper also appear in the lyrics. But there is no perceptible line of thought. It would be interesting to take the structure of the paper into account in the generation, by changing the probability value of words according to the current verse number. For instance, in the first verse, words from the paper's introduction would be more likely to be chosen than others. We would also like to try different techniques, such as an evolution strategy, to see if the outcome presents higher or lower semantic meaning in comparison to this approach. Further mechanisms for dealing with the meaningfulness of lyrics need to be applied.

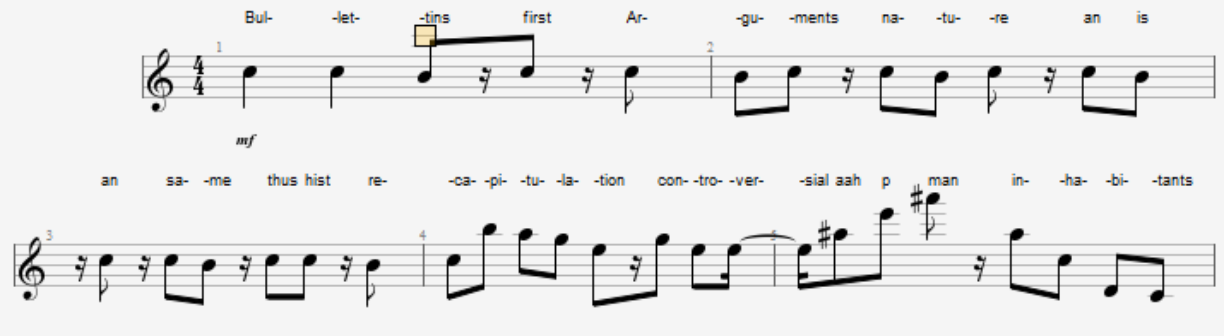



Figure 4: Excerpt from the score generated from Darwin's paper *On the origin of species by means of natural selection* (Darwin 1991) C major.

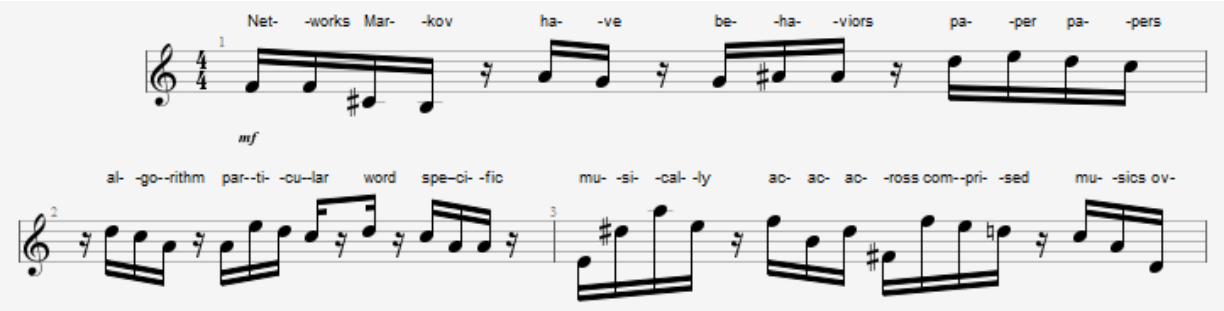



Figure 5: Excerpt from the score generated from this paper in C major.

## Music

The main point we have to discuss is our choice of adopting evolution of Markov chains instead of the more common method of training them. While this method is more time consuming, we believe it is interesting. There is an argument of novelty, because the method of evolving Markov chains for music production, while not completely new, is not very explored.

As we stated at the beginning of the **Music Generation** section, we believe that this method results in lower dependency on the training set than traditional training. We think that, this way, our chains should be able to express a greater music space while maintaining some structure from the training set. One cost we expect to have to pay is a smaller rate of emulation of the training set style. Sadly, at the moment we don't have enough data to support this statement, but an evaluation study is already planned. Another pitfall is the possibility of getting in a part of the melody space where there is not enough information to create musically interesting melodies, degenerating in the worst case scenario to a random search.

As seen in section , we see some interesting emergent behaviour (like the almost key changes and the jazzier sections) which might hint to how the Markov model might not be very effective at producing a coherent whole.

Still, we believe that our approach will be able to capture the style of a specific genre/style of music with a large enough corpus of songs to use in our fitness function. We have to recognize how we might have achieved better results by having a bigger training set, but we believe we have already achieved some very interesting results.

Finally by observing the increase of the fitness function of our evolved population in Figures 1 and 2 we notice how the duration chain evolves much faster and with higher fitness. This is due to the smaller space we consider for this chain, which is less than half of the notes chain's one.

## Conclusions

We have presented a method for creating melody and lyrics using real-world data. To do so, we developed a musical generator that evolves Markov chains to create melodies, and a lyric generator, that extracts content from academical papers and transforms them into songs. We have a fully functional system that complete both tasks, taking an academic paper in PDF format and outputting a melody and the according lyrics. Our generator seems to produce interesting music/lyrics combinations, but we still have to conduct further studies to prove their interestingness. The generator also still shows much room for improvement, as discussed previously, and future work will be in both fine-tuning the evolutionary approach and introducing more features in the lyrics generation, such as rhyming, stricter metric structure and improved semantic content transfer from the original paper. Still, we need to recognize that there might be issues inherent to using Markov chains for melody production that

might not be resolved, like insuring the production coherent whole.

Ultimately, we believe think these techniques might be used in music-based games to add and customize content.

## References


Abrams, S.; Oppenheim, D. V.; Pazel, D.; Wright, J.; et al. 1999. Higher-level composition control in music sketcher: Modifiers and smart harmony. In *Proceedings of the ICMC*.

Ames, C. 1989. The markov process as a compositional model: a survey and tutorial. *Leonardo* 175–187.

Bell, C. 2011. Algorithmic music composition using dynamic markov chains and genetic algorithms. *Journal of Computing Sciences in Colleges* 27(2):99–107.

Biles, J. 1994. Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, 131–131. International Computer Music Association.

Brown, D. 2012. Mezzo: An adaptive, real-time composition program for game soundtracks. In *Proceedings of the AIIDE 2012 Workshop on Musical Metacreation*, 68–72.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Cope, D. 1996. *Experiments in musical intelligence*, volume 12. AR editions Madison, WI.

Darwin, C. 1991. On the origin of species by means of natural selection, 1859. *Murray, London*.

Díaz-Agudo, B.; Gervás, P.; and González-Calero, P. A. 2002. Poetry generation in colibri. In *Advances in Case-Based Reasoning*. Springer. 73–87.

Hieu Nguyen, B. 2009. Rap lyric generator.

Houge, B. 2012. Cell-based music organization in Tom Clancy's EndWar. Demo at the AIIDE 2012 Workshop on Musical Metacreation.

Karmaflow. Karmaflow: The rock opera videogame.

Manaris, B.; Hughes, D.; and Vassilandonakis, Y. 2011. Monterey mirror: Combining markov models, genetic algorithms, and power laws. In *Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, 33–40.

Manurung, H. 2004. An evolutionary algorithm approach to poetry generation.

Maxis. 2008. Spore.

Nintendo. 1998. The legend of zelda: Ocarina of time.

Norris, J. R. 1998. *Markov chains*. Number 2008. Cambridge university press.

Oliveira, H. G.; Hervás, R.; Díaz, A.; and Gervás, P. 2014. Adapting a generic platform for poetry generation to produce spanish poems. In *5th International Conference on Computational Creativity, ICCC*.

Oliveira, H. G. 2012. Poetryme: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence* 1:21.

Porter, M., and Boulton, R. 2001. Snowball stemmer.

Shaker, N.; Togelius, J.; and Nelson, M. J. 2014. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer. (To appear).

Smith, B. D., and Garnett, G. E. 2012. Improvising musical structure with hierarchical neural nets. In *Proceedings of the AIIDE 2012 Workshop on Musical Metacreation*, 63–67.

Sridhar, R.; GANGA, K.; PRABHA, G. D.; et al. 2014. Automatic tamil lyric generation based on ontological interpretation for semantics. *Sadhana* 39(1):97–121.

Studio, D. F. 2009. Brutal legend.

Suriyah, M.; Karky, M.; Geetha, T.; and Parthasarathi, R. 2011. Special indices for laalalaa lyric analysis & generation framework. In *Proc. Internat. Tamil Internet Conf*, 287–292.

Togelius, J.; Kastbjerg, E.; Schedl, D.; and Yannakakis, G. N. 2011. What is procedural content generation?: Mario on the borderline. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, 3. ACM.

Toivanen, J.; Toivonen, H.; Valitutti, A.; Gross, O.; et al. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the Third International Conference on Computational Creativity*.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; et al. 2013. Automatical composition of lyrical songs. In *The Fourth International Conference on Computational Creativity*.

Wooller, R.; Brown, A. R.; Miranda, E.; Diederich, J.; and Berry, R. 2005. A framework for comparison of process in algorithmic music systems. In *Generative Arts Practice 2005 — A Creativity & Cognition Symposium*.

# Generative Mixology: An Engine for Creating Cocktails

**Johnathan Pagnutti** and **Jim Whitehead**
Computer Science Department
University of California, Santa Cruz
{jpagnutt, ejw}@ucsc.edu

**Abstract**

This paper details an expert cocktail generation system. After using expert knowledge to break down cocktails into eight categories, the system generates cocktails from a particular category using a context-free stochastic grammar. These cocktails were then evaluated by human participants in a research setting. Participants evaluated the cocktails on the basis of quality, novelty and typicality to check the creative potential of the generator's output.

## Introduction

Some domains, such as music and visual art, have been studied in depth by the computational creativity and procedural content generation (PCG) communities. Yet other domains, such as preparing food, have not. Part of this is that food preparation is a complex task, not only in dealing with which particular combinations of ingredients should be used, but also how those ingredients should be prepared and transformed into a finished product. Even simple domains, such as chocolate chip cookies, can have many ingredient and preparation step permutations (Kenji López-Alt 2013). However, there is a strong interest in artificial chefs, servers and bartenders, as evidenced by the steady rise of restaurants featuring robotic servers and bartenders (Sloan 2014; Kross et al. 1976) as well as home meal serving and bartending robots (Glass 2014; Monsieur, LLC 2015). The next step for this niche mechanization of the food and beverage industry is to implement an AI system that can create new dishes or drinks to prepare for patrons.

A factor analysis on the Creative Achievement Questionnaire (CAQ), a creativity assessment test, revealed three categories of creative achievement: Expressive (Visual Arts, Writing, Humor), Performance (Dance, Drama, Music), and Scientific (Invention, Scientific, Culinary). This result shows that culinary creativity falls into a similar domain as scientific and innovative creativity (Carson, Peterson, and Higgins 2005). This implies that techniques used in creative recipe generators have applications in problem solving and research direction. Therefore, the development of creative recipe generation and other culinary arts may have applications for more general-purpose problem solving AI.

We can break recipes into two parts: the static ingredient list and the dynamic preparation instructions. The ingredient list is composed of the ingredients that the recipe will use; the preparation instructions are how those ingredients are transformed into a final dish. However, there are a very large number of potential ingredients that could go into any dish, and even more ways those ingredients can be combined to become a final product. Therefore, work with smaller, less complex domains is needed to gain insight into the problem of an artificial chef. One such useful domain is mixed drinks, as the potential ingredient space for cocktails is smaller than that of culinary dishes and the mixing instructions are far simpler, while still retaining a lot of the interesting complexity. As such, we developed an expert system for cocktail generation and evaluated the artifacts it generated to start understanding the nature of computational cooking.

## Related Work

PIERRE (Morris et al. 2012) uses a genetic algorithm to generate crock-pot recipes from a corpus gathered from various websites. The fitness function is based around novelty, trying to maximize the number of rare n-grams in a recipe. Recipes have also gotten attention from case-based AI planners, such as CHEF (Hammond 1986). Both these generators have a high chance to output a 'bad' recipe. PIERRE makes no claims about the quality of its output, and CHEF needs to learn from bad examples in order to create good ones. A similar branch of research to this is JULIA (Hinrichs 1992), which uses case-based design techniques such as case adaptation to determine how to best design and present a meal. Our work does not need to learn from 'bad' examples and attempts to always produce a believable drink.

Pinel and Varshney have worked on a recipe generator (Pinel and Varshney 2014), which unlike PIERRE or CHEF does not deal with a particular style or type of cooking. Using a cognitive model of creativity and a large knowledge base built from scraping recipe wikis, they created a mixed initiative generator that produces ingredient lists and rough steps to completing a recipe. This work is part of a larger system by Pinel, Varshney and Bhattacharjya (Pinel, Varshney, and Bhattacharjya 2015) that generates recipes by mining data from the Wikia recipe repository and Wikipedia to build an extensive knowledge base of recipes. From there, the system uses a mixed initiative approach, in which a new recipe is generated with user-selected categories. Varshney

et al. (Varshney et al. 2013) discuss many of the difficulties in working with recipe generation, emphasizing that how something tastes is actually the result of all five classical senses working together, plus several psychological, neurological and social phenomena.

Cocktail generation has been done, although not on any formal level. The Mixilator (Haigh 2004) is an online cocktail generator based on the writing of mixologist David Embury. The Mixilator picks a random ingredient from each of the three categories defined by Embury, and makes a predefined cocktail from it, with mixing instructions hardcoded. Although it uses an impressive amount of ingredients, the generator is highly constrained—Embury believed all drinks should contain at least three ingredients, so the Mixilator can never create a gin and tonic, for example. The Mixilator also has no knowledge about how combinations of ingredients function. It assumes that, as long as it picks from each correct category, the resultant cocktail will be good. Yet, as the authors point out, no considerations for quality went into its development. While investigating the Mixilator in writing this paper, ingredient combinations like lime sherbet and maple syrup were suggested for cocktails. Another drink called for "2 drops of liqueur" without ever specifying which flavor of liqueur. This makes the Mixilator's output appear more whimsical than structured cocktail generation. While related investigations like the Mixilator are hobbyist projects, and some large scale recipe generators give a passing glance to the cocktail domain, we aim to be the first to take a domain sensitive, computational creativity approach to cocktail generation, and maintain a critical eye towards drink quality and expressive potential.

Mixologists have been inventing new drinks in popular literature. One of the first books on mixology as an art, *The Fine Art of Mixing Drinks*(Embury 1953), by David Embury, details a basic ratio to follow for cocktails, as well as several ingredient categories to use. More recently, *DIY Cocktails* (Simmons 2011), details several basic ratios for a wide variety of drinks. However, mixology books commonly focus on presentation or are just a compiled list of cocktail recipes(such as (Regan 2003; Joseph 2012)). Sadly, mixologist blogs ((Bovis 2015; English 2015; Jamieson 2015), for example) also tend to focus on recipe compilation or product review rather than cocktail theory.

Computational creativity is a vibrant field, with a plethora of definitions, theories and evaluation methods for creativeness in computer programs. Much modern work stems from three techniques (Boden 1998) and their formalisms(Wiggins 2006) for establishing creativity in AI: by producing novel combinations of familiar ideas, by exploring potential conceptual spaces or by making transformations that allow the generation of previously impossible ideas. These relate to creativity in the process of artifact generation. The other side of the coin refers to creativity as a quality in generated artifacts (e.g. the difference between "this painting was made by a creative person" vs. "this painting is creative"). In these terms, metrics for evaluating the creativity of generated artifacts have been proposed (such as (Pease, Winterstein, and Colton 2001)), and we evaluate our cocktails on the categories of quality, novelty and typicality as defined in (Ritchie 2007).

## Expert System Constraints

The cocktail generation system defined here is derived from the rules and opinions of two primary texts: *The Fine Art of Mixing Drinks* (Embury 1953) and *DIY Cocktails* (Simmons 2011). Both texts treat cocktail creation as a process, and outline several basic rules to follow in the terms of ratios and ingredient categories. In addition, they provide mixing instructions for various categories.

Multiple source texts were used to try to minimize the amount of author bias in the system. *The Fine Art of Mixing Drinks* is an older text. Several common modern cocktails are impossible to create by following its rules alone, and today there are far more popular cocktail ingredients than there were in Embury's time. By augmenting Embury's rules with a more modern text, the generator can be more expressive and better reflect modern cocktail design aesthetics.

In addition, *DIY Cocktails* (Simmons 2011) gives a theoretical basis for which ingredients work well together. This helps the cocktail generator avoid various pitfalls in ingredient choice (such as combining a citric acid and a cream, which will curdle the cream), and also be smarter in selecting which ingredients to use to create a cocktail.

Finally, there were some constraints set at the discretion of the authors. Shooters and shots are not considered cocktails, and are ignored. In addition, the generator does not use overproof spirits (those that contain more alcohol than proof spirit), as they can be difficult to acquire.

## Cocktail Properties

We divide a recipe into two parts: the mixing instructions (dynamic instruction) and the ingredient list (static elements). Cocktail mixing instructions are either derived from the ingredients used in the cocktail or previously decided steps in the mixing process. Lighter ingredients (juices and spirits) only require stirring; heavier ingredients (syrups and purees) may require shaking or rolling in a cocktail shaker. There are a few generally uncommon preparation instructions that are more common to cocktails, such as muddling (mashing the ingredient in the bottom of the glass). As it makes no sense to muddle an ingredient in a shaker for mixing and/or rolling (the straining head of the shaker would keep the muddled ingredients in the shaker and not in the glass), step order occasionally matters. However, someone could shake various juices and pour them into an ingredient they had muddled, so keeping track of what process is being applied to which ingredient is important. There are several other ways to mix a drink that deal with spectacle: floating a high proof liquor on the top of a drink before setting the liquor on fire, or floating several ingredients on top of each other to provide a layering effect. These techniques do not have a strong bearing on flavor, so they are not considered by the cocktail generator.

The ingredient list is more complicated. Depending on the source, the raw ingredients of a cocktail can either have many very fine qualities (such as undertone, notes or hints) or be very basic (sweet, sour). This makes it difficult to

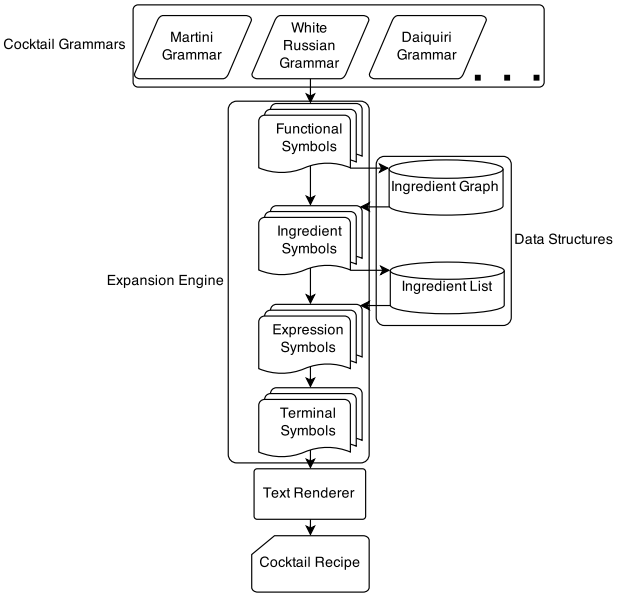



Figure 1: System Architecture. A grammar is chosen from a list of various cocktail grammars and then expanded as a set of symbols, from functional to terminal. For some expansions, symbols are built on the fly by requesting information from an external data structure. Once the grammar has expanded to terminal symbols, it is rendered as a human readable recipe and presented to a user.

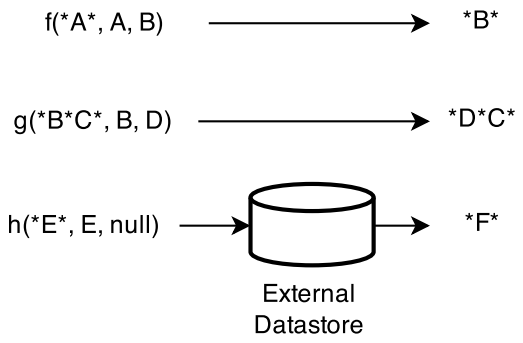



Figure 2: Three general categories of grammar expansion.

ascertain a good top-down or bottom-up model of how a particular ingredient tastes—do notes of elderberry work well with sour? Should undertones be sweet or smoky or smoky-sweet, and does any of that work well with strawberries? However, this also is not how common sense reasoning about taste functions. When someone hears the ingredients in a drink (or dish), they recall what each ingredient tasted like in the past, and make some guesses as to what they might taste like together. The cocktail generation system was built on this basic reasoning concept. Rather than try to accurately model how each ingredient tastes, the generator keeps track, on an abstract level, which ingredients work well together, and then creates drinks with combinations of good ingredients. These ingredient pairings were built based on expert knowledge, rather than a database or chemical hypothesis, as in (Ahn et al. 2011).

In addition, the cocktail generation system breaks cocktails into eight categories. Each of these categories is based around a particular set of exemplars in cocktail literature. These exemplars either all share an ingredient category (such as the use of cream for drinks derived from a White Russian) or a particular ratio (2 to 1 ratios for drinks derived from a Gin and Tonic). It is important to note that all of the International Bartenders Association official cocktails roughly fall into these eight categories. Although this categorical system does not perfectly cover every potential drink, it encapsulates most of the space of potential cocktails.

## Generator Architecture

The cocktail generation system has four main components: a set of stochastic, context-sensitive cocktail grammars, an engine to expand the grammars, a set of outside data structures used to build grammar symbols and a text rendering system to present generated recipes, as seen in Figure 1. All four of these components work in a serial fashion to generate a new cocktail; no part of the system runs in parallel. One of the main difficulties when designing the cocktail generation system was the amount of symbols in the grammar. There are at least 260 symbols, so writing rules out directly would have taken a large amount of human authoring time.

### Cocktail Grammars

Following the research on mixology, most cocktails can be broken down into eight categories of drinks. The categories are all based on exemplar drinks; the Old Fashioned category uses the same ratios present in an Old Fashioned, for example. Sometimes a drink is considered an exemplar because it has a unique and useful ratio (the margarita's 3 parts strong : 2 parts sweet : 1 part sour), or a particularly important ingredient (the cream in a White Russian). Usually, a category also has a trend: Old Fashioned based drinks always have muddled ingredients or syrups, while Margarita-like drinks always use a liqueur as one of their sweetening agents. As such, to capture these trends, a unique grammar needs to be built for each drink category. The categories are Old Fashioned, Martini, White Russian, Margarita, Daiquiri, Mai Tai, Gin and Tonic, and Mojito.

All the grammars use the same set of symbols, but each category has its own unique production rules and constraints. Several rules were reusable (a single context free replacement rule, for example), however, each grammar has several custom, unique rules.

There are three basic ways that a grammar expands, as seen in Figure 2. The first two examples shown here are deterministic, although they both have stochastic variants where a random choice is made from a list of potential symbols. The last example is always stochastic. First, grammar symbols can expand without considering what other symbols are currently in the grammar, commonly referred to as context-free expansion. This expansion is shown at the top

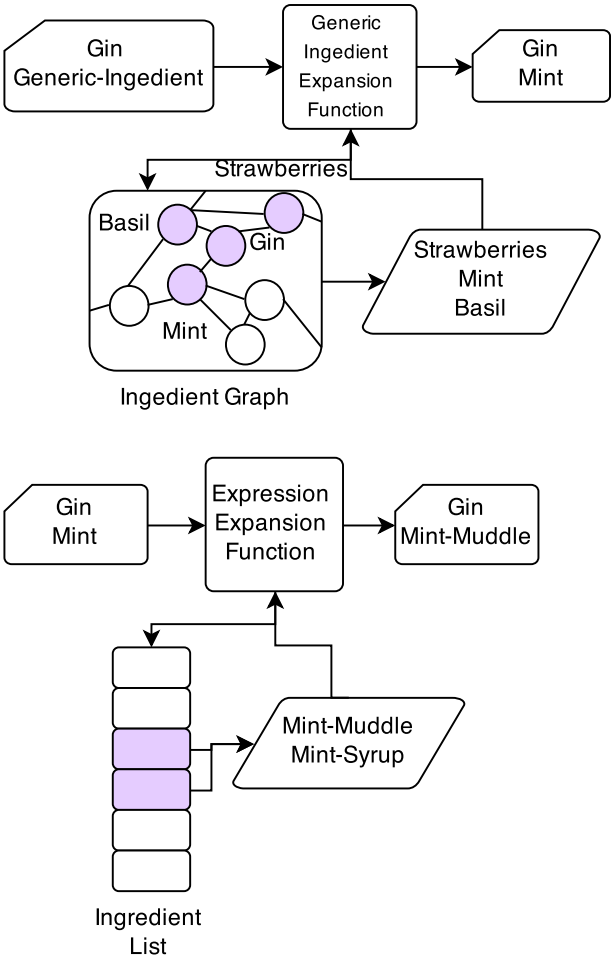



Figure 3: Overview of database expansion.

of the figure, where an expansion function, f(), takes an input grammar, the symbol to be expanded ('A') and the symbol to expand to ('B') and returns an output grammar where 'A' has been replaced with 'B'. Second, grammar symbols sometimes do care about context, and look at the other symbols in the current string before expanding. If certain symbols are present, then that symbol expands differently. This is referred to as context-sensitive expansion. This expansion is shown in the middle of Figure 2. The expansion function, g(), takes an input grammar, the symbol to be expanded ('B') and the symbol to expand to ('D'). g() scans the input string, and since the grammar contains both B and C, transforms B to D. Unlike other context sensitive grammars, the C's location in the string is unimportant—if the string contains a C, B will transform to D. Finally, some rules, instead of going from one symbol to the next, instead request an external structure to supply the next symbol. These rules can be context free or context sensitive. This expansion is shown at the bottom of Figure 2. The expansion function, h(), takes an input grammar, the symbol to be expanded ('E') but does not have a symbol to expand to. Instead, h() makes a request of an external data store as to what 'E' should expand into. The data store returns 'F', and the function replaces 'E' with 'F' and returns the grammar.

External data structures make no promises about being able to fulfill a request. When a database cannot fulfill a request, grammar expansion is restarted from the axiom. External database calls are outlined in Figure 3. The top graph in Figure 3 shows expansion using the ingredient graph. When queried, the expansion function either passes a symbol that has a node in the graph (in this case, gin) or polls the graph randomly. The node's neighbors are returned, and the function chooses one to use for expansion. The bottom graph shows expansion using the ingredient list. When used, the list is supplied a symbol that needs to be expressed (in this case, mint). The list returns all the possible expressions of the symbol, and then the expansion function chooses which one to use.

Symbols also have a flag that is set to 'false' for context-free symbols and 'true' for context-sensitive symbols. This flag is used to allow for context-free symbols to be expanded before context sensitive symbols, so that symbols that need context will have as much information as possible before expanding.

To help keep track of how a grammar is expanding, symbols are actually a (symbol, type) tuple. The symbol is what gets replaced or used for replacement, while the type helps various context sensitive rules determine the right time in the expansion to execute. Types work like walls, all symbols need to be of a particular type before the next set of rules can apply. The types used are functional, ingredient, expression, and terminal. Functional symbols are qualifiers like "strong", "sweet" or "sour". They describe the function of a particular ingredient, according to a ratio. So, a margarita can be described in rough terms as 3 parts strong : 2 parts sweet : 1 part sour. Ingredient level symbols fill in the functional symbols with high level ingredient qualifiers, so, "lemons" could fill in for "sour". The next type of symbols, expression symbols, tell us how each ingredient is going to be expressed in a cocktail. So, "lemons" could become "lemons-juice" or "lemons-muddled". Most expression symbols are also terminal symbols, however, occasionally the grammar needs to add a few more details to a symbol before it can get rendered to text.

Symbols keep track of what they replaced, which allows us to trace a symbol's lineage. This commonly happens when we divide up the ingredients into parts. If a drink calls for 2 parts sour, and both lime and lemon juice are being used, then the cocktail generation system checks that the juices both come from the same original sour symbol. It then correctly divides the parts equally among the juices.

It is also possible for a rule to rewrite a symbol's lineage, as in Figure 4. Lineage rewrites perform abstraction and recategorization within a set of rules. This increases the expressive potential of a particular category, so that it can still accurately represent the drinks that fall into that category without resorting to having a collection of starting axioms. This also allows for axiomatic change based on how a current grammar is expanding. If it suddenly makes more sense for a particular expansion of symbols to have been derived

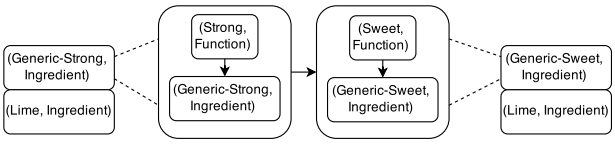



Figure 4: Overview of a lineage rewrite rule. The rule transforms a strong symbol into a sweet symbol. Normally, new symbols are appended as children to the symbol they expanded from, but for lineage rewrites, we replace the symbol and the symbols it descended from.

from a similar axiom, the axiom can shift to reflect that.

Each grammar expands until it hits a set of terminal symbols. Then, the set of symbols is passed off to the text renderer to generate a human readable recipe. Axioms for a cocktail grammar start with a set of functional symbols, and the amount of each symbol that should be in the final drink expressed in parts. An example grammar (the Old Fashioned grammar) is presented in Figure 5.

## Ingredient Representation

There are two data structures that the expansion engine can query for information to build new symbols. How both of these structures are used is outlined in Figure 3. The first structure is used primarily for expanding ingredient symbols; the second structure is used for expanding expression symbols. The first data structure is the ingredient graph, a bidirectional graph where each node corresponds to an ingredient such as chocolate or strawberries. Adjacent ingredients on the graph work well together in a drink, according to experts. So, if we already know we are going to use one ingredient, we can get that ingredient's neighbors in order to see what other ingredients should be used with it. There are some nodes that are connected to every other node, but it is, in general, a sparse graph.

The other main data structure is a list of ingredient expressions, organized by ingredient. This lets us look up how lemons can expand, and pick an expression that fits a rule. The list makes no promises about having the right entry for a rule. If a rule is looking for a puree and is trying to expand lemons, the query on the list will return nothing (as lemon puree was not considered a valid ingredient by experts).

## Expansion Engine

The expansion engine takes a cocktail grammar axiom and expands it in turn, from functional symbols to ingredient symbols, then to expression symbols, then to terminal symbols, as seen in Figure 1. The engine also makes requests of the outside data structures to get information needed to create a symbol when required. The engine has a hard rule: expand context free symbols before context sensitive symbols. In addition, when two or more context sensitive symbols can expand, and no context free symbols can expand, a random one is chosen to expand first.

The best way to go over the expansion engine is to go through a sample run. A user has asked for the system to generate something based off of a White Russian. The grammar starts with three symbols: (strong, function), (sweet, function) and (mild, function). Now, the system looks through the current rules it has and there are two that can potentially apply: perform a lineage rewrite to transform the strong symbol into a sweet symbol, or expand the strong symbol into an ingredient level base spirit. As these are both equally valid rules, the system picks one at random, and decides to transform the strong symbol into a sweet symbol. The grammar now reads like (sweet', function), (sweet, function), (mild, function). The grammar now has several context-free rules it can apply, expanding sweet' into a ingredient level sweet symbol, expand sweet into an ingredient level sweet symbol and expand mild into an ingredient level mild symbol. The grammar randomly picks among these three rules, as all of them are context-free. After those rounds of expansion, the grammar now looks like (generic-sweet, ingredient), (generic-sweet, ingredient), (generic-mild, ingredient). Now, there is a context-free rule for expanding the generic-mild symbol, whereas expanding generic-sweet is context-sensitive. So, generic-mild gets expanded next and the grammar now looks like (generic-sweet, ingredient), (generic-sweet, ingredient), (cream, expression). At this point, we start to expand one of the generic-sweet symbols and the grammar needs outside help, as there are many symbols that it can expand into. The ingredient graph is queried, looking for neighbors of the cream symbol. A list of neighbors is returned. This is stored, in case the grammar needs to expand another ingredient from the graph. One is picked from them: mint. The first generic sweet symbol is expanded, and now the grammar looks like (mint, expression), (generic-sweet, ingredient), (cream, expression). The grammar then checks the stored symbols from the last graph query and selects another one to expand the last generic-sweet symbol into. The grammar now looks like (mint, expression), (chocolate, expression), (cream, expression). The last round of expansion has the grammar query the ingredient list three times, once for each of these symbols to look for valid ways to express them. The end grammar string is (mint-creme, terminal), (chocolate-liqueur, terminal), (heavy cream, terminal).

## Text Rendering

The last part of the system is the text renderer. After expanding out the grammar, we have ingredients and the amount of each ingredient expressed in parts. This still needs to be converted to a human readable recipe. For the most part, this means replacing the dash in the symbol with a space. Some symbols are important to a cocktail, but are not given a part amount because they are used for garnishes, taste or in such small amounts it makes no sense to display them as a part. A prime example would be bitters, used in cocktails based on the Old Fashioned. In this case, amounts given in dashes are used (or other garnishing terms, like a "twist of lemon").

The mixing instructions are appended to the ingredient list. The mixing instructions come directly from the original category of drink and the ingredients used, with some simple replacement (such as ingredient names rather than functional terms) to make the recipe easy to follow. Finally, a name (currently an adjective, noun pair) is added to the cocktail. An example final recipe is presented in Figure 6

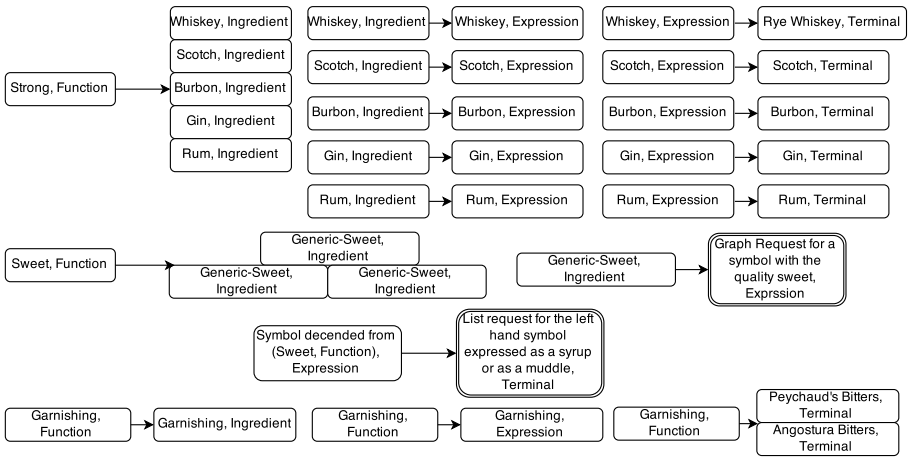



Figure 5: Diagram of the Old Fashioned grammar

brave crocodile
4 parts tequila
1.5 parts strawberry liqueur
1.5 parts passion fruit puree
1 part lemon juice
Pour all ingredients into a cocktail shaker with ice and shake for about 15 seconds.
Strain into an ice-filled highball glass.

Figure 6: A cocktail generated with the Mai Tai grammar.

## Expressivity

In both PCG and computational creativity, the expressive range of a generator is a strong consideration for how well that generator performs. Expressive range can be thought of as the range of parameters that change the kind of content the generator can produce(Smith and Whitehead 2010). For a cocktail grammar, expressive range is tied with the connectivity of the ingredient graph (the more connections the graph has, the more symbols a particular grammar can access). In addition, we can look at how 'open' a particular grammar is to various ingredient expressions. If a grammar can use a lot of ingredient expressions, then it can generate many more combinations.

To measure this, each grammar generated 1,000 cocktails, and the amount of times a particular terminal symbol occurred was counted. As we have a list of all potential terminal symbols, if a symbol was never used, it was given a use count of zero. The result of this count is shown in Figure 7. Each cocktail grammar is not equally expressive. Some categories are more restrictive than others, and lean more heavily on particular ingredients. However, each grammar seems to focus on different parts in the potential ingredient space, and when looked at all together, the entire system does a good job of making sure that all provided ingredients get used.

## Evaluation

To see if a particular generated artifact is creative, three metrics were used: quality (the measure of how well an artifact performs a particular purpose), novelty, (the measure of how unique an artifact is to an evaluator), and typicality (the measure of how well the artifact fits in a particular class of artifacts). For cocktails, quality is how well the cocktail tastes as compared to other cocktails the taster has drank. Novelty is how different a cocktail tastes as compared to other cocktails the taster has drank. Typicality is how much like a cocktail a current cocktail tastes like. This forms an evaluation space, where differing rating triples have meaning. High ratings in quality but low ratings in novelty imply that a cocktail was good, but very similar to what the taster usually orders. High in novelty and low in quality implies an interesting cocktail, but one that does not taste very good. A low score in typicality implies that the cocktail does not taste like a cocktail at all, and tastes closer to a non-alcoholic drink or straight base spirit. In order to be considered creative, a generated artifact needs to perform highly in all three categories, as per artifact-focused definitions of creativity.

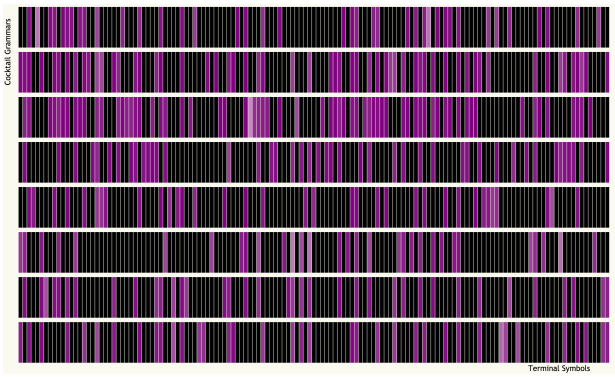



Figure 7: Ingredient use heatmap. The x-axis graphs individual ingredients, the y-axis graphs the grammars using them. As squares get lighter, they were used more times in the generated run.

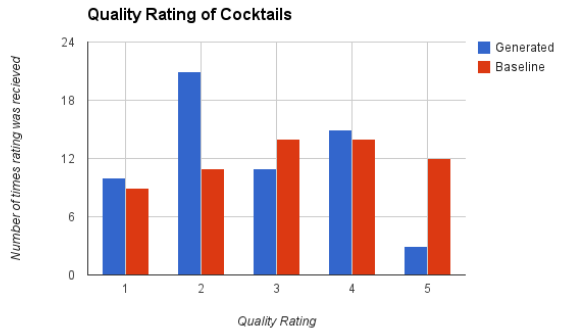



Figure 8: Quality ratings for the generated and baseline drinks. Quality in the generated drinks appears to be more polarizing than quality in the baseline drinks.

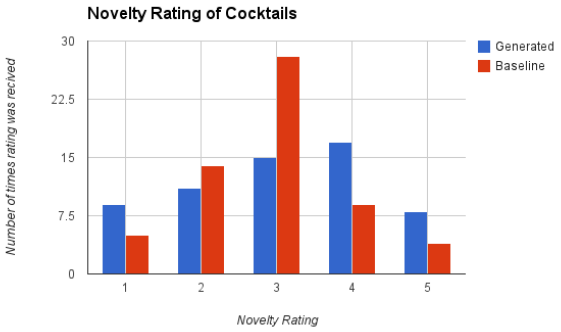



Figure 9: Novelty ratings for the generated and baseline drinks.

Table 1: P-Values

| Category | Quality P-Values | Novelty P-Values |
|---|---|---|
| Overall | 0.048 | 0.344 |
| Margaritas | 0.001 | 0.003 |
| Martinis | 0.505 | 0.118 |

Margaritas have detectable differences in both quality and novelty, martinis have no detectable differences and there is a detectable difference in quality but not novelty overall.

To this end, we had two of the eight grammars evaluated for quality, novelty and typicality by human tasters. Two cocktails from both the Margarita grammar and the Martini-based grammar were generated. In addition, two established cocktails that followed the rules for each grammar were chosen as a baseline to compare the generated cocktails against. Participants tasted each cocktail and then evaluated the cocktail based on their sip. Cocktails were presented in a random order, and participants were told that all eight cocktails were generated.

The use of a baseline to compare the generated drinks against also helped reduce taste effects—if a particular participant did not like martinis, for example, they were expected to rate both the generated martinis and the baseline martinis low.

One third of the participants had prior experience mixing drinks. All participants had at least two cocktails over the past year, with 20% having had two to four cocktails, 33% having had five to seven cocktails and the rest having had more than seven. All participants were at least 21 years old. 60% of participants were 25-29, $\approx$ 27% were 21-24, and the rest were 30 or older. 40% of participants were female, the rest were male. The tastings occurred in an office environment.

Participants were asked to evaluate the cocktails on quality and novelty using a five point scale, with a score of one being low and a score of five being high. To rate the cocktails for typicality, participants were asked if they believed what had been served was a cocktail, and to try to classify which exemplar the cocktail was based on. Table 1 contains the p-values from an unpaired t-test between the baseline and the generated cocktails. There was not a detectable difference in the novelty metric, overall. However, the generated drinks, overall, did perform slightly detectably worse in the quality metric. These results are captured in Figures 8 and 9.

For typicality, the generator performed well, with very few participants believing that they were not served a cocktail. However, when asked to try and identify which exemplar drink the cocktail had come from, participants did poorly. Participants only correctly identified the exemplar cocktail 26.67% of the time. This can imply two things: 1) that a general audience does not have enough skill in cocktails to taste where particular drinks came from and/or 2) the classification scheme used by the generator is not how the average person classifies cocktails.

## Threats to Validity

With no detectable difference in novelty between the baseline cocktails and the generated drinks, we can not conclude anything about the generated drinks compared to the baseline. In addition, the generator and evaluation did not take into account the environment the cocktail should be consumed in. It is possible that bar ambiance could impact the perception of flavor. Garnish selection is not considered in the current generator, and garnishes can strongly impact how people perceive cocktails.

There are weaknesses in any expert system—how well did the experts describe their process, and how well was that process encapsulated in the system? The majority of the cocktail generation system came from expert knowledge, from the structure of the ingredient graph to the types and numbers of grammars used. This still leaves out certain cocktails. A Cement Mixer, for example, breaks one of the cardinal rules of the system (citric acid and cream should not be mixed) to create a novel texture.

There are several weaknesses with the open loop of generating, then evaluating with human evaluators. The generator itself cannot react to the evaluations of its own output and make adjustments to its internal drink mixing philosophy. As pointed out by Stokes(2011) as well as others, this

implies that the current generator is not creative, regardless of how highly its output is scored. In addition, the generator makes no attempt to account for any sort of taste. It blindly puts ingredients together without understanding why those ingredients might work well together. This generator will also never modify either ingredient representation to discover new cocktails.

## Discussion and Future Work

Other computational ways to evaluate the system could be employed. Output recipes could be compared to existing rated recipes from online websites and databases, and a quality, novelty and typicality metrics could be derived from this comparison. However, the use of human evaluators, at least in the current state of the field, is important, because there are aspects of taste not captured in an ingredient bill or preparation steps.

Data driven cocktail generators are a strong next step. This, generally, would mean scraping various lists of cocktails (either from the web or from popular literature) and attempting to derive some heuristic for cocktails from the data. Online databases are particularly attractive, as they may have both good and bad examples to learn from.

As alluded to earlier, typicality can be a tricky part of the generator to evaluate. A way around this problem would be to have evaluators rate several well-known variations, to establish a baseline for 'cocktail recognition' that the generator's output can be compared to.

Finally, there is a need to evaluate cocktails on the merits of taste. In turn, we need a computational model of taste to see how potential drinks might taste. This lets us truly close the loop so the generator can evaluate its own work. This sort of model would allow for the use of modification or repair to a poorly evaluating dish, like several case-based reasoning techniques modify plans to best fit the current scenario. In addition, such an evaluator could evaluate many recipes, far faster than a human could.

## References


Ahn, Y.-Y.; Ahnert, S. E.; Bagrow, J. P.; and Barabási, A.-L. 2011. Flavor network and the principles of food pairing. *Scientific reports* 1.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103(1):347–356.

Bovis, N. 2015. The liquid muse. `http://theliquidmuse.com/`.

Carson, S. H.; Peterson, J. B.; and Higgins, D. M. 2005. Reliability, validity, and factor structure of the creative achievement questionnaire. *Creativity Research Journal* 17(1):37–50.

Embury, D. A. 1953. *The Fine Art of Mixing Drinks*. Faber.

English, C. 2015. Alcademics. `http://www.alcademics.com/`.

Glass, C. 2014. *Cocktail Automation Management System*. Ph.D. Dissertation, Cornell University.

Haigh, T. 2004. The mixilator. `http://www.cocktaildb.com/mixilator`.

Hammond, K. J. 1986. Chef: A model of case-based planning. In *AAAI*, 267–271.

Hinrichs, T. R. 1992. *Problem solving in open worlds: A case study in design*. Lawrence Erlbaum Hillsdale, NJ.

Jamieson, J. 2015. Liquor snob. `http://www.liquorsnob.com/`.

Joseph, P. 2012. *Boozy Brunch: The Quintessential Guide to Daytime Drinking*. Rowman & Littlefield.

Kenji López-Alt, J. K. 2013. The food lab: The science of the best chocolate chip cookies. `http://sweets.seriouseats.com/2013/12/the-food-lab-the-best-chocolate-chip-cookies.html`.

Kross, R. W.; Fiel, L. D.; Crockett, C. R.; Neely, G. B.; and Benton, L. J. 1976. Automatic mixed drink dispensing apparatus. US Patent 3,940,019.

Monsieur, LLC. 2015. monsieur. `http://monsieur.co/`.

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.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In *Workshop on Creative Systems, 4th International Conference on Case Based Reasoning*, 129–137.

Pinel, F., and Varshney, L. R. 2014. Computational creativity for culinary recipes. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, 439–442. ACM.

Pinel, F.; Varshney, L. R.; and Bhattacharjya, D. 2015. A culinary computational creativity system. In *Computational Creativity Research: Towards Creative Machines*. Springer. 327–346.

Regan, G. 2003. *The Joy of Mixology*. Random House LLC.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Simmons, M. 2011. *DIY Cocktails: A simple guide to creating your own signature drinks*. Adams Media.

Sloan, G. 2014. Robot bartenders? this new cruise ship has them. `http://www.usatoday.com/story/cruiselog/2014/11/01/quantum-robot-bar-cruise/18308319/`.

Smith, G., and Whitehead, J. 2010. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, 4. ACM.

Stokes, D. 2011. Minimally creative thought. *Metaphilosophy* 42(5):658–681.

Varshney, K. R.; Varshney, L. R.; Wang, J.; and Myers, D. 2013. Flavor pairing in medieval european cuisine: A study in cooking with dirty data. *arXiv preprint arXiv:1307.7982*.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

# Stimulating and Simulating Creativity with Dr Inventor

**[*]Diarmuid P. O'Donoghue, [*]Yalemisew Abgaz, [*]Donny Hurley, [§]Francesco Ronzano, [§]Horacio Saggion**

[*]Department of Computer Science, Maynooth University, Ireland.
[§]Universitat Pompeu Fabra, Barcelona, Spain
diarmuid.odonoghue@nuim.ie

**Abstract**

Dr Inventor is a system that is at once, a computational model of creative thinking and also a tool to ignite the creativity process among its users. Dr Inventor uncovers creative bisociations between semi-structured documents like academic papers, patent applications and psychology materials, by adopting a "big data" perspective to discover creative comparisons. The Dr Inventor system is described focusing on the transformation of this textual information into the graph-structure required by the creative cognitive model. Results are described using data from both psychological test materials and published research papers. The operation of Dr Inventor for both focused creativity and open ended creativity is also outlined.

## Introduction

This paper describes the Dr Inventor project that is both a creativity support tool while its internal operation means that is also functions as a model of creative discovery. One of the core artifacts processed by Dr Inventor to boost scientific creativity is represented by Research Objects (RO) (Belhajjame et al., 2012), which are creative academic outputs including academic publications, patent applications and related data. Dr Inventor aims to actively explore creative bisociations (Koestler, 1964) between these Research Objects using a cognitively inspired model of creative thinking. This paper adopts a big data perspective on Research Objects attempting to uncover latent creative comparisons that might lie undiscovered within its dataset. Dr Inventor directly addresses two of Honavar's (2014) facets of computationally mediated scientific discovery: firstly the development of computational representations and secondly, computationally augmenting scientific discovery.

This paper is structured as follows. We first present a case for bisociative and analogy-based creativity, addressing some issues arising from Boden's attribution of bisociative reasoning to a category called "combinatorial creativity" (Boden, 1998). We then describe the Dr Inventor model, focusing on the processes that enable it to identify analogies between its text-based inputs. Next, we outline some results from text-based sources including human psychological tests and published research papers, illustrating its operation as both a tool for focused creativity and also for open ended creativity. Finally a summary and some concluding remarks are made.

## Analogical Reasoning and Creativity

The model of bisociative reasoning developed in this paper is built primarily on a computational model of analogical reasoning, which is extended to include additional background information. While computational treatments of analogy originally focused on the analogy per se, recent attention has focused more on situated models addressing topics like Ravens Progressive Matrices (Kunda, McGreggor and Goel, 2013). The analogy process provides a unique perspective from which to view computational creativity, lying at the crossroad of research in areas including cognitive science (Gick and Holyoak, 1980), developmental psychology (Rattermann and Gentner, 1998), computer science (Ramscar and Yarlett, 2003; O'Donoghue, Bohan and Keane, 2006; O'Donoghue and Keane, 2012) and neuroscience (Green et al., 2010) . Research in these areas often constrain one another and offer the possibility of uncovering truly deep insights into the creative process. This may ultimately lead to formation of a cohesive multi-perspectival vision of one mode of creativity.

### Analogy in Creative Reasoning

Psychological evidence has highlighted people's ability to reason using analogical comparisons in the laboratory setting (Gick and Holyoak, 1980). Subjects are typically presented with two analogous stories and are required to develop the latent analogy as a key to solving a problem in one of those stories. Later in this paper we shall demonstrate Dr Inventor's ability to take the texts used in these psychology tests and develop the same analogies as observed in (many) human participants in these trials.

The use of analogy has also been described in a "real world" scenario. Blanchette and Dunbar (2001) recorded and described the use of real-world analogies during laboratory meetings of molecular biologists and immunologists. They examined 16 different meetings in a number of different laboratories. They identified over 99 analogical comparisons and scientists typically used anything from 3

to 15 analogies in a one-hour meeting. The majority of the analogies discovered were between biological and immunological information – the so called "within-domains" analogies. However, the authors noted that scientists used more "between-domains" analogies (involving semantically distant source domains such as literature or engineering), when the goal involved a creative task such as formulating an hypothesis.

Goldschmidt *et al*, (2011) and others have highlighted that "problem fixation" often frustrates peoples efforts to think creatively. That is, people experience difficulties in seeing new uses for existing information. The authors argue that to overcome this fixation and to promote creative thinking, that people be presented with semantically distant comparisons for a given problem. Research by Bowden *et al* (2005) and others has highlighted that insight occurs when problem solvers suddenly see a connection that previously eluded them. One possible mechanism of supporting insight is the discovery or a creative bisociations, like analogies and blends (Fauconnier and Turner, 1998).

## Analogy and Transformational Creativity

Margaret Boden (Boden, 1990) offers three well-known levels of creativity, with increasingly impressive impact at the levels of *improbable, exploratory* or *transformational* creativity. Boden argues that analogy is effectively the lowest form of creativity (improbable); however we argue that when analogical reasoning is seen within the context of a cohesive system of human reasoning the picture is less clear. If the inferences mandated by an analogy contradict some fundamental axiomatic belief, especially beliefs with that numbers of associated deductions and inferences, then resolving this contradiction might well involve the "*shock and amazement*" associated with Boden's highest level of transformational creativity. It appears that analogies may in fact, drive creativity at any of Boden's levels of creativity. Our creativity model is domain independent and does not include a pragmatic component or domain context. So, as our model does not use domain-specific knowledge, arguably it cannot be easily cast as one of improbable, exploratory or transformational creativity in Boden's terms.

## Creativity Producers and Consumers

Creativity is generally seen from the perspective of the creator. But, Dr Inventor needs to make a distinction between itself and its users who are consumers of its creative outputs. O'Donoghue and Keane (2012) made the point that a creative process may present a creative comparison so as to highlight the latent similarities, perhaps using terminology that highlights this commonality. However, discovering such creative comparisons will generally have to combat these differences in order to discover that commonality *ab initio*.

When they encounter a creative artefact, the interested consumer should also experience an episode of creativity, once they engage properly with the artefact. The process of engaging with a creative artefact should empower the consumer, ultimately leading them to a new conceptual space akin to that of the creator. If the artefact doesn't cause this reaction, then its creative impact is greatly lessened and may be considered less creative. So, a truly creative output is not merely a recorded by-product of the creative experience of its creator, but it must also engender creativity within those consumers that engage properly with it. To achieve this, creative artefacts must have communicative potential and arguably, multiple creative artefacts may be necessary to clarify a new conceptual space - or to convince an unwilling consumer.

We call *secondary creativity* the act of engaging with a creative artefact so as to transform ones conceptual space, with *primary creativity* being the initial creative episode. We believe that secondary creativity is also essential for truly creative artefacts, helping wide adoption of this new perspective. Dr Inventor is concerned with both finding creative bisociations and with presenting these outputs to its users. It will use both ontology and visual analytics to support this secondary creativity.

## Dr Inventor

Dr Inventor is a computationally creativity system that can both model scientific creativity and can also use its outputs to stimulate creative thinking within its users. It is as concerned with the process of creativity as it is with the products that arise from these processes (Stojanov and Indurkhya, 2012). Dr Inventor is built on a cognitively inspired model of human bisociative reasoning, based on analogical comparisons and the counterpart projection of conceptual blends (Fauconnier and Turner, 1998; Veale, O'Donoghue and Keane, 2000). CrossBee (Jursic et al., 2012) looked at exploring scientific papers, its focus lay in finding bridging terms between them. The focus of Dr Inventor is on finding and extending systemic similarities for creative purposes.

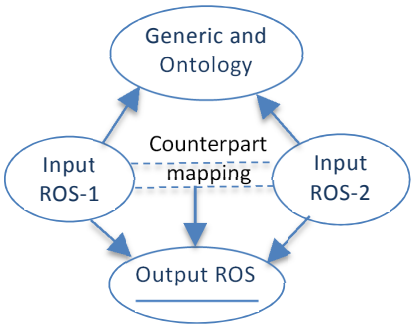



**Figure 1:** Conceptual spaces used by the bisociative model of Dr Inventor including the analogically founded mapping between the inputs

This paper focuses primarily on three of the four spaces of conceptual blending, namely the two input descriptions and the generic space. The dotted lines in Figure 1 indicate the correspondences between these inputs, derived with the help of Gentner's structure mapping theory (1983). Dr Inventor's 3-space model identifies a generic space contain-

ing the ontological similarity between paired relations from the Input 1 and Input 2. Dr Inventor thus identifies the generic space corresponding to the aligned items from the bisociation. This generic space also enables Dr Inventor to monitor the semantic congruity within a bisociation, to uncover comparisons more in fitting with the users' needs. Finally, the output space represents the new interpretation of one of those inputs. As each "target" maybe re-interpreted by multiple sources, and because that target may also act as a source for some other Research Object Skeleton (ROS), each newly created ROS is stored separately. For simplicity, this paper generally uses the terms *source* and *target*, unless specific point about the Blend is being made.

This means that a new ROS may act to later inspire subsequent creativity. Thus, Dr Inventor can potentially operate as a "Self-sustaining" creativity model as of described in O'Donoghue et al (2014). One of the chief obstacles hindering Dr Inventor in achieving this self-sustaining creativity lies in the quality of the new ROS and a sufficiently diverse knowledge base from which to progress.

The core data artefacts used by Dr Inventor are Research Objects (Belhajjame et al., 2012), which are research outputs including publications, patents, data, software (O'Donoghue et al., 2014b), social network information and other resources. Dealing with such heterogeneous data sources, characterized by consistent amounts of information to integrate and process, big data approaches and technologies are essential in order to enable the computational approaches to creativity in Dr Inventor. This paper focuses on the textual contents of RO, particularly of publications and patents. These documents are first subject to a number of processing activities to properly mine their contents in order to generate inputs that are useful to Dr Inventor's analogy-based model.

From each RO Dr Inventor generates a graph-based representation called the Research Object Skeleton (ROS) representing the key concepts and relationships extracted from that RO. Dr Inventor identifies similarities between these ROS with a view to extending these similarities and uncovering creative possibilities.

## Dr Inventor Model

The overall Dr Inventor model contains components that deal with document summarization, information extraction; ontology learning, matching and personal recommendation; ROS generation, assessment, similarity and analogy/blending; validation, mapping, retrieval and finally visual analytics. The discussion in this paper will focus on the ROS generation, analogy/blending model and the creativity assessment components.

### Mining Textual Contents to Populate ROS

In Dr Inventor, Research Object Skeletons (ROSs) are built by mining the contents aggregated by the corresponding Research Objects (ROs). To populate a ROS, Dr Inventor mainly relies on the extraction of information from the textual contents of a RO. To analyze these contents, Dr Inventor integrates a Natural Language Processing Pipeline (DRI-NLP pipeline) that aggregates and customizes several Information Extraction (Piskorski and Yangarber, 2013) and Text Summarization (Saggion, 2014) approaches and tools.

Since scientific publications constitute one of the main kinds of textual documents included in a RO, DRI-NLP pipeline has been properly structured to support the analysis of research papers. The great majority of papers are currently available as PDF files. As a consequence, the conversion of PDF into plain text constitutes an essential prerequisite to properly perform any further text analysis. To this purpose, DRI-NLP pipeline relies on PDFX (Constantin, Pettifer and Voronkov, 2013) that converts a PDF document of a scientific publication to a semi-structured text (XML). The plain text output of PDFX is thus processed so as to identify sentences by means of a custom rule-based sentence splitter. Each sentence is processed by means of the MATE dependency parser (Bohnet, 2010) to extract dependency relations which are represented in a dependency tree. DRI-NLP pipeline dependency parser has been customized in order to properly deal with several peculiarities of scientific publications, including the presence of inline citations. In particular, inline citation markers like "(AuthorA *et al*.)" or "(1)" are excluded from the dependency tree if they have no syntactic functions in the sentence where they are present. Dr Inventor is focused on the discipline of computer graphics as its test-bed, thus a particular challenge has been dealing with the many mathematical expressions in these papers and allowing their treatment separately from the main body of the text. Besides dependency parsing, DRI-NLP pipeline enables the creation extractive summaries of papers by ranking their sentences by relevance (Saggion, 2014).

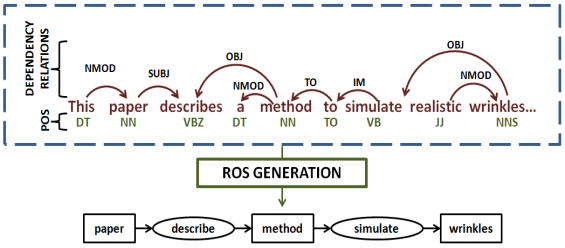



**Figure 2:** Processing PDF papers by Dr Inventor Natural Language Processing Pipeline

As result of dependency parsing, each word of a sentence is characterized by its Part-Of-Speech (POS) (noun, verb, adjective, etc.) and dependency relations (subject, object, verb chain, modifier of nominal, etc.). The linguistic information extracted from each publication can be condensed in the tables: the Syntactic dependency and the POS tag table. In particular, Figure 2 focuses on the analysis of a specific sentence taken from the abstract of a paper.

While Dr Inventor is focused on the test-bed of computer graphics publications, it remains a general model capable of dealing with arbitrary text inputs. This paper also uses data derived from psychology text materials and work is ongoing using the texts of patent applications.

**ROS Generation**

The next task for Dr Inventor is to generate a ROS from the results of the parsing process. The representation we chose for these graphs is sufficiently general to represent different types of RO. Since we want to structure objects and their inter-relationships this information is stored as a graph, aimed at supporting the later structure mapping process (Gentner, 1983).

| **Karla the Hawk:** |
|---|
| *Karla, an old hawk, lived at the top of a tall oak tree. One afternoon, she saw a hunter on the ground with a bow and some crude arrows that had no feathers. The hunter took aim and shot at the hawk but missed. Karla knew the hunter wanted her feathers so she glided down to the hunter and offered to give him a few. The hunter was so grateful that he pledged never to shoot at a hawk again. He went off and shot a deer instead.* |
| **Zerdia True Analogy:** |
| *Once there was a small country called Zerdia that learned to make the world's smartest computer. One day Zerdia was attacked by its warlike neighbor, Gagrach. But the missiles were badly aimed and the attack failed. The Zerdian government realized that Gagrach wanted Zerdian computers so it offered to sell some of its computers to the country. The government of Gagrach was very pleased. It promised never to attack Zerdia again* |

**Table 1:** Textual contents of "Karla" and "Zerdia"

Each ROS is constructed as an *attributed relational graph* (ARG), which is a directed graph where nodes and edges may contain additional properties like labels, categories and numeric values. If required, we can store additional identifying information (e.g. Author, Affiliation, etc.) within the graph, but this information is not required for the analogy process *per se*.

The primary information in a ROS is the concept nodes (nouns) and the relationships (verbs) between them. Concept nodes are not linked directly to one another but are connected with relation nodes. To generate the ROS we use the general structure "subject" – "verb" – object" - as required by SMT. These triples arise from the dependency and POS tables as the input to ROS generation. Early testing has shown that taking triples directly from the dependency table typically leaves many of them incomplete, leaving ROS without the necessary structure to support identification of creative inter-domain correspondences. Therefore, Dr Inventor performs a deeper exploration of the tables in order to generate more useable ROS structures.

By constructing a dependency graph from the tables and applying a set of heuristics to the graphs, a more complete set of triples is generated. The heuristics involve combining some of the nodes and tracing through the graph finding pairs for each verb.

Figure 3 depicts two ROSs generated for the "Zerdia" and "Karla" stories (Table 1) used in human psychological studies (Gentner and Landers, 1985). They were generated by the text mining and ROS generation techniques discussed earlier, but some manual post-editing was performed to identify co-referencing concepts nodes in the ROS. In the "Zerdia" story the word "it" is used twice, but the ROS were edited so one instance was replaced by the referent "Zerdia" and another by "Gagrach". In the "Karla" story the word "she" refers to "Karla" and "he/him" refers to "hunter". While these co-referents were resolved manually work is underway in the text pipeline to automatically resolve these referents.

Dr Inventor explicitly represents higher-order (causal relations connect first-order relations) relations within a ROS. A distinct set of nodes represents the higher-order relations, these connecting the first-order (and potentially other high-order) relations. However, ROS generated from within our Research Objects corpus show that high-order (causal) relations are rarely explicitly identified. As we shall see, this influenced our choice of mapping algorithm.

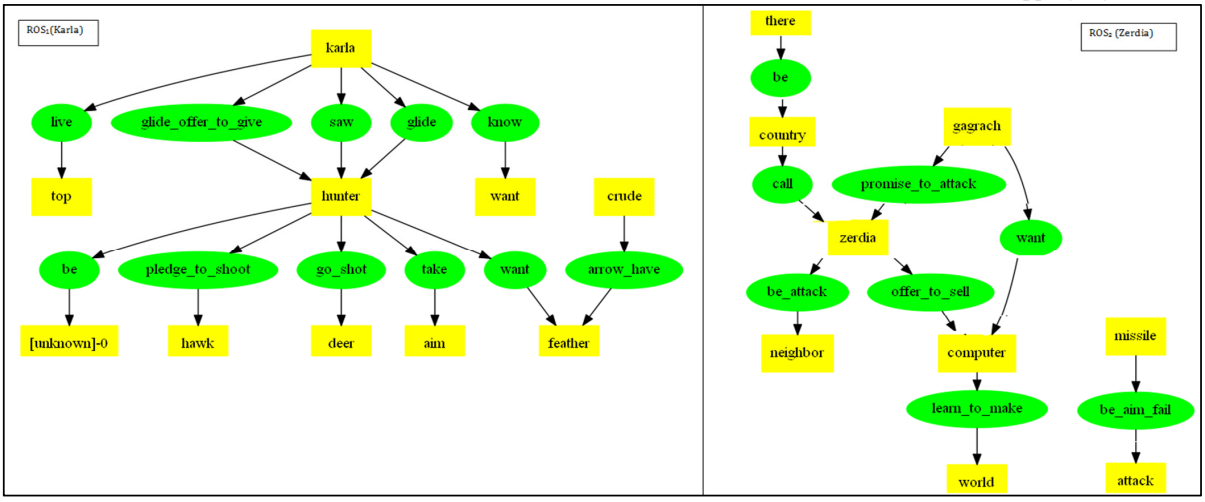



**Figure 3:** ROS for the "Karla" and "Zerdia" analogy used in human studies and Dr Inventor

**Graph Storage**
Dr Inventor uses the Neo4j graph database to store its ROSs. Neo4j has as its core structures; nodes, relationships between them and properties on both, this being the same structure as the ARG. Additional information such as the SentenceID or SectionTitle for each triple can also be stored in the Neo4j database. This can be useful when we want to map only between particular sections (e.g. Abstract or Conclusion) and also to reference back to the original sentences from which the triples were extracted.

**Data Differences**
Previous analogy models like SME (Forbus, Ferguson and Gentner, 1994), IAM (Keane and Bradshaw, 1988) and Kilaza (O'Donoghue and Keane, 2012) used hand coded data. The ROS generated above differs from the earlier hand-coded data in at least two significant respects. First ROS contain very few high-order relations, which are heavily used by mapping models mentioned above. Dr Inventor does not focus on the hierarchical structure of hand-coded data, using instead some lower level topological structure. Secondly (as mentioned in (O'Donoghue and Keane, 2012)) hand coded data often simplifies the comparison process by using relations that highlight the latent similarity. Dr Inventor must uncover and identify the hidden similarity even in the absence of such lexical cues.

## Dr Inventors Creativity Engine

This paper focuses on the creativity engine that lies at the heart of Dr Inventor. Thus, we focus on creative analogy-based comparisons and show a number of features of Dr Inventor that specifically attempt to support the identification and generation of these creative analogies.

### Creative Analogies

A number of properties appear to be shared amongst many creative analogical comparisons (O'Donoghue and Keane, 2012) and these facets are used to generate novel and potentially useful analogies and blends. Firstly the source (domain) of inspiration is typically semantically different from the given target problem. That is, creative sources tend to be sufficiently different and any similarity is non-obvious and has not been previously explored in detail. Secondly, the creative source contains the necessary structural similarity that is required to generations of viable analogy with the given problem.

To this end, Dr. Inventor specifically seeks out bisociations that involve two semantically distant domains, that form a rich inter-domain mapping and that yield inferences suggesting something new about one of those domains.

### Graph Mapping

To support creative analogies Dr Inventor's retrieval and mapping activities makes frequent use of topological features derived from each ROS. For analogical mapping we exploit features such as type of the nodes (verb, noun), types of relationships (subject, object), degree (in-degree, out-degree) and node rank values calculated by Node Rank algorithm (Bhattacharya et al., 2012).

We initiate the mapping process by calculating the Node Rank and by sorting the nodes in a descending order. The ranking allows us to start the graph matching process from the most centrally connected and useful node. This will further be used to serve as a threshold to screen useful nodes to improve the performance of the mapping process. The results presented in this paper have been generated using smaller RO (such as the abstracts of graphics paper RO), so performance has not been an issue. However this situation will change when mapping between ROS with large number of nodes is required.

The relation (verb) nodes in each ROS are represented distinctly, with one instance of a relation node for each verb contained in the RO. Verb nodes are central to the process of representing the content of the RO, however their connectedness is limited to a degree of 2 and thereby affects the resulting Node Rank values. However, multiple references to the same concept (noun) node will appear in the ROS as a single concept node – but referenced multiple times by each distinct relation node. Thus concept nodes have the greatest direct impact on the Node Rank values as a single concept node may be linked through many relations within a ROS. The mapping process avails of this referential structure when generating the largest graph-mapping between two ROS.

To identify a pair of mapping nodes from the source to the target, we used structural similarity score (using the connectedness of the nodes) and the literal similarity score. Using structural similarity, we consider two nodes as candidate mapping nodes if they have a higher similarity score. Whereas, literal similarity calculates the similarity coefficient between two words and yields a value between 0 and 1, where 0 indicates no similarity and 1 indicates complete similarity (synonym). This is achieved by using the Wu&Palmer (Wu and Palmer, 1994) WordNet-based similarity metric.

The mapping algorithm firstly selects a pair of nodes *P(sNode, tNode)* from the source and the target respectively, with the highest node ranked nodes being selected first. In this way, the algorithm focuses on highly connected nodes within the graph because they contribute most to the mapping and analogical inference activities. Secondly, the mapping process checks if the selected pair *P(sNode, tNode)* is structurally feasible for analogical mapping. A structurally feasible pair contains a source node which has degree (in degree and out degree) greater than or equal to degree (in degree and the out degree) of the target node respectively. The comparison ensures the identification of a sub-graph or an isomorphic graph of the target graph in the source graph. It further assesses the semantic similarity of the two nodes using Wu& Palmer. Next, mapping adds *P(sNode, tNode)* to the inter-domain map to incrementally build a mapping sub-graph, if *P(sNode, tNode)* is feasible. The mapping stores the pair of mapping nodes along with their similarity scores.

The mapping process then generates new candidate mappings by expanding *sNode* and *tNode* of *P(sNode, tNode)* to their respective connected nodes that are not already expanded. By following the "subject" or "object" relationship path it reaches the connected nodes of the graph, incrementally adding these to the inter-domain mapping. After the candidate pairs are generated, they are ranked using their semantic similarity score. Ranking the candidate pairs will give us a chance to expand pairs with the highest semantic similarity first.

After including all mappings arising from the initial root mapping, the process then resumes with the next highest ranked and unmapped predicates. The algorithm employs depth first search to expand the nodes in the graph to identify new mapping pairs. Finally, it selects the mapping that contains the largest sub-graph and returns the mapping nodes together with their semantic similarity score.

We now look at the results produced by generating a mapping between the "Karla" and "Zerdia" psychology materials listed above, with the corresponding ROS being depicted in Figure 3. We note that this simulation of human analogy process began with the same text materials that were presented to human subjects. This comparison is an example to focused creativity, where both the source and target have been pre-identified.

The mapping between "Karla" and "Zerdia" gives us 11 mapped nodes between the source and target (Table 2). For example the noun node "Karla" maps to "Zerdia", "Feather" maps to "Computer" and "Hunter" maps to "Gagrach". Such a mapping identifies analogous items between the source and the target and is crucial for transferring new knowledge form the source to the target. In this specific example 50% of the nodes in the target ROS are mapped to the source ROS with an average Wu&Palmer similarity score of 0.56. The original domains can often include information that does not participate in the mapping, such as the (*missile be-take-aim attack*) in the Zerdia story. However the absence of this relation from the mapping is not terribly significant as it is an isolated fragment of information and does not contribute largely to the main story – that contributes to the largest connected component of that ROS.

| **Mapping Nouns** | | **Mapping Verbs** | |
|---|---|---|---|
| **Source** | **Target** | **Source** | **Target** |
| Hunter | Gagrach | Want | Want |
| Crude | World | Live | Offer_to_sell |
| Feather | Computer | Arrow_have | Learn_to_make |
| Want | Country | Glide_offer_to_give | Be_attack |
| Karla | Zerdia | Glide | Promise_to_attack |
| | | Know | Call |

**Table 2:** Mapping between "Karla" and "Zerdia".

**Analogies within Graphics Collections**

To examine the mapping process, we used 10 papers from computer graphics domain. The abstracts of the papers were extracted and were processed using the previously described steps. Each ROS were mapped to the other 10 ROS including itself. The most basic step is to compare a ROS against itself. For all the 10 papers Dr Inventor yields the highest number of mapped nodes when a ROS is compared with itself – with all or almost all nodes being successfully mapped. This could be considered as a very basic step toward the evaluation of the mapping component of Dr Inventor.

The mapping of a ROS against the remaining 9 ROS identifies pairs that have the highest mapping nodes and pairs that have the lowest mapping nodes. The most analogous papers are those with large number of mapped nodes and highest similarity score. For example, the most similar non-identical mapping among the 10 papers is between "*Bar-Net_Driven Skinning for Character Animation*" and "*Real Time Large Deformation Character Skinning in Hardware*" with 14 mapping nodes and an average Wu&Plamer similarity score of 0.36. While the semantic similarity score may appear quite low, this was achieved from within a small collection of papers. We conducted a quick manual comparison between the abstracts of these papers and initial results indicate that these papers can be considered somewhat analogous to one another as for example, both papers present different approaches to the computer graphic topics of "skinning". This analogy arose from the desire to identifying the largest mapping with the strongest semantic similarity from within the 10 papers however, the next section will discuss a more creative Use Case scenario.

The lowest mapping occurs between the papers "*Curve Skeleton Skinning for Human and Creature Character*" and "*Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation*" with 5 mapping nodes and average similarity of 0.35. The mapping process, as it is expected, is not symmetrical, i.e. mapping between *(S, T)* may not be the same as mapping between *(T, S)*. For example, saying "*a man is like a pig*" is not the same as saying that "*a pig is like a man*"! However, in this specific data set, it does not significantly affect the highest mapping node pairs.

**Analogical Inference and Pattern Completion**

Once we find a mapping between the two ROS, the next phase is to generate the resulting inferences by applying a "pattern completion" process to that mapping. This adds the newly inferred information to the target ROS to produce the new interpretation of that concept. In the exploratory analogy mapping process, the user may be interested to explore all the candidate nodes once he/she knows the existence of analogy between the source and the target.

**Creativity Support and Evaluation in Dr Inventor**

Dr Inventor is focused firstly on operating as a Creativity Support Tool (CST) and secondly, as a simulation of the analogy process. Shneiderman (2007) noted that there are no obvious metrics to quantify for CST's and this problem lies at the core of creativity assessment and evaluation. The following two approaches are useful to evaluate the level of creativity support provided by Dr Inventor.

Among the functionality being developed for Dr Inventor is an "Inspire Me" Use Case, enabling users to creatively re-interpret one of their own papers. This will be achieved by using the paper as a target and searching the archive for papers that can act as a creative source domain, forming a large and semantically well-balanced mapping by making use of the topological structure of each ROS. Dr Inventor will identify and present to the user those analogies offering a collection of novel inferences that highlight the potential benefits of adopting this creative analogical comparison. Internal metrics will serve to select the most promising analogies to present to users, assessing the structural and semantic foundations of the comparison.

Implicit feedback on the presented analogies will be gathered by the user interface, enabling comparative evaluation of different comparisons by monitoring user engagement. Explicit user feedback also plays a very important role in evaluating Dr Inventor, using experts in computer graphics for evaluation. The Creative Support Index (CSI) (Cherry and Latulipe, 2014) is a psychometric survey that will serve to assess the creativity support provided by Dr Inventor. It is quick and easy to administer and is composed of two sections; a rating scale section and a paired-factor comparison section. It identifies 6 major factors of creativity, namely: enjoyment, exploration, expressiveness, immersion, results worth effort and collaboration. Under each of the factors, CSI asks two questions that are rated between 0 and 10, where 0 indicates the lowest value and 10 indicate the highest achievement. The paired-factor comparison section consists of each factor paired against every other factor for a total of 15 comparisons. As Dr Inventor will support users with different levels of expertise (first year PhD students to experienced Professors), this factor in particular will have to be controlled monitored during the evaluation process.

The Creative Achievement Questionnaire (CAQ) (Carson, Peterson and Higgins, 2005) is a very broad and general creativity assessment technique. Within the context of Dr Inventor, achievements appear to be primarily assessed by qualifying the number of published scientific papers. But the CAQ provides poor coverage of lower levels of creative achievement (before publication) that could guide development of the Dr Inventor project. However, the CAQ might be useful for the final evaluation of Dr Inventor.

We also note that (Jordanous, 2014) identifies five criteria to support meta-evaluation of computational creativity *per se*, as opposed to our current focus on Dr Inventor as a creativity support tool.

## Summary and Conclusions

Dr Inventor is a computationally creative model that acts as both a simulation of human creative reasoning and also as a creativity support tool. We described how Dr Inventor performs text extraction from research publications presented in pdf format, describing how it addresses many complications that result from use of the pdf format. The dependency parser was described, as was the process of constructing the graph representation used by the core model. Some peculiarities of the resulting graphs were noted, particularly the extreme rarity of identifiable higher-order causal relations. Some implications were noted the process of identifying the inter-domain correspondence. The text used from human psychological trials showed the ability of Dr Inventor to generate comparison using these same textual materials. Results for other Research Objects were outlined.

The mapping and evaluation process uses ontological information as a preference criterion to choose comparisons with the greatest potential for creativity on Dr Inventor users. Ontology also opens the way to re-describe the original documents, highlight the identified similarities. While this work is ongoing, it opens the way for early evaluation of Dr Inventor by comparing the impact upon users of creatively re-interpreted documents.

## Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme ([FP7/2007-2013]) under grant agreement no 611383.

## References

Belhajjame, K., Corcho, O., Garijo, D., Zhao, J., Missier, P., Newman, D., Bechhofer, S., García, E., Gómez-Pérez, J.M., Palma, R., Soiland-Reyes, S., Verdes-Montenegro, L., Roure, D.D. and Goble, C. (2012) 'Workflow-centric research objects: First class citizens in scholarly discourse', 9th Extended Semantic Web Conference, Hersonissos, Greece.

Bhattacharya, P., Iliofotou, M., Neamtiu, I. and Faloutsos, M. (2012) 'Graph-based Analysis and Prediction for Software Evolution', in *Proceedings of the 34th International Conference on Software Engineering*, Piscataway, NJ, USA: IEEE Press.

Blanchette, I. and Dunbar, K. (2001) 'Analogy use in naturalistic settings: The influence of audience, emotion, and goals', *Memory & Cognition*, vol. 29, no. 5, pp. 730-735.

Boden, M. (1990) *The Creative Mind*, Weidenfeld and Nicolson.

Boden, M. (1998) 'Computer Models of Creativity', in Sternberg, R.J. *Handbook of Creativity*, Cambridge University Press.

Bohnet, B. (2010) 'Very high accuracy and fast dependency parsing is not a contradiction.', Proc. 23rd International Conference on Computational Linguistics, 89–97.

Bowden, E.M., Jung-Beeman, M., Fleck, J. and Kounios, J. (2005) 'New approaches to demystifying insight', *Trends in cognitive sciences*, vol. 9, no. 7, pp. 322-328.

Brown, T.L. (2003) *Making Truth: Metaphor in Science*, University of Illinois Press.

Carson, S., Peterson, J.B. and Higgins, D.M. (2005) 'Reliability, Validity, and Factor Structure of the Creative

Achievement Questionnaire', *Creativity Research Journal*, vol. 17, no. 1, pp. 37–50.
Cherry, E. and Latulipe, C. (2014) 'Quantifying the Creativity Support of Digital Tools through the Creativity Support Index', *ACM Transactions on Computer-Human Interaction*, vol. 21, no. 4.
Constantin, A., Pettifer, S. and Voronkov, A. (2013) 'PDFX: fully-automated PDF-to-XML conversion of scientific literature', Proceedings of the 2013 ACM symposium on Document engineering, 177–180.
Fauconnier, G. and Turner, M. (1998) 'Conceptual integration networks', *Cognitive science*, vol. 22, no. 2, pp. 133--187.
Forbus, K.D., Ferguson, R.W. and Gentner, D. (1994) 'Incremental structure-mapping', Proc 16th Conference of the Cognitive Science Society, 313–318.
Gentner, D. (1983) 'Structure-mapping: A theoretical framework for analogy', *Cognitive Science*, vol. 7, pp. 155-170.
Gentner, D. and Landers, R. (1985) 'Analogical reminding: A good match is hard to find', Proceedings of the International Conference on Systems, Man, and Cybernetics., Tucson, AZ.
Gick, M.L. and Holyoak, K.J. (1980) 'Analogical problem solving', *Cognitive psychology*, vol. 12, no. 3, pp. 306-355.
Goldschmidt, G. (2011) 'Avoiding Design Fixation: Transformation and Abstraction in Mapping from Source to Target', *The Journal of Creative Behavior*, vol. 45, no. 2, pp. 92-100.
Green, A.E., Kraemer, D., Fugelsang, J.A., Gray, J.R. and Dunbar, K.N. (2010) 'Connecting long distance: semantic distance in analogical reasoning modulates frontopolar cortex activity', *Cerebral Cortex*, vol. 20, no. 1, pp. 70--76.
Honavar, V.G. (2014) 'The Promise and Potential of Big Data: A Case for Discovery Informatics', *Review of Policy Research*, vol. 31, no. 4, pp. 326-330.
Jordanous, A. (2014) 'Stepping Back to Progress Forwards: Setting Standards for Meta-Evaluation of Computational Creativity,', Proc ICCC, 129-136.
Jursic, M., Cestnik, B., Urbancic, T. and Larvac, N. (2012) 'Cross-domain Literature Mining', Proc ICCC, pp 33-40.
Keane, M.T. and Bradshaw, M. (1988) 'The Incremental Analogical machine', in Sleeman, D. (ed.) *3rd European Working Session on Machine Learning*, Kaufmann, CA, USA.
Keane, M.T., Ledgeway, T. and Duff, S. (1994) 'Constraints on Analogical Mapping: A Comparison of Three Models', *Cognitive Science*, vol. 18, no. 3, pp. 387-438.
Koestler, A. (1964) *The Act of Creation*, Penguin Books.
Kuhn, T.S. (1962) *The structure of scientific revolutions*, University of Chicago press.
Kunda, M., McGreggor, k. and Goel, A.k. (2013) 'A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations', *Cognitive Systems Research*, vol. 22-23, pp. 47-66.
O'Donoghue, D.P., Bohan, A. and Keane, M.T. (2006) 'Seeing Things: Inventive Reasoning with Geometric Analogies and Topographic Maps', *New Generation Computing, 24, (3) (special issue on Computational Creativity)*, pp. 267-288.
O'Donoghue, D.P. and Keane, M.T. (2012) 'A Creative Analogy Machine: Results and Challenges', 4th International Conference on Computational Creativity (ICCC), UCD, Dublin, Ireland, 17-24.
O'Donoghue, D.P., Monahan, R., Grijincu, D., Pitu, M., Halim, F., Rahman, F., Abgaz, Y. and Hurley, D. (2014b) 'Creating Formal Specifications with Analogical Reasoning', PICS-Publication Series of the Institute of Cognitive Science.
O'Donoghue, D.P., Power, J., O'Briain, S., Dong, F., Mooney, A., Hurley, D., Abgaz, Y. and Markham, C. (2014) 'Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes', nternational Conference on Computational Creativity (ICCC), Ljubljana, Slovenia.
Piskorski, J. and Yangarber, R. (2013) 'Information extraction: Past, present and future', *Multi-source, multilingual information extraction and summarization*, pp. 23-49.
Ramscar, M. and Yarlett, D. (2003) 'Semantic grounding in models of analogy: an environmental approach', *Cognitive Science 27*, pp. 41-71.
Rattermann, M.J. and Gentner, D. (1998) 'More evidence for a relational shift in the development of analogy: Children's performance on a causal-mapping task', *Cognitive Development*, vol. 13, no. 4, pp. 453-478.
Saggion, H. (2014) 'Creating Summarization Systems with SUMMA.', Proceedings of Language Rescources and Evaluation Conference.
Shneiderman, B. (2007) 'Creativity support tools: Accelerating discovery and innovation', *Communications of the ACM*, vol. 50, no. 12, pp. 20-32.
Silvia, P.J., Wigert, B., Reiter-Palmon, R. and Kaufman, J.C. (2012) ' Assessing Creativity with Self-Report Scales: A Review and Empirical Evaluation.', *Psychology of Aesthetics, Creativity, and the Arts*, vol. 6, no. 1, pp. 19-34.
Stojanov, S. and Indurkhya, B. (2012) 'Perceptual Similarity and Analogy in Creativity and Cognitive Development.', SAMAI Workshop at ECAI, Montpelier France, 19-24.
Veale, T., O'Donoghue, D.P. and Keane, M.T. (2000) 'Computation and Blending', *Cognitive Linguistics 11 (3/4)*, pp. 253-281.
Wu, Z. and Palmer, M. (1994) 'Verbs Semantics and Lexical Selection', in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, Las Cruces, New Mexico: Association for Computational Linguistics.

# Casual Creators

**Kate Compton and Michael Mateas**
Expressive Intelligence Studio
University of California, Santa Cruz
kcompton/mmateas@soe.ucsc.edu

**Abstract**

Many creativity tools exist to support task-focused creativity, but in recent years we have seen a flourishing of autotelic creativity tools, which privilege the enjoyable experience of explorative creativity over task-completion. Because these tools are much smaller in scope, less commercially significant, and less "serious" than their larger siblings, they have been overlooked in academic research. This paper coins the term "Casual Creators" for these tools, and provide a definition to identify tools that belong to this category. We also identify the particular design considerations that arise from autotelic creativity, and propose a number of strong design patterns that serve those considerations, patterns which are demonstrated by case studies of software built with those patterns. We believe that once this field is identified and named, the currently-isolated practitioners who make these casual creators will be able to share knowledge, like these design patterns, and develop a community of practice.

## Introduction

### An alternative design space

Often when we talk about tools to support creativity, the 'creators' exhibiting creativity are task-oriented professionals or amateurs, who have a specific problem to solve, or design task to accomplish. There exist many complex, powerful, and frequently expensive tools for different kinds of creative tasks: Maya for 3D modeling, FinalCutPro for editing video, Ableton for music production, Photoshop for images. These professional tools must support a broad range of possible actions with a focus on efficient task completion, as their users are typically being paid to complete predefined design tasks. Design, as an activity, is "goal-oriented", "intentional, purposeful" (Gero 1994). "Task" is an appropriate, and common, term for the primary action of using these tools, because the goal is to enable productive labor from the user.

Is this the only way to use creativity tools? Is productivity the only goal of creativity?

We propose the category of **Casual Creators** as an alternate design space for tools which support creativity as an intrinsically pleasurable activity, rather than as an extrinsically-motivated way to accomplish tasks.

Creativity is often an autotelic activity: we paint, draw, sculpt, sew, write and make music creatively, often with complete disregard for the quality of the final product (much less concern for task productivity), because the activity itself is so enjoyable.

This autotelic, intrinsically-rewarded form of creativity is psychologically quite distinct from creativity exhibited in a environment with extrinsic motivation (Amabile, Goldfarb, and Brackfield 1990). We should expect that support tools for autotelic creativity will be correspondingly different. There is a thriving ecosystem of appropriately-designed software tools to support task-focused creativity, so why isn't there a corresponding set of software to support autotelic creativity?

One reason is economic: the labor of the creative person is of commercial value to either their employer or their client, so there is a market incentive to maximize that output with effectively-designed tools, either to be purchased by the creative person themselves (for an independent contractor) or by their employer.

The other reason is the perceived "seriousness" of output. This division between serious and frivolous output mirrors the division in creativity research, between psychological ("P") creativity and historical ("H") creativity (Boden 2009). Psychological creativity is the 'everyday' (Kaufman and Beghetto 2009) creativity of average people, while historical creativity is the 'eminent' creative ability ascribed to famous world-changing creators and their innovative creations. Historical creativity is more valued socially and economically, so when building a tool to support creativity, tool designers like to imagine that it will be used by the next great artist or genius inventor (Shneiderman 2003), or at least used to make some famous or commercially successful product. We believe, however, that it is also important to build tools supporting "everyday" creators in enjoying pleasant and fulfilling creative exercise, even if they never produce world-changing output.

### Existing creators

Despite these reasons for not receiving "serious" attention, there are many small applications that do exist to support casual creativity. The recently-developed app marketplaces for mobile have provided a perfect haven for this sort of creativity app. These apps each support creating only a single kind

of artifact, such as abstract generative pictures (Secretan et al. 2011), virtual pottery ("Let's Create! Pottery" 2014), or 3d printable bracelets (System 2015). These apps create artifacts from a greatly-reduced possibility space, compared to the previously mentioned general-purpose professional tools. The narrowness of the possibility space allows the tool to provide greater support for the user, eliminating potential bad artifacts and speeding the process of creating good ones, at the expense of flexibility and versatility. This loss is acceptable, however, as these products aren't being created in response to the exacting demands of a boss or client, but rather because they are intrinsically *fun to make*. In fact, the end product may be discarded entirely after completion! (Nilsson 2003).

Other examples of autotelic creativity tools have come from games, a field that has always valued pleasurable user experience over productivity. Not every game contains creative activities, but many games do feature the creation (and curation) of a house, creature, or avatar, and these creativity tools can end up being more fun than the eventual gameplay.

And yet, though these tools exist, there is no central community in which their designers can communicate shared knowledge. There is no set of best practices that can be referenced by those who are attempting to make such a tool, or even a name for them so that the tool-maker can describe what they are making. The World of Warcraft (Blizzard 2004) character-creator and the "Let's Create! Pottery" virtual potter's wheel (to choose two out of many, many existing tools) may not seem to have much in common at first glance, or even share the same marketing category. We hope that by identifying this software genre, including lessons and design patterns learned from existing examples, we can define a distinct area for future research and tool implementation.

## Introducing *Casual Creators*

Casual creators can be distinguished from other creativity support tools by their goal of supporting autotelic creativity, not task-focused creativity. From this initial difference in goals arises a variety of other differences: in design considerations, optimal design patterns, and the psychological states that they encourage in the user. To that end, we propose the following definition, which encapsulates the very exciting alternate design space of Casual Creators.

### Definition

> A Casual Creator is an interactive system that encourages the fast, confident, and pleasurable exploration of a possibility space, resulting in the creation or discovery of surprising new artifacts that bring feelings of pride, ownership, and creativity to the users that make them.

Casual Creators are interactive systems. There are historical examples of non-digital casual creators, like the classic generative art toy Spirograph or the knitting toy Knit Magic, though digital software systems provide more affordances. They do, however, need to be interactive, driven by the user, because the learning and creating process is so core to the psychological experience of using one. Computational creativity can be used to assist the design process, but must be in a mixed initiative partnership with the user.

Casual Creators are tools that create artifacts, of some kind, which may be instances of virtual models or static images, or more abstract artifacts like story grammars or AI behaviors. Each creator has some *possibility space*, the set of all possible artifacts that could be created using that tool. The user creates (or discovers) artifacts by searching or exploring the space for 'good' artifacts. For the casual creator to be successful tool, there must be a way for users to find artifacts meeting functional and aesthetic criteria, avoiding getting stuck in a space of bad artifacts.

The possibility space should be narrow enough to exclude broken artifacts (such as models that fall over or break when 3D printed) but broad enough to contain surprising artifacts as well. The *surprising* quality of the artifacts motivates the user to explore the possibility space in search of new discoveries, a motivation which disappears if the space is too uniform. It also provides feelings of ownership and creativity when the artifact is discovered. In a sufficiently multidimensional possibility space, 'search' and 'creation' become blurred, as the only way to arrive at a particularly interesting artifact is to move through the space intentionally, rather than randomly searching. The user will feel greater ownership and creativity the more they attribute their discovery to their own actions, and pride is increased further when they feel that their discovered artifact is somehow special, *surprising* within the possibility space.

How does the user navigate this possibility space? Do they make tiny adjustments, tentatively inching through the possibility space, or do they make wild jumps, from solution to novel solution, exploring large regions of the space over a short period of time? An optimal creative process is described as making 'creative leaps', so we want to guide the user toward a fast-moving and confident exploration of the possibility space. The user's experience should feel playful, powerful, and pleasurable, like a flow state.

The user of a casual creator is a *casual* user, and the system can expect no previous domain knowledge, no previous technical experience, or adherence to a long learning process. All of the learning and creativity described above must occur in the first few minutes, and provide a good experience even if the user never spends time to gain mastery.

This definition focuses on the design goals of a Casual Creator, the experiences that Casual Creators are particularly suited to create. How we can design tools that *achieve* these goals is explained in the rest of this paper, through a description of design patterns and case studies that tested them. Some patterns come from existing Casual Creator-like tools, like the Spore Creature Creator, Nervous System's design tools (System 2015), and academic experiments like Picbreeder (Secretan et al. 2011). More patterns come from our current understanding of autotelic creativity, anticipating design patterns that support such a psychological state. To predict these new potential patterns, we draw from existing theories of creativity, flow, and design.

## Related Fields

### Reflection-in-action and Direct Manipulation

During the creative design process, the user modifies the artifact, moving quickly through a cycle of evaluation, planning, modification and reevaluation. This process shows the "Reflection-in-action" theory of learning, in which a learner hypothesizes, acts, and reflects on the results as a way to iteratively understand a domain or problem. The originator of that theory, Donald Schon, also applied it to the process of designing (Schon 1992) in which the designer "sees, moves and sees again". The *seeing* and *moving* are grounded in the materials themselves. This cycle cannot take place disembodied in the mind but must be enacted in dialogue with the artifact.

*Seeing* may include the user's visual perception of the artifact, but is also a way to describe the evaluation of the artifact. Is it aesthetically pleasing, stable, strong? How does the designer predict that it will perform in its intended role? Some of these evaluations could be performed or assisted computationally. Schon surmises that while the reflective design process itself is not well suited to unsupervised computational processes, computation could provide new ways of "seeing", or provide constrained micro design spaces "extending the designer's ability to construct and explore them." He concludes that "The design of design assistants is an approach that has not in the past attracted the best minds in AI. Perhaps the time has come when it can and should do so."

In Direct Manipulation (Shneiderman 1993), a UI concept that parallels reflection-in-action, a complex software system provides "continuous representation of the object of interest" and "[r]apid, incremental, reversible operations whose impact on the object of interest is immediately visible" and promises that "after obtaining reinforcing feedback from successful operation, users can gracefully expand their knowledge of features and gain fluency."

The user can manipulate the system with rapid operations, then evaluate the effect immediately, because the artifact is always visible and responds immediately to the modification. The actions are reversible so the user is encouraged to experiment without anxiety, incremental, so subtle changes can be observed, and rapid (and rapidly seen), so that this learning cycle can operate continuously with each tiny iteration.

### Flow

Csikszentmihalyi's Flow theory is influential in games and creativity studies, but seems particularly well suited to the autotelic creativity of casual creators as"[i]deally, flow is the result of pure involvement, without any consideration about results."(Csikszentmihalyi 2000) For flow to be achieved, the activity must have goals to create a sense of progress, immediate feedback so that progress can be sensed, and a balance between their perceived skills and challenges. Flow can be disrupted if the user feels frustrated, intimidated, or overwhelmed by choices.

Flow has a complex relationship with goals. Though the activity should be enjoyable in itself, without the pursuit of an outside reward, goals provide the required *direction and progress*. Goals can be provided as preset challenges, but often it is better to encourage the user to develop their own internal design goals. A good goal can be evaluated moment-to-moment, may change over time, and can be either highly specific to the user, or come from knowledge of the design space.

The flow state is very conducive to both creativity and an autotelic experience, and so provides important design considerations for potential Casual Creators, especially for avoiding conditions which disrupt flow, like choice paralysis or hard failures.

### Creativity Support Tools

Lubart (Lubart 2005) identifies several categories of human and computer collaboration in the creative process: the computer can act as nanny, coach, pen-pal or colleague. Riedl and O'Neill (Riedl and ONeill 2009) suggest "audience" as a fifth role for the computer. These categories provide a useful taxonomy, but do not provide implementable patterns.

The field of Creativity Support Tools, of which Casual Creators could be considered a subcategory focused on autotelic creativity, provides many concrete design patterns. Resnick et al (Resnick et al. 2005) identify many such patterns. Some, like "Support Exploration" and "Make It As Simple As Possible - and Maybe Even Simpler" are patterns to support flow experiences and reflection-in-action styles of learning. Other principles like "Support Many Paths and Many Styles", "Low Threshold, High Ceiling, and Wide Walls" reflect how users will start with diverse goals and skills, which will further evolve as they use the system.

Some principles, "Choose Black Boxes Carefully," "Support Collaboration," and "Support Open Interchange", ask the designer to reflect on the communities in which creative collaboration and learning occur, and how creativity develops as multiple users share knowledge. When we look at the creative communities that flourished for tools like Spore, Scratch (Resnick et al. 2009), and Twine (Klimas 2012), it becomes clear that, though the design of the single-user software is important, the technology decisions of data format, data interchange, hosting, and modifiability are equally critical to enabling creativity and fostering ownership. Creativity occurs between the user and client-side application, but also in the communities of practice that develops outside of the app, so creativity support tools must consider both sites, personal and communal (Maher 2012).

### Generative Methods and Computational Creativity

Computational creativity is the science (and art) of encoding human-style creative process as automatable systems, with the goal of building a system which "exhibits behavior that would be deemed creative in humans." (Colton et al. 2009). How 'creativity' can be detected in the finished artifacts of these systems is its own difficult problem (Maher 2012), but the field has successfully built generators that can design artifacts for domains as diverse as jokes (Petrovic and Matthews 2013) and paintings (Colton 2012).

These systems create artifacts by encoding the process of creating art (or literature, jokes, game levels, music, etc). The resulting algorithms must be able to create not only one

successful example, but a wide and interesting space of possible valid artifacts, some of which should be able to surprise even the person who wrote it. Such algorithms can be called *generative methods* (Compton, Osborn, and Mateas 2013); they use a range of technologies (genetic algorithms, grammars, declarative modeling), but all share the goal of creating large possibility spaces of valid-yet-surprising artifacts. This is the optimal type of possibility space for computational creativity systems, and also for Casual Creators.

Computational creativity and generative methods are often a poor fit for productivity-focused creativity apps. Professional creativity involves creating to very specific requirements, requires complete control and the ability to fine tune the resulting work. Generative methods create a lot of work, very fast, but with minimal control over the output (compared to hand-authored content) and often no way to iterate on the output. A casual user, *without* the need for complete control, is willing to trade a loss of control for the speed, power, and surprise of generative methods.

The expressive range of such systems must always be balanced with the need to produce valid content. A system could produce a wide variety of mostly broken artifacts, or produce a set of high-quality yet homogeneous artifacts, but both of these are failures. We have found the phrase '1000 bowls of oatmeal' useful to describe the common antipattern of generating a set of artifacts which are technically distinct to the computer, but perceived by humans as uniform.

Computational creativity systems usually run autonomously and unsupervised by humans. Pairing these methods with human users can add additional power to the process (Davis et al. 2014), as humans provide aesthetic evaluations and intuitive leaps to the rapid generativity of the computation creativity processes. Mixed initiative systems, in which the computer and human users operate simultaneously or by turn-taking, support a creative cycle in which each user reflects on the previous contributions of their collaborator and modifies the artifact according to their particular abilities. The end products of the creative process are improved, and ideally the user enjoys the experience of collaboration, if the system is well designed. Interaction with a highly generative system has a particular set of pleasures, whether in the context of a game or a creativity tool. Chaim Gingold refers to such pleasurably interactive systems as 'Magic Crayons' (Gingold 2003): computational, accessible, sketchable, expressive systems which invite the user to play with them and discover hidden secrets and affordances.

## Design Patterns

The definition of a Casual Creator as an autotelic creativity tool provides an abstract guide for what we would want a potential Casual Creator to accomplish. To actually design such a tool, these high-level patterns must be interpreted into concrete design patterns. We have identified a number of these patterns, drawing from existing Casual Creators, and from the related fields, and tested them by using them to create a wide variety of systems, described in the Case Studies section below. These design patterns are not exhaustive, but are representative ones that are versatile, common, and easy to apply.

**Instant feedback** Recall that both direct manipulation and reflection-in-action require the user to observe the artifact, make a change, and see the results, a process which allows them to discover patterns and affordances in their possible changes, mastering the system while iterating on an artifact. In the *instant feedback* pattern, the changes should be immediately visible in the modified artifact. However, just visually regenerating the artifact in response to changes, even in real-time, is not necessarily enough to provide appropriate feedback.

'Seeing', in the reflection-in-action model, encompasses more than just 'looking at'. 'Seeing' actually encompasses the entire process of sensing and evaluating the artifact's fitness according to both the potential use case and the user's own design model. For objects with a strictly aesthetic role, this is easy: the user glances at it, and can instantly decide their opinion of it. Other evaluations are complex, and must be either mentally simulated by the user, or else evaluated by the system. Requiring the user to mentally simulate complex consequences will take a lot of time and attention, and the evaluation could be inaccurate or flawed, slowing the iteration process. The *instant feedback* pattern would recommend computationally simulating and visualizing as much as possible so that the user can get feedback at a glance.

**The Chorus Line** Named after the choreography concept in which many dancers all execute the same routine simultaneously, the *chorus line* pattern was used in Spore (Hecker et al. 2008) as an internal tool to test animations on a wide range of creature morphologies. The chorus line is a subpattern for *instant feedback*, in situations where what is being generated is not a single artifact, but a space of artifacts. In that situation, the user should be able to 'see' (in the reflection-in-action sense) the space of their creation, instantly. Instead of generating one example, this pattern suggests generating many examples, and overlaying them (spatially, temporally, graphically) to make subtle differences and similarities easier to spot.

**Simulation and approximating feedback** Automated visualization becomes especially important when the artifact being generated would take minutes or even hours for the user to evaluate, rather than milliseconds for an image, or seconds for an animation. For artifacts such as game levels, the artifact is judged by the many gameplay traces over time that could be played on it, which cannot be visually evaluated with much accuracy by a casual user. Nor can a system show the user all possible gameplay traces, so the user must be shown a proxy of the evaluation. When Riedl and O'Neill (Riedl and ONeill 2009) add 'computer as audience' to Lubart's categories, their simulation proposed to accurately model how a human reader would evaluate generated stories. In Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014), the system calculates "navigational and topological properties" as the user interacts with it, providing instant feedback for a complex artifact. This evaluation does not fully encapsulate the actual gameplay im-

plications of the map, which for a finished level being put in a game, could be a potential design issue. However, for a Casual Creator, the goal of the evaluation is to provide the sense of progress towards a goal necessary for achieving flow. Only the *perception* of progress is necessary: as long as the user perceives progress, the accuracy of the evaluation is irrelevant.

**Entertaining evaluations** One nice benefit of relaxing the need for accurate evaluations is that the evaluations can *themselves* be pleasurable and entertaining. In the Spore Creature Creator, when the user modifies their creature the creature will respond by laughing and shaking the new body part in appreciation, or, less commonly, expressing distress. The choice of happy or sad reaction does not actually represent any real system state, it just provides arbitrary feedback. That feedback is psychologically significant, for encouraging the flow state, but also for letting the user feel pride in pleasing their little AI judges. Even if the user starts with no particular design direction of their own (a common issue with casual artists) having a simulated critic present can suggest a direction for the user, even if they choose to ignore it. The abstract generative art game BECOME A GREAT ARTIST IN JUST 10 SECONDS (Brough and McClure, 2014) waggishly compares the user's glitch art to classic masterpieces and rates it with a percent similarity, an intentionally arbitrary metric that still serves to provide optional direction to the casual user.

**No blank canvas** One benefit of focusing on these intrinsically-motivated users is that they are often much more flexible about the final product. In contrast to a system like Maya, which must support extremely broad use cases and a high degree of fine-tuning in order to make a very particular finished product, a casual user will have more flexible requirements for their product. They likely want it to be functional and aesthetically pleasing, but are willing to consider many more possible kinds of solutions, or may not even start with any particular solution in mind (Nilsson 2003)

Professional artists know the terror that comes from facing a blank canvas (Bayles, Orland, and Morey 2012), but this experience is also intimidating and paralyzing to the novice user. However, this can be very easily mitigated, by providing either a starting shape (Spore) or a suggested challenge (Let's Create! Pottery). The first move is the hardest, so this restricts the first move to a single decision: accept the prompt, or discard it. Once this one move is taken, subsequent actions are easier.

**Limiting actions to encourage exploration** This restriction of actions can be useful even after the user has moved past the blank-canvas moment. To achieve a flow state (Csikszentmihalyi 2000), the user should be able to quickly and confidently make decisions, which is easier if the available choices are appropriate, limited in number, and their consequences are clear (or at least suggested) to the user.

One strategy is modal interaction: limiting actions by the particular mode that the creator is in. This approach is common in character creators like that in World of Warcraft and Spore Creature Creator, which have different modes corresponding to user actions like painting or building and panels with sub-actions within those modes, to choose hair or faces, revealing only actions for the mode that the user is currently in. Another approach is to limit the actions available, slowly unlocking them in response to experience, challenges defeated, purchases, or some other pacing mechanism. If the possibility space is temporarily restricted, the ability to more fully explore the space scaffolds the user's understanding of the possibility space.

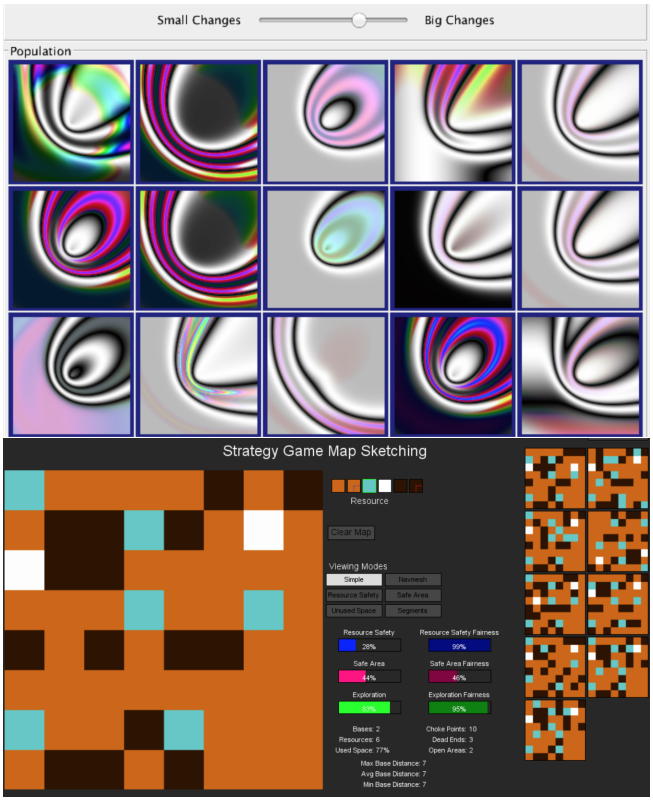



Figure 1: Top: *Mutant-shopping* for images in Picbreeder. Though the user can control the rate of mutation, they can only 'create' an image by selecting the parents of the next generation. Below: Sentient Sketchbook shows automated evaluations, allows direct editing, and also provides some alternate mutants on the right

**Mutant shopping** One feature that can help the user find unexpected solutions in the possibility space is not a creative 'action' at all, but the availability of suggested alternatives, like artifacts near the current one in the possibility space. In some tools, the user is not given any way to edit the artifact, and must navigate the possibility space by picking one of the new options, as in Picbreeder (Secretan et al. 2011).

In other cases, as in the parametric tree modeler Dryad (Talton et al. 2008), the user may use these alternative to browse the space, but can also further edit the artifacts that they discover in that way. A third framing of this pattern is found in Sentient Sketchbook (Yannakakis, Liapis, and

Alexopoulos 2014), in which the user edits a game level normally, but the system uses that information to generate additional suggested artifacts that are 'nearby' for some more abstract calculated metric, rather than 'nearby' in their underlying representation.

Although this process has a lot in common with evolutionary algorithms (specifically human-guided evolutionary algorithms (Klau et al. 2010)), the focus is not on producing an optimal specimen, but on the enjoyment that the user feels from this process. For this reason, we named this pattern *mutant shopping* to capture the psychological pleasures and motivations of a less-directed browsing and discovery process like shopping.

**Modifying the meaningful** In Spore, parts can be placed anywhere on the creatures, then modified by rotation or pulling on their morph handles. In a traditional sculpting program like Maya, these handles would be expected to control a clear parameter like z-scaling, for maximum control over the changes. The Spore designers discovered that it was more interesting to have these handles control higher level changes, like shifting a jaw from top-heavy overbite to jowly underbite, or extending a foot's shape from round toes to pointed claws. Higher-level modifications like these give the user a more meaningful space to explore.

**Saving and sharing** As noted in the "Design principles for tools to support creative thinking" report, the client-side application where the user is editing their artifact is only one site where creativity occurs, and designers of Casual Creators should also consider how they support creativity *outside* of their app. One example of this principle is the use of common, free, human-readable filetypes for saving data, such as JSON or images. Spore embedded the creature's save data stenographically in a PNG image, and the latest version of Twine 2 embeds the editable hypertext into the HTML that plays the Twine game. Even if the client app is still necessary to rebuild the content from the saved data, as in Spore and Twine, users can share their data using existing platforms. Most hosting sites allow text and images, but not arbitrary files. If users can easily host their save files on such hosting sites, they can build communities independently from the makers of the original casual creator app.

**Hosted communities** An alternate pattern is to provide a hosted community that is tied more closely to the client app, as Picbreeder and Let's Create! Pottery do. Casual creators should encourage the user's pride in their discovered or created artifacts, so providing a showcase where user's can publish their work to share it with others supports this feeling of ownership. Creations are often annotated or tagged, and usually there is a commenting and messaging system, enabling a large community to communicate within itself. Modification is its own form of communication, so if the system supports modification of artifacts, they should show their ancestry, and notify the original creators so that they can take pride in their influence.

**Modding, hacking, teaching** Users of casual creators will quickly find that the tool does not support every action that they want. The tool and its surrounding community support should facilitate users in teaching each other mods and hacks that expand the boundaries of what's possible with the tool. The previous two patterns support this pattern, as this teaching can happen on external sites, or internal ones, but the easier it is to find a clever hack, import it into the tool, and modify it and republish the new results, the quicker these ideas will spread through the community.

## Case Studies

### Instant Feedback: PendantMaker

PendantMaker is an online design tool for creating 3D printable pendants. We observed that although 3D printing is interesting to many people, the tools to create printable content are difficult, with many potential pitfalls for making unprintable and broken content. By restricting the domain space to extruded tubes, we could guarantee that our generated geometry would be valid for printing, and print reliably on a cheap printer (a difficult set of physical constraints). When combined with turning sliders, supporting the Direct Manipulation patterns of "rapid, incremental, reversible operations" (Shneiderman 1993), PendantMaker provided a very 'safe' place for the user to experiment without fear of failure.

We also noted that casual users often doubt their drawing ability (Bayles, Orland, and Morey 2012) and lack direction, so we designed a generative algorithm in which undirected scribbles from the user would be reflected around an axis, creating a design of surprisingly attractive symmetry. We provide a canvas for the user to draw a line, which is extruded, shaped, and reflected into the many intersecting tubes on the right, creating the printable pendant *in real time*. This very immediate feedback was critical: users could draw aimlessly, but notice when the reflections would intersect or join together, allowing the users to easily create a complicated knotwork of intersecting tubes that would be impossible to predict without feedback. We also added sliders for a variety of tuning values, reflecting the *Modifying the Meaningful* pattern above. Some sliders corresponded to clear values like thickness and arm count, but 'bloom' performed a complicated sculptural task of flaring the outermost tubes in a curved shape. Complicated tools like bloom are only usable with rapid feedback: their action is indescribable to the user, but with a little experimentation, the user quickly learns how to use them artistically.

### Sharing and Ownership: IceMaker

IceMaker was an evolution of PendantMaker's design, to create extruded 3D snowflakes, and similarly uses tuning sliders, symmetry, and extrusion to create complex geometry that is both modifiable and guaranteed valid, with immediate feedback. The extrusion path is not controlled by the user's drawing, as in PendantMaker, but rather by a particle simulation. The behavior of the particles would be *very* hard for a casual user to program, so instead we provide sliders for values that represent the resulting appearance of the path ('complexity', 'wiggle', 'sharpness') allowing the user to explore the possibility space while not having to understand the complex process behind it (Fig. 2).


Figure 2: Ice-Maker, a 3D snowflake maker, guides the user to create a snowflake and further personalize it with a message, then embed the design into a single URL that the user could share.

Since this interaction provides less agency than the drawing interface in PendantMaker, we wanted to augment IceMaker with other ways to declare ownership of the discovered snowflake. Following the *Saving and Sharing* pattern, we encoded the snowflake into a unique URL which the user could share, post or send as a 'saved' version of their artifact.

## Search and Discovery: Funky Ikebana and Tiny Dancer

Creativity-as-discovery is further explored in Funky Ikebana, in which L-system flowers are are generated from a 'DNA' of floating point tuning values. Similar to exploration process in Dryad (Talton et al. 2008), the user iteratively selects the flowers that they like, and the system generates more nearby examples. This human-guided evolutionary algorithm allows for 'optimization' of the flower, but as this was designed as a Casual Creator, we focus on the pleasures of *mutant shopping* more than potential optimization. The flowers are arranged together, which makes different ones easy to spot, so the user can pick from flowers that are very different, or mostly the same. Regeneration when one flower is selected and its children repopulate the space is instantaneous, so the user can very quickly move through the space of flowers. Picking from one of 10 children limits the number of actions that the user can take, so choice paralysis does not occur, as shown in previous mutant shopping examples like Picbreeder (Secretan et al. 2011).

Unlike the Dryad and Picbreeder systems, we were also able to use the L-system to create a simple animation for the flowers, causing them to 'dance'. Flowers danced differently, an emergent property of their morphology, and users could selectively evolve flowers for their movement instead of just shape. Because we used the *chorus line* pattern to show many flowers dancing at once, the user was able to notice particularly graceful or vigorous ones, and select for that. Tiny Dancer takes this idea further by simultaneously evolving the morphologies of ragdoll dancers and their dance-responses to music, so that the dances can also be selected by mutant shopping on a chorus line.

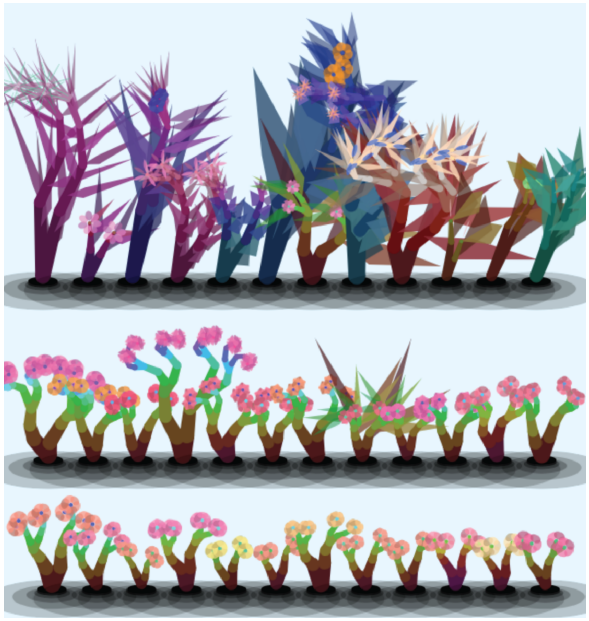


Figure 3: Iteratively evolving smaller flowers in Funky Ikebana, starting with the center flower in the first image. The user's current heuristic is to pick small simple flowers, but that heuristic can change each time the user spots a flower style that they like better.

## Interventions: BotPrint and Binary Fission

The Casual Creator framework has been usefully applied as an intervention in two existing designs, successfully modifying the designs to improve the user's creative experience.

BotPrint was an existing application to design laser-cuttable robotics kits for children. Users could drag handles to shape the outline of the bot's chassis, and some automation would occur to figure out placement of components. Unfortunately, the implications of moving components and changing chassis size were not visible to the users, so making modifications felt meaningless. Using casual creator design patterns, we updated the system to simulate the bots moving in an 'arena' with many other similar bots. This provided a way for the user to evaluate the behavior of the bots visually (chorus line), see and select variants (mutant shopping) and enjoy watching the bots struggle for victory (entertaining evaluations), while also directly modifying the bots and then rereleasing them into the arena.

Binary Fission is a game designed to help the user make binary decision trees to filter loop invariant data for a crowdsourced science task about software verification. At first, this does not seem to be a creative task, but by using casual creator patterns to emphasize the creative side of selecting the filters to build the filter tree, users enjoyed the task and were able to explore the possibility space of trees much faster. The biggest insight provided though the casual creator lens was to show many filters for each choice point. Calculating how well a filter would behave is a very arduous evaluation for the user to perform themselves, so we colored

each by how well it filtered data at that point. Users were able to glance through this potential 'filter space' for suitable filters, and were able to apply them, see their implications, and rebuild trees very quickly, turning what could have been an opaque and arduous task into a fun reflection-in-action learning experience.

## Conclusion

This paper defines a new term, *Casual Creators*, to identify a category of interactive systems which prioritizes the experience of autotelic creativity above productive output, an exciting new design space that is distinct from existing productivity-focused creativity support tools. We have illustrated the distinct design considerations of *Casual Creators* by identifying and describing representative design patterns drawn from theories of creativity and current successful systems. These patterns were used to design several new systems, and to evolve some existing designs to better support casual creativity. From these case studies, we learned that these patterns do clarify and inspire the process of building systems to support casual creativity, as it was easy to identify new system features from the patterns. Additionally, using the lens of *Casual Creators* enabled us to easily find examples of those features in a wider range of existing systems than would have otherwise been possible.

## References


Amabile, T. M.; Goldfarb, P.; and Brackfield, S. C. 1990. Social influences on creativity: Evaluation, coaction, and surveillance. *Creativity research journal* 3(1):6–21.

Bayles, D.; Orland, T.; and Morey, A. 2012. *Art & fear*. Tantor Media, Incorporated.

Boden, M. A. 2009. Computer models of creativity. *AI Magazine* 30(3):23.

Colton, S.; López de Mantaras, R.; Stock, O.; et al. 2009. Computational creativity: Coming of age. *AI Magazine* 30(3):11–14.

Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and creativity*. Springer. 3–38.

Compton, K.; Osborn, J. C.; and Mateas, M. 2013. Generative methods. In *The Fourth Procedural Content Generation in Games workshop, PCG*.

Csikszentmihalyi, M. 2000. *Beyond boredom and anxiety.* Jossey-Bass.

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 *Proceedings of the 5th International Conference on Computational Creativity*.

Gero, J. S. 1994. Towards a model of exploration in computer-aided design. In *Formal design methods for CAD*, 315–336.

Gingold, C. 2003. *Miniature gardens & magic crayons: Games, spaces, & worlds*. Ph.D. Dissertation, Georgia Institute of Technology.

Hecker, C.; Raabe, B.; Enslow, R. W.; DeWeese, J.; Maynard, J.; and van Prooijen, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, volume 27, 27. ACM.

Kaufman, J. C., and Beghetto, R. A. 2009. Beyond big and little: The four c model of creativity. *Review of general psychology* 13(1):1.

Klau, G. W.; Lesh, N.; Marks, J.; and Mitzenmacher, M. 2010. Human-guided search. *Journal of Heuristics* 16(3):289–310.

Klimas, C. 2012. Twine. Accessed: 2015-02-24.

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: who's being creative? In *Proceedings of the 3rd International Conference on Computational Creativity*, 67–71.

Nilsson, B. 2003. 'i can always make another one!'–young musicians creating music with digital tools. *Musicianship in the 21st century: issues, trends and possibilities (Sydney, Australian Music Centre)*.

Petrovic, S., and Matthews, D. 2013. Unsupervised joke generation from big data. In *ACL (2)*, 228–232.

Resnick, M.; Myers, B.; Nakakoji, K.; Shneiderman, B.; Pausch, R.; Selker, T.; and Eisenberg, M. 2005. Design principles for tools to support creative thinking.

Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; et al. 2009. Scratch: programming for all. *Communications of the ACM* 52(11):60–67.

Riedl, M. O., and ONeill, B. 2009. Computer as audience: A strategy for artificial intelligence support of human creativity. In *Proc. CHI Workshop of Computational Creativity Support*.

Schon, D. A. 1992. Designing as reflective conversation with the materials of a design situation. *Research in Engineering Design* 3(3):131–147.

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.

Shneiderman, B. 1993. Direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction* 17:1993.

Shneiderman, B. 2003. *Leonardo's laptop: human needs and the new computing technologies*. Mit Press.

System, N. 2015. Nervous system.

Talton, J.; Gibson, D.; Hanrahan, P.; and Koltun, V. 2008. Collaborative mapping of a parametric design space. Technical report, Citeseer.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative cocreativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*.

# Interaction-based Authoring for Scalable Co-creative Agents

**Mikhail Jacob, Brian Magerko**
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA USA
{mikhail.jacob, magerko}@gatech.edu

**Abstract**

This article presents a novel approach to authoring co-creative systems - called interaction-based authoring - that combines ideas from case-based learning and imitative learning, while emphasizing its use in open-ended co-creative application domains. This work suggests an alternative to manually authoring knowledge for computationally creative agents that relies on user interaction "in the wild" as opposed to high-effort manual authoring beforehand. The Viewpoints AI installation is described as an instantiation of the interaction-based authoring approach. Finally, the interaction-based authoring approach is evaluated within the Viewpoints AI installation and the results are discussed guiding development and further evaluation in the future.

## Introduction

Within the computational creativity community, our research has focused on domains that are open-ended, artistically performative, improvisational, and co-creative between human and AI agent. co-creative AI agents that can succeed in these kinds of domains tend to be large-scale and knowledge-rich since they have to collaborate creatively on an equal footing with humans. Therefore, one of the key bottlenecks for developing co-creative agents has been the knowledge-authoring bottleneck. According to Csinger et al. (1994), the difficulty, cost, or delay in acquisition of expert instantial knowledge followed by its subsequent structuring and storage so as to enable efficient future utilization is often referred to as *the knowledge-authoring bottleneck*. In fact, the knowledge-authoring bottleneck has historically been a significant problem for the intelligent agent community in general and the computational creativity community in particular.

Many solutions have been proposed in the past to mitigate the problem. Case-based reasoning (CBR) approaches and machine learning approaches have utilized online case acquisition and data mining from corpora as fundamental methods for dealing with the knowledge-authoring bottleneck. Data mining approaches have faced a general lack of corpora for instantial or behavioral content within improvisational performative domains. Traditional CBR systems while learning from experience still require instantial content to be authored in the form of an initial case library. Learning by demonstration or observation can avoid these pitfalls, but traditionally require explicit training or teaching phases before they can be used in the final task.

Within the games research community procedural content generation (PCG) research has focused on developing algorithms to generate the instantial content that was once manually authored by expert designers. This has seen success with the development of procedurality-centric games such as Spore and Galactic Arms Race (Hecker et al. 2008; Hastings et al. 2009). However, PCG systems have yet to focus on generating behavioral content that is flexible enough to work in open-ended improvisational domains.

In contrast to the previous authoring approaches mentioned, this article describes a hybrid knowledge-authoring paradigm that combines case-based learning with learning by observation / imitative learning – called *interaction-based authoring*. Interaction-based authoring aims to i) minimize the authoring bottleneck while ii) ensuring that the subjective experience of interacting with the system is high quality and that iii) the computer collaborator supports equal creative agency (the extent to which a creative collaborator can take decisions, make choices, and affect co-creation). It proceeds to demonstrate the interaction-based authoring paradigm within an improvisational interactive art installation called *Viewpoints AI* (Jacob et al. 2013a) after comparing the installation to related work in the field. A brief updated system description is provided (c.f. Jacob et al. 2013b). Finally, the paper details an initial attempt to evaluate the interaction-based authoring approach instantiated within the Viewpoints AI installation and discusses the results as a guide for iteratively developing / refining the installation.

## Interaction-Based Authoring

Interaction-based authoring is a hybrid approach to authoring instantial knowledge and control knowledge for co-creative interactive systems, combining case-based learning with imitative learning. While using an interaction-based authoring approach learning occurs over the lifetime of the full performance and not during an explicit training or teaching phase. This is done to boost participant motivation and engagement encouraging prolonged interaction

with the agent thereby facilitating greater knowledge acquisition.

There are three main aspects to interaction-based authoring. First, case-based learning is used to index and store agent experiences in a reusable manner that can be utilized to drive future behavior or responses in general. Cases can be stored as input–output pairs (from the agent's perspective) with a process to map between inputs and outputs in order to use them interchangeably.

Second, an imitative learning / learning by observation system (Tomasello 2000) that can model the way a human partner responds to the agent's actions is utilized in order to interact with other partners in other interaction contexts. If the new partner's input action (from the new interaction context) is similar enough to an input action it has learnt a model for in the past, it can use that to select an output action. The agent takes the interactor's role in that case and responds, as they (presumably) would have.

Finally, an open-ended co-creative improvisational domain in which to situate the agent is required so that the participant or interactor is engaged and therefore motivated to teach the system for an extended period of time. The open-ended nature of the domain encourages exploration of the interaction space, increasing the coverage of the learning algorithms for future interactions. The co-creative and improvisational aspects of the domain emphasize the egalitarian nature of creative decision-making. They also encourage the user to further explore novel regions of the interaction space in the event that the system makes a 'poor' choice of response, thinking of it as an interesting offer that they hadn't considered rather than a mistake. The interaction-based authoring approach has been instantiated in an interactive improvisational human–AI art installation called *Viewpoints AI*. A description of the installation follows a brief account of related work.

## Related Work

Technology has been contemporarily used to augment performances and art installations (Reidsma et al. 2006; Latulipe 2011; MacDonald et al. 2015). These pieces use performance technology as an integral part of their overall aesthetics and content of the artwork. However, these technologies have been subservient to human performers, with shallow knowledge, and / or a lack of clear collaboration between the machines and humans on stage.

Combining research in arts, AI, cognitive psychology and philosophy, the field of computational creativity has focused on many different creative domains (c.f. Boden 2003; c.f. Colton 2012). However, most traditional computationally creative systems assemble pre-authored content in novel combinations, without attempting to solve the knowledge-authoring bottleneck, leading to small systems with limited scope. In addition, many in the past have ignored creative collaboration or co-creativity focusing on systems that do not involve humans except as consumers or evaluators of the creative artifact or process.

Computationally co-creative systems on the other hand collaborate with humans in order to participate meaningfully in the creative process or outcome. Much work has been done on co-creative agents in the music improvisation domain (Thom 2000; Hsu 2008; Hoffman and Weinberg 2010). The Digital Improv Project virtual agents that could perform theatrical improvisation (O'Neill et al. 2011) and the Computational Play Project virtual agents that could play pretend with people using toys (Magerko et al. 2014) are examples of co-creative systems in other domains. Both however, required extensive pre-authored instantial content to produce improvisational behavior. The Digital Apprentice (a virtual collaborator for abstract visual / sketch art creation; Davis et al. 2014) is a co-creative system that closely resembles an instantiation of the interaction-based authoring approach.

Figure 1: The Viewpoints AI Installation

## The Viewpoints AI Installation

The Viewpoints AI installation is a participatory interactive installation where a human interactor and a virtual agent – named *VAI* – collaborate to improvise movement-based performance pieces together in real-time. The installation (see Figure 1) is composed of a large translucent muslin projection screen that has a human-sized manifestation of VAI projected onto it from the front and the interactor's shadow cast onto it from the rear. This enables an occlusion-free juxtaposition of the interactor's shadow onto the projected virtual agent when their positions overlap. While the installation is highly participatory in nature and the experience of improvising is intrinsically tied to it, an audience can also watch the unfolding performance from the front of the installation.

Participants interact with the virtual agent behind the muslin screen while a Microsoft Kinect depth camera senses and records their movements. Recorded movements are analyzed systematically using a formal version of the Viewpoints framework, as described by Bogart and Landau (2005). Viewpoints is used in theatrical movement and the staging of scenes to focus on the physicality of action and analyze performance in terms of the physical Viewpoints of time (tempo, duration, kinesthetic response, and repetition) and space (shape, gesture, spatial relationship, topography, and architecture), as well as the vocal Viewpoints (pitch, dynamics, acceleration, silence, and timbre). The

participant's movements are interpreted through a subset of the Viewpoints framework and are then responded to by the agent. The formalization of Viewpoints is thus used as a framework to represent and reason about movement.

The Viewpoints AI installation uses contrasting visual elements of light and shadow to showcase how the human participant and the virtual agent arrive at this liminal interaction space from two very different worlds. Visually, VAI is a glowing anthropomorphic character composed of a playful cloud of fireflies. The participant's crisp shadowed form is transported to the ephemeral 2D space between the two worlds through the medium of shadow theatre.

## System Description

The Viewpoints AI installation is powered by an agent architecture that is conceptually composed of three modules – *perception*, *reasoning*, and *action*. Earlier versions of the system are described in Jacob et al. (2013a; 2013b). The following sections describe the agent architecture briefly, going into more detail where necessary to illustrate updated aspects of the system.

## Perception

The Viewpoints AI agent architecture receives input from the depth camera as a frame of joint positions in continuous 3D space at a certain frame rate to get "joint space" gestures. It then discretizes the joint space gestures and derives additional information about them in real-time using a formalization of the Viewpoints framework to get discrete "predicate space" gestures. These two types of gestures are then sent along to the reasoning module.

**Parsing Viewpoints Predicates** The Viewpoints predicates that have been formalized to date make up a subset of the physical Viewpoints, including tempo, duration, and repetition, as well as parts of spatial relationship, topography, shape, and gesture. The current version of the installation has a general purpose machine learning toolkit (Hall et al. 2009) integrated within the agent architecture that classifies Viewpoints predicates using classifiers trained using supervised learning on expert movement-practitioner / dancer data. Adding new predicates to the system is as straightforward as training new classifiers with more data demonstrating or exemplifying that particular attribute or aspect of the Viewpoints framework. Emotional content of the performance portrayed through gestures are also classified through this supervised learning process.

The current movement analysis pipeline employs modular feature detectors for motion-based features (eg. vertical knee velocity, tangential knee acceleration, etc.) of the joint space gestures. These are then used to feed classifiers (with the specific classification algorithm chosen empirically according to classification performance). Training data for classification is obtained by collaborating with expert local movement-practitioners and dancers.

**Turn-taking Model** Turn-taking refers here to the process of naturalistically timing the use of the shared performance space so as to coordinate each other's (potentially overlapping or simultaneous) movements. This can be decomposed into the problems of how to best time the agent's movement turns coordinating with the interactor and how to segment a user movement turn or gesture. The first problem is solved by the interaction convention that the agent moves whenever the interactor does, either mirroring them (when they move arrhythmically) or improvising an original response to their movements (when they perform rhythmic repeated movements). The second problem is discussed below.

In the current version of the installation the agent tries to detect a beat to the interactor's movement (helped by playing dance music during the interaction) and segments their gestures using the detected beat. It does this by creating a set of 1D motion vector-based local beat detectors for each moving joint. These report possible joint-level candidate beats by looking at the half period of the joint motion. When candidate beats are repeated multiple times, they are confirmed and reported to a global tracker. The global tracker then chooses a candidate local beat as the global beat, which is then used to segment the movement at the start and end of the beat duration. Additional trimming of the segment is done so that the start and end are the same.

## Reasoning

Segmented gestures in both joint and predicate space are sent to the reasoning module for the agent to determine an appropriate response gesture. Joint space gestures are then stored in a gesture library in exemplar clusters, each cluster having a universally unique identifier number (UUID). These clusters are produced through an approximate gesture recognition algorithm using a content vector of aggregated versions of the same set of motion-based features used earlier in Viewpoints predicate classification. This is done in order to find patterns in interactions and cluster similar gestures together. It is a simplification of the hard problem of online matching in real-time of an input gesture to one (or potentially none) out of a potentially unbounded set of gestures without prior training of any sort. The corresponding predicate space gesture is then sent to a Soar agent (Laird 2012) for further processing in order to choose a response gesture. This case-based learning is a key mechanism within the Viewpoints AI installation that helps it instantiate interaction-based authoring.

**Response Strategies** The Soar agent has a set of strategies for selecting responses to the input gesture that are then output to the action module. These strategies are selected amongst using pragmatic and aesthetic rules for agent behavior. The response strategies were chosen using an analogy to methods that jazz improvisers use to respond to offers from fellow musicians. For example, repetition is important for establishing a motif, signaling understanding or acceptance of a communicative intention, signaling which performer is being lead by another, etc.

The most important response strategy, which forms the lynchpin of the interaction-based authoring approach, is the

application of observationally learned input–response gesture pairs. The agent observes the collaborator's response to its action and builds an association with parameters to control its application. The use of observed action response association leverages the collaborator's more advanced reasoning faculties in order to respond to some other interactor in another context.

For example, when the agent learns to associate "waving" gesture inputs to "bowing" gesture responses by watching the collaborator execute "bowing" responses to its own "waving" gestures, it can respond using the learned association of "waving" and "bowing" gestures when a new interactor "waves" at the agent. Of course in this example, "wave" and "bow" gestures are actually clusters of gestures with corresponding IDs to which the actual input gestures match approximately (as mentioned earlier) – no semantics of the words "wave" or "bow" are implied to be understood by the agent.

A key assumption that the input response association is based on is that the interactor's response is always related to the previous gesture from the agent and that there is always some reason behind it. Both of these could well be false, if the interactor gets bored and tries something completely new for example. However, associations that are seen more often are given positive reinforcement helping to weed out weaker associations. This role-taking process forms the key mechanism for learning by observation and imitative learning within the Viewpoints AI installation that helps it instantiate interaction-based authoring.

Another response strategy is the selection of emotional reflex reactions to emotional content portrayed in the gestures. There is an "emotional algebra" authored in the system that responds according to a commonsensical set of rules (e.g. responding to angry input gestures with angry or fearful responses). This emotional algebra is rigid and uncomplicated yet enables a simple short-circuit reflex response system to quickly respond to portrayed emotionally salient content within gestures.

An important response strategy is for the system to mimic the interactor's input gesture back to them. Mimicry / repetition is important in facilitating smoother interactions between people (Behrends et al. 2012). In contrast, a (trivial) response strategy involves performing no response at all, though this promotes a sense of uncertainty and is thus discouraged unless as a last resort.

Another response strategy is for the agent to consider an existing gesture and transform it. This creates a variation of that gesture using dimensions or aspects of the Viewpoints framework (eg. faster in tempo, smoother in movement, adding repetitions, etc.). In addition, the system can use acontextual functional transforms to add variety in the enacted form of the gesture, such as reversing the direction of movement, changing the limb in which movement takes place, etc. See Jacob et al. (2013a; 2013b) for more detail.

A final response mode is for the agent to consider past experiences from its episodic memory and choose a similar gesture to bring into the new interaction context. This is achieved using Soar's episodic memory partial graph matching capabilities in order to approximately match the Viewpoints predicates of gestures and / or the direct predicate space representation of their movements from other interaction contexts to the current interaction context. This is valuable to inject novelty into the current interaction context. It uses the lower dimensionality (and higher level of abstraction) of the Viewpoints predicate space to pick a gesture that is roughly midway on a scale of novelty (from completely identical to absolutely novel). This episodic retrieval process is a key mechanism for case-based learning within the Viewpoints AI installation that helps it instantiate interaction-based authoring. Viewpoints predicates form the index vocabulary for the case-based retrieval. It should also be noted that this particular response strategy introduces novelty to the creative experience, balancing the predictability of other response strategies such as the application of observationally learned patterns.

## Action

The action module receives both predicate and joint space gestures from the reasoning module and proceeds to create the suitably transformed and rendered virtual agent embodiment procedurally. The Viewpoints predicates associated with the gesture being performed directly affect the visualization (e.g. the energy of the agent's movements control the colors of the agent). The visual embodiment of VAI is an anthropomorphic figure with a body composed of glowing particles that keep to the bounds of the figure while flying around probabilistically. In the current version of the installation, the agent also has a region around the chest of a corresponding interactor that glows with a diffuse red colour in time to a rhythm if the agent has detected the user moving to a beat. This has the visual effect of a glowing heart beat that rises and falls with the interactor's movements. This was also designed to serve as a subtle form of coordination between the two collaborators. For more detail see Jacob et al. (2013a; 2013b).

## Interaction-Based Authoring Beyond the Viewpoints AI Installation

The Viewpoints AI installation instantiates the interaction-based authoring approach to acquire knowledge from interactors while attempting to provide a high quality subjective experience to the interactors and support their creative agency. It does this through knowledge acquisition of two kinds. Firstly the case-based learning component stores all gesture content it has seen or experienced in episodic memory. Secondly it learns how to use these gestures to respond to people by learning interaction patterns or pairs of gestures from observing people and then imitating their actions in a novel context. Finally the installation is situated in a co-creative performative domain so that there is a low bar for meaningful collaboration as well as to encourage exploration of the interaction space due to player engagement and acquire more knowledge as a result. The approach differs from others by attempting to provide a full-fledged co-creative experience right from the outset

without requiring explicit training or teaching phases.

The approach can be extended beyond the movement-improvisation domain to increase the scalability of other co-creative agents as well. The instantial gesture content that the system learns using case-based learning can be generalized to other types of response content, for example strokes on a canvas or notes played on a synthesizer. Imitation learning in turn can also be used to learn more general response control knowledge. For example, the system could learn patterns of strokes on a canvas or sequences of notes. Currently the Viewpoints AI installation only does a first order pairwise learning of gestures, however that could be extended to higher order sequences of patterns.

## Evaluation Methodology

The following sections describe an initial effort to evaluate the success of the installation in addressing three main research questions. **RQ1**: Can the interaction-based authoring approach minimize the authoring bottleneck? **RQ2**: Can usage of the interaction-based authoring approach create high quality subjective experiences using the system? **RQ3**: Can systems built with the interaction-based authoring approach support collaboration with equal creative agency (the extent to which a creative collaborator can take meaningful decisions, make meaningful choices, and affect the co-creative process or product)?

RQ1 was evaluated with formal analysis of *authorial leverage* (Chen et al. 2009) as an initial attempt. More detailed, pragmatic testing is required next. For the analysis, three cases were compared: 1) a purely mirroring version of the installation where the agent would only mirror the movements of the interactor but not respond in any other way, 2) a version of the installation with a pre-authored tree of 'plot points' (pairs of input gestures and agent responses) of arbitrary length and branching factor, and 3) the full Viewpoints AI interactive art installation.

The RQ2 and RQ3 were evaluated using empirical quantitative and qualitative methods in a pilot study (sample size of 10). For the empirical evaluation, three different experimental conditions were used. Condition 1 had only mirroring of interactor movement as our baseline for comparison. Condition 2 had mirroring of interactor movement along with random movement responses, selected from a library of prerecorded movements, when the participant was performing rhythmic repeated movements. Finally, condition 3 had the full response capabilities of the agent available to respond whenever the interactor was making rhythmic repeated movements. The order of the experimental conditions was also randomized each time. In each case, participants interacted with the experiment for 3 minutes, filled out two surveys administered online, and then debriefed with a semi-structured interview. RQ2 was evaluated using a set of validated survey instruments measuring *system usability*, *flow*, and *enjoyment* of the installation (Brooke 1996; Jackson et al. 2008; Vorderer et al. 2004). RQ3 was evaluated using a set of validated survey instruments measuring the *creativity support index* (CSI) and *effectance* of the installation (Cherry and Latulipe 2014; Klimmt et al. 2007). The individual scales (excluding the CSI) were administered online as part of the IRIS Measurement Toolkit (Vermeulen et al. 2010). The CSI had responses on a 7 point Likert scale while the IRIS Measurement Toolkit used a 5-point Likert scale.

## Results

### Formal Analysis

The interaction-based authoring approach was designed primarily to address the knowledge-authoring bottleneck. Therefore the results of the formal analysis directly estimate how much of an improvement is achieved using this approach for acquiring knowledge within a co-creative agent in the movement improvisation domain. For the three experimental conditions described earlier (as with most existing literature in the field) only the variability was used as a factor for quality of the user experience. Therefore authorial leverage was calculated as the ratio of the number of unique experiences (variability) to the number of authorial inputs involved in creating the system. In addition, a few assumptions were made during the calculation. 1) In order to compare the Viewpoints AI installation variants to existing interactive narrative literature, the notion of plot points was loosened to represent sequences of human – agent movements or gestures. 2) The authorial inputs were considered to be the sum of the number of gestures that were authored prior to the start of the calculation in addition to any manually authored transition rules between them or between interactor gestures and agent responses.

For the first condition evaluating the purely mirroring agent, it was assumed that both interactor and agent movement responses were occurring simultaneously. Thus a plot point would represent the interactor gesture and the same agent gesture performed simultaneously. Therefore the same sequence of **N** interactor gestures input to the system would always return the same sequence of **N** gestures back as responses. The authorial leverage is thus nearly infinite since there is almost no prior manual authoring of instantial content (authorial inputs near zero).

For the second condition with the pre-authored branching tree of input gesture and agent response pairs, a tree of average branching factor b and depth d would have at most a total of $\mathbf{(b \times (b^d)-1) / (b-1)}$ nodes or loose analogs to plot points. Also, such a tree would have at most $\mathbf{(b^d)}$ linear paths through it from root to leaf node representing unique experiences. Therefore, the authorial leverage is roughly $\mathbf{(b^d) \times (b-1) / (b \times (b^d)-1)}$. This function has an asymptotic upper bound of **1** given any **b** or **d**.

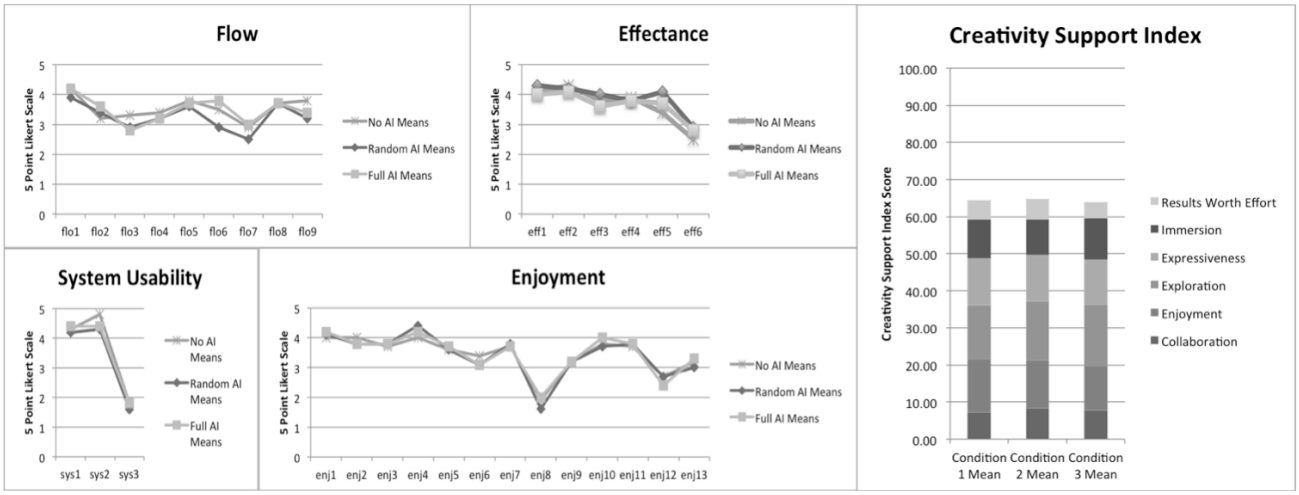



Figure 2: Results of Pilot Empirical Study. For descriptions of specific questions refer provided citations.

Finally, the third condition with the full installation active has the capability to select responses dynamically based on the reasoning processes mentioned earlier. For the first condition, the only possible response was to mirror the interactor's input gesture simultaneously. In the full installation that capability is present (though mimicking not mirroring) in addition to various other responses possible. Therefore the number of unique responses possible for a specific input gesture can only be higher.

For the same **N** input gestures as in the first condition, the number of possible agent responses would be **RN**, where **R** is the total number of unique responses to any one input gesture given a set of response strategies (rather than just mirroring). In the worst case this is **1** and **RN** reduces to **N** possible unique agent responses. In the best case, this becomes $\mathbf{\Sigma R_iN}$ where $\mathbf{R_i}$ represents the total number of unique responses to one input gesture from the $\mathbf{i^{th}}$ response strategy. Each of the response strategies is analyzed below.

For the "no response" response strategy, there is only ever one agent response. For the "repeat input gesture" response strategy, $\mathbf{R_iN}$ is **N** since each input gesture returns the same gesture as the agent output. For the "transform input gesture" response strategy, $\mathbf{R_iN}$ becomes $\mathbf{2^{(V+F)}N}$, where **V** and **F** are the number of Viewpoints dimensions and functional transformations that the agent can use to transform the input gesture into an agent response. In this case $\mathbf{2^{(V+F)}}$ represents the cardinality of the power set containing **(V+F)** elements. For the "emotion algebra" response strategy, the number of emotionally appropriate gestures available to respond with is dependent on the past history of gestures learned by the agent. For a history of **H** gestures with h appropriate gestures, that amounts to **hN** possible responses. In the worst case, this reduces to repeating the input gesture and $\mathbf{R_iN}$ reduces to **N**. This is justified by equating the emotional mirroring taking place with emotional contagion (Hatfield et al. 1994). In the best case, the entire history of gestures has the appropriate emotional content and $\mathbf{R_iN}$ becomes **HN**. For the "novel response from episodic memory" response strategy, the exact magnitude of $\mathbf{R_iN}$ is difficult to estimate for the best case since it is completely dependent on the past ordering of learned gestures and received input gestures. However, the lower bound for $\mathbf{R_iN}$ is **N** since in the worst case, if no gesture is found that is similar to the input gesture, the input gesture is repeated as the agent output. Finally, for the "learned interaction patterns" response strategy, given a set of **b** learned responses on average for the right hand side of each of N input gestures, the $\mathbf{R_iN}$ would be **bN**.

Therefore **RN** or $\mathbf{\Sigma R_iN}$ for all the **i** response strategies in the Viewpoints AI installation becomes $\mathbf{1 + N + 2^{(V+F)}N + N + N + bN}$ or $\mathbf{(1 / N + 3 + 2^{(V+F)} + b)\times N}$ in the worst case. This becomes $\mathbf{(1 + N + 2^{(V+F)}N + HN + \geq N + bN)}$ or $\mathbf{(1 / N + \geq 2 + 2^{(V+F)} + H + b)\times N}$ in the best case. Thus, the number of unique experiences possible with the full installation is much higher than in the first condition. The amount of authorial input is equally minimal in the full Viewpoints AI system. Therefore, since the authorial leverage for the first condition is very large, the authorial leverage for the full Viewpoints AI system is even larger. In addition, if complexity were a factor in our calculation of authorial leverage, it is visibly clear that the full installation has significantly higher complexity in its decision-making and in the user experience offered than the mirroring version of the system.

## Pilot Empirical Study

The aggregated results for both the IRIS Measurement Toolkit and Creativity Support Index are presented in Figure 2. The system usability, flow, and enjoyment scales were used to evaluate the system in terms of its ability to produce high quality experiences for the user. The effectance and creativity support index scales were used to evaluate the ability of the system to co-create alongside the participant with equal creative agency. The results show that each of the experimental conditions did well, though no statistically significant results could be obtained between the different conditions potentially due to the small sample size (sample size of 10). However, regardless of the apparent lack of difference between the conditions, the

survey ratings for the third condition show clearly that usage of the interaction-based authoring approach instantiated within the Viewpoints AI installation can indeed both create high quality subjective experiences for participants interacting with the installation as well as support collaboration with equal creative agency.

The semi-structured interviews were used to guide future development and contained questions regarding feedback about the experience, goals that users had while interacting with the system, what they liked / disliked about the installation, etc. The feedback was overwhelmingly positive, with particular emphasis on the aesthetics of the VAI's visual representation, freedom of creative expression felt by participants, amount of fun had by users, and sheer "cool factor" of the installation. Some of the negative feedback suggested that more was required to show that the agent was actually doing something other than mimicking the user. In addition, potentially indicating a miscommunication of the design goals for the installation, it became clear that some users felt like they should have been able to control the agent's actions to a greater degree. The goals of the users varied depending on how many times they had interacted with it and how inhibited they were. The goals generally went from exploring the boundaries of the system, to trying to get the agent to do certain reactions / responses, to trying to do novel interactions with the system that hadn't been tried before.

## Discussion

The results given above help answer the three questions used to evaluate the interaction-based authoring approach instantiated within the Viewpoints AI installation. Using the interaction-based authoring approach led to a significantly higher authorial leverage (the ratio of variability of the experience, or more generally the quality of the system, to the amount of authorial input) than any pre-authored or pure mirroring version of the installation. The pilot study showed that the interaction-based authoring approach also led to high quality experiences, as judged by the system usability, flow, and enjoyment metrics administered. In addition, the study revealed that the interaction-based authoring approach was able to support collaboration with equal creative agency using the effectance and creativity support index metrics. However, it did not show a significant difference in ratings between the three experimental conditions for any of the survey metrics.

The lack of significant different between ratings for the different experimental conditions could be because of a number of reasons. Firstly, the study was conducted using a very small sample size. However, given that the ratings were so similar for all three, it is also possible that users had difficulty distinguishing between the different conditions in terms of the metrics used. Secondly, in terms of the evaluation, users were blind to the nature of the experimental condition as well as blind to the processes occurring within the virtual agent. According to Colton (2008), the process and the product are equally important to influence evaluation of creativity within the system. Therefore Turing-test style approaches to evaluation are found lacking. This seems particularly true when the creative domain is improvisation where participants evaluate the improvisational experience / process.

The results (especially from the semi-structured interviews) suggest that in order to improve the differential ratings of the full Viewpoints AI installation to the other conditions, the system's actions and outputs should be more noticeably different to highlight the system's original efforts better. This points to the requirement for a more full featured list of implemented transforms (both Viewpoints and functional transforms as well as gestural combination). In addition, video analysis showed that novice users had a hard time triggering the system's rhythmic repeated movement gesture segmentation mechanic. Thus current efforts focus on replacing the existing gesture segmentation algorithm with a more naturalistic automated gesture segmentation algorithm from Kahol et al. (2004). Finally, the experimental design is being refined to make the framing more explicit and will be scaled up.

## Conclusion

In conclusion, this paper introduced a hybrid approach to knowledge authoring for co-creative systems called interaction-based authoring. The approach incorporates ideas from case-based learning and imitative learning, while emphasizing incorporation into open-ended co-creative application domains. This paper then presented an instantiation of the interaction-based authoring approach within the Viewpoints AI installation. The installation was then evaluated in terms of the extent to which it mitigated the knowledge-authoring bottleneck, produced high quality subjective experiences, and supported equal creative agency. Finally, the results of the evaluation were discussed in terms of guiding the future iterative development and evaluation of the installation.

## References


Behrends, A., Müller, S., & Dziobek, I. 2012. Moving in and out of synchrony: A concept for a new intervention fostering empathy through interactional movement and dance. *Arts in Psychotherapy*, *39*(2), 107–116.

Boden, M. A. 2003. *The creative mind: myths and mechanisms*. Psychology Press.

Bogart, A., & Landau, T. 2005. *The viewpoints book: a practical guide to viewpoints and composition*. New York: Theatre Communications Group.

Brooke, J. 2013. SUS : A Retrospective. *Journal of Usability Studies*, *8*(2), 29–40.

Chen, S., Nelson, M. j, Sullivan, A., & Mateas, M. 2009. Evaluating the Authorial Leverage of Drama Management. In *Proceedings of the AAAI 2009 Spring Symposium on Interactive Narrative Technologies II* (pp. 20–23).

Cherry, E., & Latulipe, C. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support

Index. *ACM Transactions on Computer-Human Interaction*, *21*, 1–25.

Colton, S. 2008. Creativity Versus the Perception of Creativity in Computational Systems. In *Proceedings of AAAI Spring Symposium on Creative Systems* (pp. 14–20).

Colton, S., & Wiggins, G. A. 2012. Computational creativity: The final frontier? In L. De Raedt, C. Bessiere, & D. Dubois (Eds.), *20th European Conference on Artificial Intelligence:* (Vol. 242, pp. 21–26).

Csikszentmihalyi, M. 1997. *Flow and the Psychology of Discovery and Invention*. New York: HarperPerennial.

Csinger, A., Booth, K. S., & Poole, D. 1994. AI meets authoring: User models for untelligent multimedia. *Artificial Intelligence Review*, *8*, 447–468.

Davis, N., Popova, Y., Sysoev, I., Hsiao, C.-P., Zhang, D., & Magerko, B. 2014. Building Artistic Computer Colleagues with an Enactive Model of Creativity. In *Proceedings of the Fifth International Conference on Computational Creativity (ICCC 2014)*. Ljubljana, Slovenia.

Hall, M., National, H., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. 2009. The WEKA Data Mining Software : An Update. *SIGKDD Explorations*, *11*(1), 10–18.

Hastings, E. J., Guha, R. K., & Stanley, K. O. 2009. Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games*, *1*(4), 245–263.

Hatfield, E., Cacioppo, J., & Rapson, R. 1994. *Emotional contagion*. *Current Directions in Psychological Science* (Vol. 2, pp. 96–99). Cambridge University Press.

Hecker, C., Raabe, B., Enslow, R. W., DeWeese, J., Maynard, J., & van Prooijen, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)*, *27*(3).

Hoffman, G., & Weinberg, G. 2010. Gesture-based human-robot jazz improvisation. In *Proceedings - IEEE International Conference on Robotics and Automation* (pp. 582–587).

Hsu, W. 2008. Two approaches for interaction management in timbre-aware improvisation systems. In *Proceedings of the International Computer Music Conference (ICMC)*. Belfast

Jackson, S. a, Martin, A. J., & Eklund, R. C. 2008. Long and short measures of flow: the construct validity of the FSS-2, DFS-2, and new brief counterparts. *Journal of Sport & Exercise Psychology*, *30*, 561–587.

Jacob, M., Coisne, G., Gupta, A., Sysoev, I., Verma, G. G., & Magerko, B. 2013b. Viewpoints AI. In *Proceedings of the Ninth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. Boston, MA.

Jacob, M., Zook, A., & Magerko, B. 2013a. Viewpoints AI: Procedurally Representing and Reasoning about Gestures. In *Proceedings of the Digital Games Research Association (DiGRA)*. Atlanta, GA.

Kahol, K., Tripathi, P., & Panchanathan, S. 2004. Automated gesture segmentation from dance sequences. In *Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 883–888).

Klimmt, C., Hartmann, T., & Frey, A. 2007. Effectance and control as determinants of video game enjoyment. *Cyberpsychology & Behavior : The Impact of the Internet, Multimedia and Virtual Reality on Behavior and Society*, *10*(6), 845–847.

Laird, J. 2012. *The Soar cognitive architecture*. MIT Press.

Latulipe, C., Charlotte, U. N. C., Huskey, S., Beasley, R., & Nifong, N. 2011. SoundPainter. In *Proceedings of the 8th ACM conference on Creativity and cognition* (pp. 439–440).

MacDonald, L., Brosz, J., Nacenta, M. a., & Carpendale, S. 2015. Designing the Unexpected: Endlessly Fascinating Interaction for Interactive Installations. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '14* (pp. 41–48).

Magerko, B., Permar, J., Jacob, M., Comerford, M., & Smith, J. 2014. An Overview of Computational Co-creative Pretend Play with a Human. In *Proceedings of First Workshop on Playful Virtual Characters at the Fourteenth Annual Conference on Intelligent Virtual Agents*. Boston, MA.

O'Neill, B., Piplica, A., Fuller, D., & Magerko, B. 2011. A knowledge-based framework for the collaborative improvisation of scene introductions. In *Proceedings of the 4th International Conference on Interactive Digital Storytelling* (pp. 85–96). Vancouver, Canada.

Reidsma, D., van Welbergen, H., Poppe, R., Bos, P., & Nijholt, A. 2006. Towards Bi-directional Dancing Interaction. In *Entertainment Computing - ICEC 2006* (pp. 1–12).

Thom, B. 2000. BoB: an interactive improvisational music companion. In *Proceedings of the fourth international conference on Autonomous agents* (pp. 309-316). ACM.

Tomasello, M. 2000. *The Cultural Origins of Human Cognition*. Harvard University Press.

Vermeulen, I. E., Roth, C., Vorderer, P., & Klimmt, C. 2010. Measuring user responses to interactive stories: Towards a standardized assessment tool. In *Interactive Storytelling* (pp. 38–43).

Vorderer, P., Klimmt, C., & Ritterfeld, U. 2004. Enjoyment: At the Heart of Media Entertainment. *Communication Theory*, *14*, 388–408.

# Imagining Imagination: A Computational Framework Using Associative Memory Models and Vector Space Models

**Derrall Heath, Aaron Dennis** and **Dan Ventura**
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
dheath@byu.edu, adennis@byu.edu, ventura@cs.byu.edu

**Abstract**

Imagination is considered an important component of the creative process, and many psychologists agree that imagination is based on our perceptions, experiences, and conceptual knowledge, recombining them into novel ideas and impressions never before experienced. As an attempt to model this account of imagination, we introduce the Associative Conceptual Imagination (ACI) framework that uses associative memory models in conjunction with vector space models. ACI is a framework for learning conceptual knowledge and then learning associations between those concepts and artifacts, which facilitates imagining and then creating new and interesting artifacts. We discuss the implications of this framework, its creative potential, and possible ways to implement it in practice. We then demonstrate an initial prototype that can imagine and then generate simple images.

## Introduction

The concept of imagination is not often talked about in cognitive psychology without reference to creativity (Gaut 2003; Vygotsky 2004). In fact, the term 'imaginative' is many times used as a synonym for 'creative'. Defining imagination, like creativity, is difficult because the word is used broadly and depends on the audience, the level of granularity, and the context (Stevenson 2003). In cognitive psychology, imagination is commonly generalized as thinking of something (real or not) that is not present to the senses (Beaney 2005). In terms of creativity, it is being able to conceive of and conceptualize novel ideas. Imagination, thus it seems, should be an important consideration when developing creative systems.

In the field of computational creativity, imagination is discussed explicitly only on rare occasions, such as Colton's creative tripod (2008). Most creative systems incorporate imagination implicitly and do not model it directly. In this paper, we propose a computational framework that attempts to explicitly model imagination in order to perform creative tasks. Our framework, called the Associative Conceptual Imagination (ACI) framework, uses associative memory models (AMMs) combined with vector space models (VSMs) to enable the system to imagine and then create novel and interesting artifacts.

We begin by looking more closely at the psychology literature in order to establish a cognitive basis for imagination, which will motivate the design of our framework. We then consider how current computational models of creativity both succeed and fail at addressing imagination. We then outline in detail the ACI framework for imagination and demonstrate an initial implementation (proof-of-concept) in the domain of visual art. Finally, we discuss the possibilities this framework can afford us in building creative systems and talk about questions regarding its application.

## Psychology of Imagination

Imagination is ubiquitous in everyday life. We can visually imagine a world described through narrative, or imagine how to get to the grocery store, or imagine what it would be like to be a celebrity. We can imagine what a lion crossed with an eagle could look like, or imagine new ways to express meaning through art. Although most often thought of as visualizing in the mind, we can imagine in conjunction with any of our senses. Indeed, we can talk about imagination across the whole range of human experience. Imagination is a broad term with many different taxonomies and ways to interpret it. We restrict our view to two major types of imagination that are commonly used by psychologists (Currie and Ravenscroft 2002).

The first type of imagination is *sensory* (or reproductive) imagination. This is mentally recalling past experience, which is directly related to our memories. For example, one can imagine what their favorite food tastes like without actually tasting the food, or imagine their mother's face when she is not present, or imagine an annoying song that is stuck in one's head. This type of imagination can be thought of as creative in the sense of *recreating* in one's mind a previous experience.

The second type of imagination is *creative* (or productive) imagination. It is the ability to combine ideas in different ways never before observed, or the ability to think about the world from a different perspective than previously experienced. For example, one can imagine what a hairy banana monster could look like, or what life would be like if born in another country, or imagine how to compose music that is happy and uplifting. This type of imagination is more clearly tied to creativity and some have argued that it forms a necessary basis for creativity (Vygotsky 2004), while others have argued that imagination is merely a tool used in the creative process (Gaut 2003).

Most psychologists agree that our senses, our conceptual knowledge, and our memories form the bases of imagination (Beaney 2005; Barsalou 1999). As we perceive the world and have experiences, we create memories by establishing and strengthening connections in our mind. These connections form concepts, which are in turn interconnected. Memories are often argued to be distributed and content addressable across groups of neurons (Gabora and Ranjan 2013). This means that multiple neurons respond in varying strengths to certain experiences, different experiences may activate overlapping neurons, and similar experiences will have more overlapping neurons than dissimilar experiences. This distributed memory allows the brain to implicitly associate concepts and experiences together.

Thus we have associations between concepts (e.g., rain is related to water) and between what we perceive and these concepts (e.g., apples look round and are typically reddish in color). Creative imagination cannot make something out of nothing, nor is it random; everything we imagine is anchored to things we have actually experienced in the past and on their connections (Vygotsky 2004). The novelty is in combining these experiences in different ways. When a chef imagines new recipes, she uses her knowledge of existing recipes, ingredients, methods, and kitchen tools. The new recipe is essentially a recombination of this previous information in a novel and (hopefully) delicious way.

A computational model of imagination should address the abilities to perceive, to create memories, and to learn associations between concepts. Such a model should then be able to reconstruct this information (sensory imagination), as well as recombine this information in novel ways to create new and interesting things never before experienced (creative imagination).

## Related Work

In accounting for creativity in computational systems, Colton was one of the first to explicitly mention imagination as part of the creative process (2008). In order for a system to have imagination, it should be able to produce artifacts that are novel. Others have mentioned imagination in relation to a creative system that produces narratives (Zhu and Harrell 2008).

A computational system that explicitly tries to model imagination is SOILIE (Science Of Imagination Laboratory Imagination Engine) (Breault et al. 2013). SOILIE maintains a large database of labeled images, and words are associated together when they appear as co-occurring labels. For example, a picture of a face could be labeled with 'face', 'ear', 'mouth', etc. and the system learns to associate those labels together. A word is given to the system which then finds 5-10 associated words and creates a collage out of images that have been labeled with those associated words. This system demonstrates a rudimentary form of sensory imagination in which it tries to recreate an image of the inputed word. SOILIE is similar to one of the abilities of the Painting Fool, which can extract key words from a text document and create a collage by finding images of those key words in a database (Krzeczkowska et al. 2010).

Creative imagination was partially demonstrated in a system that used recurrent neural networks to produce melodies according to a set of other melodies arranged on a 2D plane (Todd 1992). Each of the melodies in the training set were tied to a specific 2D location, and the model was trained to reproduce each melody at their respective locations. After training, the system would be given a new location on the 2D plane and could essentially interpolate a new melody according to its proximity to the original set of melodies. This is the beginnings of creative imagination in that the system is blending melodies together according to spacial proximity.

Imagination has been mentioned in conjunction with systems that perform conceptual blending to produce metaphors and narratives (De Smedt 2013; Zhu and Harrell 2008; Veale 2012). Conceptual blending is the process of taking two input mental spaces (representing concepts) and mixing them together to make a blended mental space that is novel, meaningful, and has emergent structure (e.g., lightsaber is a blend of sword and laser) (Fauconnier and Turner 1998). Computational models of conceptual blending have been used to produce narrative (Permar and Magerko 2013), poetry (Harrell 2005), and even mathematical axioms (Martinez et al. 2011).

Conceptual blending certainly has potential for imagination as it explicitly attempts to blend conceptual knowledge into novel ideas. Although there are still many technical challenges in autonomously blending input spaces, conceptual blending does seem to address creative imagination. Unfortunately, most implementations do not consider sensory information and the input spaces are typically hand engineered, so the system does not learn from experience and cannot imagine sensory type artifacts. However, one computational system does try to implement conceptual blending with images (Steinbrück 2013). The system takes two pictures that each represent a concept and blends them by extracting commonly shaped objects in one image and pasting them over similarly shaped objects in the other image (e.g., a globe in one image is pasted over a bicycle tire in another image).

Evolutionary computation is a common method incorporated into creative systems because of its innate ability to yield unpredictable yet acceptable results (Gero 1996). Indeed, evolutionary computation seems to at least partially model creative imagination in that it recombines and modifies existing artifacts through crossover/mutation and can, thus, diverge and discover novel artifacts. The fitness function also guides the evolutionary process to converge on quality results. Many systems incorporate the use of evolutionary techniques to produce artifacts in domains such as visual art (Machado, Romero, and Manaris 2007; DiPaola and Gabora 2009; Norton, Heath, and Ventura 2013), music (Miranda and Biles 2007), and semantic networks (Baydin, de Mántaras, and Ontañón 2014).

Evolutionary computation appears to have potential in addressing both sensory and creative imagination. However, the creative intent seems to reside solely in the fitness function, which is separated from the actual generation of artifacts. The creation of artifacts is an independently ran-

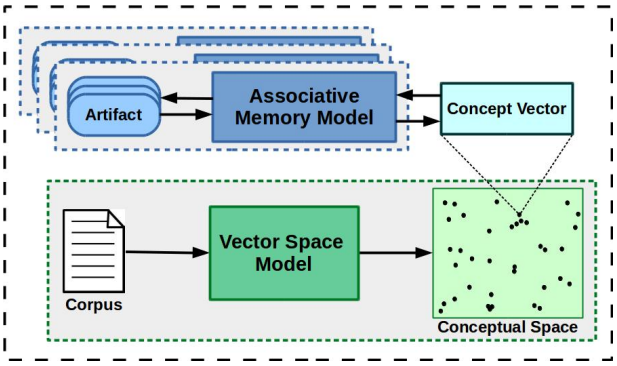



Figure 1: An overview of the Associative Conceptual Imagination framework. The vector space model learns, from a large corpus, how to encode semantic information into concept vectors that populate conceptual space. Multiple associative memory models can then learn associations between these concept vectors and example artifacts from various domains, such as art, music, or recipes. These associative memory models are bi-directional and can not only discriminate, but also generate artifacts according to a given concept vector. The semantic structure encoded in the concept vectors allows the framework to facilitate the imagining of artifacts according to concepts for which it has never seen examples.

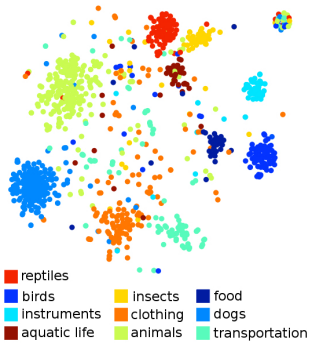



Figure 2: A 2D visualization (projected from high dimensional space) of several word vectors color coded by topics. These concept vectors were learned using the skip-gram VSM, which was incorporated into the DeViSE model (visualization courtesy of Frome et al. 2013). Note that concepts from similar topics generally cluster together because the concept vectors encode semantic relationships.

dom event that is not connected to any associations learned through experience (except for maybe the population of artifacts themselves). The act of imagination in this case is mostly a selection/filtering process, which, although viable, doesn't seem to capture the complete picture. In its basic form at least, there is no notion of associations between concepts and artifacts.

## Associative Conceptual Imagination

We attempt to explicitly model imagination through a computational framework called the Associative Conceptual Imagination (ACI) framework. ACI uses ideas from other domains in a novel way that is capable of both sensory and creative imagination. ACI is composed of two major types of components, a vector space model and associative memory models as shown in Figure 1. We will discuss the major components of the ACI framework, how they interact to perform various imaginative tasks, and the creative potential of systems built using this framework.

### Vector Space Model

Creativity is valued not just because of the novelty of things created, but also because of their utility. For example, in domains such as visual art, the value is in how the art conveys meaning to the viewers (Csíkzentmihályi and Robinson 1990). There is an element of intentionality as an artist purposefully expresses meaning through art. How can an artist intentionally express meaning without having knowledge of the world and of what things mean? Conceptual knowledge helps to provide a foundation for the ability to imagine and create. Incorporating conceptual knowledge into a creative system can potentially be achieved through Vector Space Models (VSMs) (Turney and Pantel 2010).

It is commonly agreed that a word (or concept), at least in part, is given meaning by how the concept is used in conjunction with other words (i.e., its context) (Landauer and Dumais 1997). Vector space models take advantage of this by analyzing large corpora and learning multi-dimensional vector representations for each concept that encode such semantic information. These models are based on the idea that similar words will occur in similar contexts and words that are often associated together will often co-occur close together. These models reduce words to a vector representation that can be compared to other word vectors. VSMs have been successfully used on a variety of tasks such as information retrieval (Salton 1971), multiple choice vocabulary tests (Denhière and Lemaire 2004), TOEFL multiple choice synonym questions (Rapp 2003), and multiple choice analogy questions from the SAT test (Turney 2006).

Concepts similar in meaning will have vectors that are close to each other in "vector space", which we will refer to as *conceptual space*. Associations between concepts are implicitly encoded by their proximity in conceptual space. Figure 2 shows relationships between example word vectors that correspond to various topics projected onto a 2D plane. These concept vectors capture other interesting semantic relationships that are consistent with arithmetic operations. For example, $vector(\text{“}king\text{”}) - vector(\text{“}man\text{”}) + vector(\text{“}woman\text{”})$ results in a vector that is closest to $vector(\text{“}queen\text{”})$.

The potential of VSMs in creative systems has been discussed before, and we aim to make use of them in this framework (McGregor, Wiggins, and Purver 2014). The semantic information encoded in the vectors provides a form of conceptual knowledge to the ACI framework, which will help provide a basis for imagination.

## Associative Memory Models

In addition to knowing how concepts relate to each other, the ACI framework needs to allow understanding of how concepts relate to actual artifacts. In other words, ACI systems should be able to perceive and observe the world (i.e., to be grounded in sensory information). ACI incorporates Associative Memory Models (AMMs) to learn how to associate artifacts with concept vectors. For example, models built using ACI can learn what a 'cat' looks like by observing pictures of 'cats', or learn what a 'car' sounds like by listening to sound files of 'cars'.

Here we use "associative memory model" as a generic term that refers to any computational model or algorithm that is capable of learning bi-directional relationships between artifacts and concept vectors. Not only should the AMM be capable of predicting the appropriate concept vector given an artifact, but it should also be capable of going the other direction and producing an artifact given a concept vector. Of course, the quality of learning will be dependent on the quality and quantity of labeled training data, as well as on the characteristics of the particular associative memory model that is chosen.

Bidirectional associative memory models (BAMs) seem like an obvious possible choice to implement the AMM (Kosko 1988). A BAM is a type of recurrent neural network that learns to bidirectionally map one set of patterns to another set of patterns. Given an artifact (encoded into a pattern), a BAM could return the appropriate concept vector. Conversely, given a concept vector, a BAM could return an appropriate artifact, which can essentially be thought of as performing sensory imagination. Variations of BAMs have been used in computational creativity to associate input patterns to features in order to model the phenomenon of surprise (Bhatia and Chalup 2013).

Another family of algorithms that have potential use in the ACI framework are probabilistic generative models. These models learn a joint distribution for observed data and their respective labels/classes. Once trained, not only can these models classify new data, but they can also be used generatively to create new instances of data that correspond to a particular label. For example, a Deep Belief Network (DBN) is a generative model that can also be thought of as a deep neural network in which several layers of nodes (or latent variables) are connected by weights from neighboring layers, while nodes of the same layer are not connected (Hinton, Osindero, and Teh 2006). Hinton *et al.* used DBNs to classify images of handwritten digits (0-9) by training on several examples and then used them generatively to "imagine" what a 2 looks like by creating several images that each uniquely looked like a handwritten two, thus demonstrating a form of sensory imagination.

Another generative model uses a hierarchical approach to recognize and then generate unique images of handwritten symbols, again demonstrating sensory imagination (Lake, Salakhutdinov, and Tenenbaum 2013). Sum Product Networks (SPNs) have also been used to learn bidirectional associations between patterns (Poon and Domingos 2011). Given a picture of half a face, SPNs were able to infer (or imagine) the other half. These generative models can often be applied directly to the raw inputs (i.e., directly to pixels in an image) and thus seem to exhibit advanced perceptual abilities and in turn can generate artifacts directly.

The associative memory model implementation is not limited to a single model, but could be split into separate discriminative and generative parts. A machine learning algorithm could be the discriminative part and be trained to predict a given artifact's concept vector (e.g., given a 'sad' melody, the learning algorithm predicts the 'sad' vector). The generative part could be implemented by a genetic algorithm that uses the discriminative model as the fitness function. For example, a genetic algorithm could be given the 'sad' vector to imagine a 'sad' melody, and the discriminative model knows what characteristics a 'sad' melody should have and could then guide the evolutionary process.

Other specific associative memory models could be incorporated depending on the domain, its representation, and available training data. Additionally, multiple AMMs for different domains could be incorporated into the framework simultaneously (i.e., one model learns images while another learns sounds for each concept), with the AMMs then indirectly related through conceptual space.

## Performing Imagination

Once an implementation of the ACI framework has its components in place and properly trained, it is ready to imagine, and even create, artifacts. To perform sensory imagination, an ACI model can generate artifacts for a particular concept that it has previously learned. For instance, after having seen example images of 'cats', the system has learned an internal representation for what a 'cat' looks like. The associative memory model can then start with the 'cat' concept vector and generate a unique image that would likely be associated with the 'cat' vector, presumably an image of a 'cat' (see Figure 3(a)). In the case of using probabilistic generative models, the probabilistic nature of the model and the distribution of various poses, angles, and colors learned from the many example 'cat' images allow the system to generate a unique 'cat' image each time.

To perform creative imagination, the framework takes inspiration from the DeViSE model, which uses VSMs to aid in correctly recognizing images of objects (Frome et al. 2013). The DeViSE model first learns word vectors from a large corpus using a VSM. The model is then trained with raw image pixels using a deep convolutional neural network that learns to predict the correct labels' vector (instead of the label directly). Cosine similarity is performed between the predicted vector and the other word vectors to determine what the correct label should be. Since the vectors encode semantic relationships between concepts, the model can successfully label an image with a word for which it has never seen example images (called zero-shot prediction). For example, the system may have been trained on images of 'rats' and 'mice' but not on images labeled 'gerbil'. Given a picture of a 'gerbil' the model can still successfully label it as such because a 'gerbil' is similar (according to the VSM) to a 'rat' and a 'mouse'.

Replacing the convolutional neural network with, say, a probabilistic generative model could allow the system to act

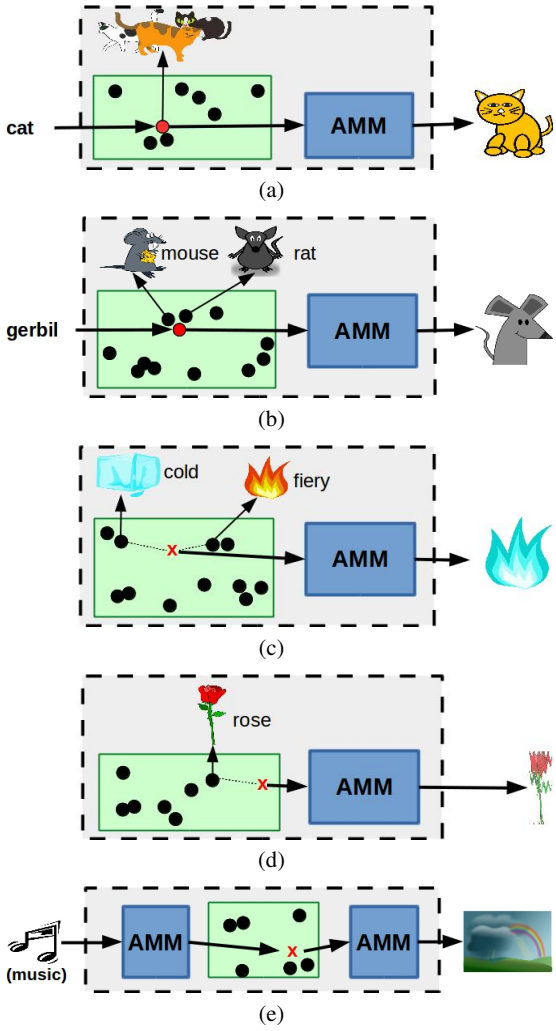



Figure 3: Different ways the Associative Conceptual Imagination framework can be used to imagine artifacts. The green rectangle with black dots represents concept vectors in conceptual space, which are learned from a vector space model. The Associative Memory Model (AMM) associates concept vectors to artifacts. The framework allows the imagining of artifacts for concepts it has previously observed (a). It can facilitate the imagining of artifacts for concepts it has not previously observed but that are similar to other concepts that is has observed (b). The framework allows the imagining of artifacts that are combinations of two (or more) previously observed concepts (c). Models based on ACI can imagine changes to a previously observed concept (d). Finally, the framework can facilitate imagination across different domains by observing an artifact in one domain and then imagining a related artifact in another domain (e).

in reverse. We could input the vector for 'gerbil' and the system could imagine what a 'gerbil' looks like without having ever seen a picture of a 'gerbil', because of the semantic knowledge encoded in the vectors (see Figure 3(b). Similarly, the system could take advantage of the semantic structure of the VSM and imagine what a concept sounds like without having heard any example sounds for that concept. For example, the system could have been trained on sounds for 'horses', 'tractors', 'dogs', and 'trumpets', but not have been exposed to any sounds for 'donkeys'. Yet, the system could still generate a unique sound for a 'donkey'. The result may not sound exactly like a 'donkey', but it will sound closer to a 'horse' than to the other concepts because the system knows that 'donkeys' are more similar to 'horses' than to the other concepts. An ACI model can imagine its own 'donkey' sound in a way that is novel, yet still reasonable by leveraging semantic information gained through the VSM and transferring it to the task of generating sound.

In another situation, a system based on ACI can imagine what a combination of concepts could look like by starting with a vector that is in between concepts in conceptual space. As shown in Figure 3(c), the system could imagine what a 'cold' and 'fiery' image looks like by starting with a vector part-way between the 'cold' and 'fiery' vectors. The system should generate a novel image that is some blending of the two concepts (and perhaps other surrounding concepts). The system is essentially imagining what new combinations of concepts look like, while being anchored in past experience.

ACI could facilitate the imagining of distortions to existing concepts by gradually venturing away from a concept's vector along different dimensions (see Figure 3(d)). The system could generate images of 'roses' starting with the 'rose' vector, but then gradually move away from the 'rose' vector. The resulting images should become distorted depending on the direction and distance from the original vector.

Finally, an ACI model could generate artifacts across different domains. The system could learn, using separate associative memory models, what concepts look *and* sound like. Given a picture of a 'dog', the system could then imagine what the 'dog' sounds like. The ACI model simply uses the AMM for images to predict the vector associated with the 'dog' picture and then feeds that predicted vector into the AMM for audio and has it generate a unique sound. The system could also be given a melody and then imagine an image to go with it, the two domains being tied together through the conceptual space as shown in Figure 3(e).

The ACI framework provides potential for these types of imaginative (and creative) abilities. It has been designed to model imagination by learning conceptual knowledge, perceiving concepts (artifacts), and generating novel artifacts never before experienced in several ways. Of course, this is only a framework, and the actual power of it depends on the abilities of the specific VSM and AMM implementations chosen for each domain (and their training data). Current state-of-the-art models are probably not yet capable of generating (or even classifying) large, detailed images of arbitrary concepts at the pixel level. Nor are they likely yet able to perceive sophisticated music in the general case. However, these capabilities do seem to be on the horizon with the advent of generative deep learning systems (such as DBNs).

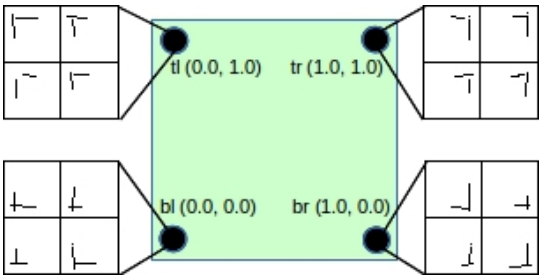



Figure 4: Example training images for each of the four known 2D vectors shown in conceptual space.

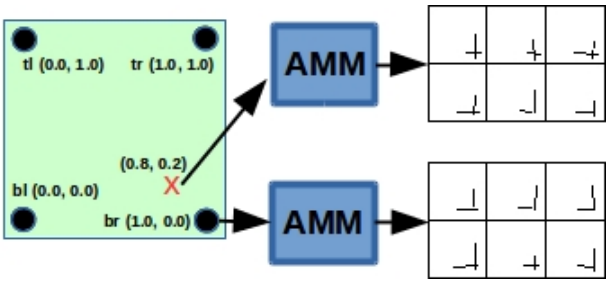



Figure 5: The bottom set of images were imagined for the vector $\vec{br} = (1.0, 0.0)$, which is one of the four vectors on which the system had been trained. The top set of images were imagined for the vector $(0.8, 0.2)$, which is a vector on which the system was not trained. The top images are similar to the bottom images because the vector $(0.8, 0.2)$ is close, in conceptual space, to the known vector $\vec{br} = (1.0, 0.0)$.

## Imagining Images

In order to show how the ACI framework could work in practice, we created a simple toy implementation that can imagine basic binary images. Instead of using a vector space model, we manually specified the conceptual space as a 2D plane in order to more easily visualize how images at various vector locations relate to one another. We then chose four vectors in the 2D conceptual space that are spatially located at four corners. The four vectors are $\vec{tl} = (0.0, 0.1)$, $\vec{tr} = (1.0, 1.0)$, $\vec{bl} = (0.0, 0.0)$, and $\vec{br} = (1.0, 0.0)$, to which we will refer as the *known* vectors.

We then generated four sets of training images for each of the four known vectors that are $32 \times 32$ pixels in dimension and are binary (i.e., black and white). The training images were pictures of actual corners, and example images for each of the four known vectors can be seen in Figure 4. We implemented the associative memory model using a sum product network (SPN) and trained the SPN using corner images paired with their associated known-vectors (perturbed slightly using Gaussian noise). To learn the structure and parameters of the SPN, we used a modified version of the LEARNSPN algorithm that is able to accommodate both categorical and continuous random variables (Gens and Domingos 2013). The result was a model that represents a joint probability distribution over image-vector pairs. We used the efficient, exact-inference capabilities of the SPN to generate novel images by sampling from the conditional probability distribution of images, conditioned on the concept vector. This was done by clamping the concept vector to a specific value and sampling the image pixel variables.

The model can perform sensory imagination by generating images for each of the four known vectors that it has learned. The bottom set of images in Figure 5 are example images imagined for the $\vec{br} = (1.0, 0.0)$ vector. Notice how each imagined image is unique yet still looks like the training images in Figure 4.

The system can also perform creative imagination by generating images for vectors for which it has never seen example images. These imagined images should look more similar to nearby known vectors than to known vectors farther away. The top set of images in Figure 5 were produced for the vector $(0.8, 0.2)$. These images are indeed similar to the images at vector $\vec{br} = (1.0, 0.0)$ (bottom set), which is the closest known vector. Although the system was never shown images for vector $(0.8, 0.2)$, it could still imagine what the images could look like by leveraging the information represented by the vectors in conceptual space (in this simple case just spatial information).

To further illustrate the imagining capabilities in this simple example, we had the system generate images at vector locations all over the 2D plane in 0.1 increments. In order to help visualize how the various generated images transition along conceptual space, we generated 100 images at each vector location and averaged them into a single image. We then arranged each averaged image on the plane according to their respective 2D vector (see Figure 6).

Moving from corner to corner on the 2D plane essentially shows the known images morphing into each other. The center image becomes a blend of all four corner shapes, while the images in the middle of the edges are a blend of the two corners on that edge. The model has only seen images for the corner vectors, which provide a basis for the other vectors in the 2D plane. The model cannot imagine images that do not relate to the four known corner images, which the results seem to confirm.

Admittedly, this toy example with a small 2D conceptual space and simplistic binary images is not visually impressive. It may be hard to ascribe imagination to a model that just seems to be doing a form of interpolation. Keep in mind that this example is only intended to be a proof-of-concept that demonstrates how the framework could work to generate actual artifacts. This example also allows us to understand why the model is generating the images that it does—because of the training images (perceived artifacts) and the spacial arrangements of the vectors (conceptual relationships). A full implementation of this framework would be dealing with thousands of concepts in a conceptual space hundreds of dimensions in size, which is a much richer representation of conceptual knowledge. Also working with real artifacts, such as actual visual art or music, has the po-

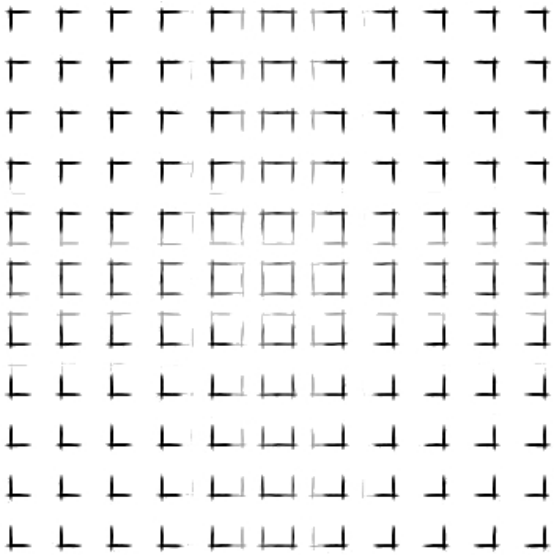


Figure 6: The average of 100 rendered images for each 2D vector in conceptual space at 0.1 increments. The system was trained on example images only for the vectors located at the four corners and then the system had to imagine what images at vectors in the middle would look like based on the images observed for each of the four corner vectors. Note how the images start to blend together as their corresponding vector approaches the middle of the space.

tential to yield much more impressive results.

## Conclusions and Future Work

We have outlined the Associative Conceptual Imagination framework, which models how imagination could occur in a computational system that generates novel artifacts. The ACI framework accounts for the cognitive processes of learning conceptual knowledge and concept perception (via artifacts). The framework proposes using vector space models to learn associations between different concepts, and using associative memory models to learn associations between concepts and artifacts. This network of associations can be leveraged by the system to produce novel artifacts.

We have demonstrated a basic implementation of ACI and applied it to simple binary images. We showed that the system could perform both sensory and creative imagination through the images it was able to produce.

The ACI framework poses some interesting questions. How will this framework perform when applied to real artifacts? What implementation and corpus should be used for the VSM? What models are appropriate to use for the AMMs? Does the choice of the model depend on the domain? Does the choice of the model depend on the artifact's representation (e.g., an image could be represented by raw pixels, extracted image features, or parameters to a procedural algorithm)? Research needs to be done to implement and refine this framework for various domains in order to explore these questions, and we are confident that the ACI framework will be useful for computationally creative systems.

In future work, we plan to apply the ACI framework to DARCI, a system designed to generate original images that convey meaning (Heath, Norton, and Ventura 2014). We plan to use the *skip-gram* VSM (Mikolov et al. 2013) trained on Wikipedia, which will learn vectors for 40,000 concepts in 300 dimensional space. Initially, we intend to implement the AMM using a discriminative model and a genetic algorithm. We will use 145 descriptive concepts (e.g., 'violent', 'strange', 'colorful', etc) to train the discriminative model to recognize those concepts in images. For example, the model will learn to predict the 'scary' vector when given a 'scary' image.

Once trained, the discriminative model will act as the fitness function to the genetic algorithm, which can then render images in ways that convey descriptive concepts (i.e., it can render a 'sad' image). The system will also be able to render images that convey concepts on which it has not been trained (beyond the 145) because of the semantic relationships encoded in the vectors. In other words, it will be able to imagine what other concepts would look like based on past experience and conceptual knowledge.

This framework could also be extended to include ideas involving conceptual blending. As it stands, the conceptual space does not change once the VSM learns the concept vectors and blending occurs through the associations between concepts and artifacts. It could be interesting to find ways to blend the concepts themselves together to produce new concepts that can then be expressed through artifacts.

## References


Barsalou, L. W. 1999. Perceptions of perceptual symbols. *Behavioral and Brain Sciences* 22(04):637–660.

Baydin, A. G.; de Mántaras, R. L.; and Ontañón, S. 2014. A semantic network-based evolutionary algorithm for computational creativity. *Evolutionary Intelligence* 8(1):3–21.

Beaney, M. 2005. *Imagination and Creativity*. Open University Milton Keynes, UK.

Bhatia, S., and Chalup, S. K. 2013. A model of heteroassociative memory: Deciphering surprising features and locations. In *Proceedings of the 4th International Conference on Computational Creativity*, 139–146.

Breault, V.; Ouellet, S.; Somers, S.; and Davies, J. 2013. SOILIE: A computational model of 2d visual imagination. In *Proceedings of the 12th International Conference on Cognitive Modeling*, 95–100.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. *Creative Intelligent Systems: Papers from the AAAI Spring Symposium* 14–20.

Csíkzentmihályi, M., and Robinson, R. E. 1990. *The Art of Seeing*. The J. Paul Getty Trust Office of Publications.

Currie, G., and Ravenscroft, I. 2002. *Recreative Minds: Imagination in Philosophy and Psychology*. Oxford University Press.

De Smedt, T. 2013. *Modeling Creativity: Case Studies in Python*. University Press Antwerp.

Denhière, G., and Lemaire, B. 2004. A computational model of children's semantic memory. In *Proceedings of the 26th Conference of the Cognitive Science Society*, 297–302. Mahwah, NJ: Lawrence Erlbaum Associates.

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.

Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive Science* 22(2):133–187.

Frome, A.; Corrado, G.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; and Mikolov, T. 2013. DeViSE: A deep visual-semantic embedding model. In *Advances In Neural Information Processing Systems*, 2121–2129.

Gabora, L., and Ranjan, A. 2013. *How Insight Emerges in Distributed, Content-addressable Memory*. Oxford University Press.

Gaut, B. 2003. Creativity and imagination. *The Creation of Art* 148–173.

Gens, R., and Domingos, P. 2013. Learning the structure of sum-product networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, 873–880.

Gero, J. S. 1996. Creativity, emergence, and evolution in design. *Knowledge-Based Systems* 9:435–448.

Harrell, D. F. 2005. Shades of computational evocation and meaning: The GRIOT system and improvisational poetry generation. In *Proceedings of the Sixth Digital Arts and Culture Conference*, 133–143.

Heath, D.; Norton, D.; and Ventura, D. 2014. Conveying semantics through visual metaphor. *ACM Transactions on Intelligent Systems and Technology* 5(2):31:1–31:17.

Hinton, G.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18(7):1527–1554.

Kosko, B. 1988. Bidirectional associative memories. *IEEE Transactions on Systems, Man and Cybernetics* 18(1):49–60.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation — with intent. In *Proceedings of the 1st International Conference on Computational Creativity*, 36–40.

Lake, B. M.; Salakhutdinov, R. R.; and Tenenbaum, J. 2013. One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems*, 2526–2534.

Landauer, T., and Dumais, S. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition induction and representation of knowledge. *Psychological Review* 104(2):211–240.

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.

Martinez, M.; Besold, T.; Abdel-Fattah, A.; Kuehnberger, K.-U.; Gust, H.; Schmidt, M.; and Krumnack, U. 2011. Towards a domain-independent computational framework for theory blending. In *2011 AAAI Fall Symposium Series*.

McGregor, S.; Wiggins, G.; and Purver, M. 2014. Computational creativity: A philosophical approach, and an approach to philosophy. In *Proceedings of the 5th International Conference on Computational Creativity*, 254–262.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

Miranda, E. R., and Biles, A. 2007. *Evolutionary Computer Music*. Springer.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2):106–124.

Permar, J., and Magerko, B. 2013. A conceptual blending approach to the generation of cognitive scripts for interactive narrative. In *Proceedings of the Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 44–50.

Poon, H., and Domingos, P. 2011. Sum-product networks: A new deep architecture. In *Proceedings of the Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence*, 337–346. AUAI Press.

Rapp, R. 2003. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, 315–322.

Salton, G. 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Steinbrück, A. 2013. Conceptual blending for the visual domain. Master's thesis, University of Amsterdam.

Stevenson, L. F. 2003. Twelve conceptions of imagination. *British Journal of Aesthetics* 43(3):238–59.

Todd, P. M. 1992. A connectionist system for exploring melody space. In *Proceedings of the International Computer Music Conference*, 65–68. International Computer Music Association.

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37:141–188.

Turney, P. D. 2006. Similarity of semantic relations. *Computational Linguistics* 32(3):379–416.

Veale, T. 2012. From conceptual mash-ups to bad-ass blends: A robust computational model of conceptual blending. In *Proceedings of the 3rd International Conference on Computational Creativity*, 1–8.

Vygotsky, L. 2004. Imagination and Creativity in Childhood. *Journal of Russian and East European Psychology* 42(1):7–97.

Zhu, J., and Harrell, D. F. 2008. Daydreaming with intention: Scalable blending-based imagining and agency in generative interactive narrative. In *AAAI Spring Symposium: Creative Intelligent Systems*, volume 156.

# Preconceptual Creativity

**Tapio Takala**
Department of Computer Science
Aalto University, Finland
tapio.takala@aalto.fi

**Abstract**

Creativity, whether seen in personal or historical scope, is always relative, subject to the contextual expectations of an observer. From the point of view of a creative agent, such expectations can be seen as soft constraints that must be violated in order to be deemed as creative. In the present work, learned conventions are modeled as emergent activity clusters (pre-concepts) in a self-organizing memory. That is used as a framework to model such phenomena as stereotypical categorization and mental inertia which restrain the mind when searching for new solutions. Using the kinematics of a robotic hand as an example, the models' dynamic behavior demonstrates primitive creativity without symbolic reasoning. The model suggests cognitive mechanisms that potentially explain how expectations are formed and under which conditions an agent is able to break out of them and surprise itself.

## Creativity is in the Eye of Beholder

Creativity is a concept that defies exact definition. The commonly accepted view that creativity is a process resulting in novel and useful products (Mumford 2003) appears to be loose, because in the strict sense even a slightest modification would make the product novel. Another often cited definition is by Newell et al. (1959) who generously view it as a problem-solving process presenting one or more of the following: novelty and value, unconventional thinking, high motivation, and ill-defined problems. They continue by admitting that no more specific criteria can be set for separating creative from non-creative thought processes.

Surprise, more or less as a synonym of unconventional or unexpected, is often considered a necessary condition for creativity (e.g. Boden 1990). However, it may be difficult to distinguish unconventional from mere novelty, as it depends on the observers' subjective experience and conventions. Moreover, novelty is a moving target: once an invention is made it becomes legacy – unless it is forgotten and may be reinvented. Like Grace and Maher (2014), we conclude that creativity is in the eye of beholder, and cannot be defined objectively.

To get a grasp of the relative nature of creativity we adapt the generate-and-verify model by Newell et al. (1959) into variable scopes (Fig.1). The products of a generator (G) passing the evaluation (E) on one level are used as input to evaluation on the next level. A person using computer as a generator ($G_p$) may find designs passing her evaluation criteria ($E_p$), but while showing these to others she (together with her computer) acts as a generator ($G_h$) for the society where others collectively act as evaluators ($E_h$).

On the societal level creativity appears to be a statistical concept formed by opinions of the population under study. Czickszenmihaly (1997) studied individuals ($G_h$) with a reputation of being creative. Maher et al. (2013) studied the evaluation ($E_h$) with a temporal regression model of car designs, where outliers have higher potential for surprise and creativity.

In this paper we concentrate on the personal level (P-creativity), trying to computationally model some of the phenomena happening in a person's mind when a creative moment is encountered. In this respect the generative process is not in our focus. Although various control strategies (analogy, negation, metaphors, etc.) can make it more efficient and interesting, it may as well be a black box. Essential for creativity is the evaluation process, which recognizes value and novelty in products of the generator. It becomes surprised if something unexpected is produced, i.e. if its expectations are violated.

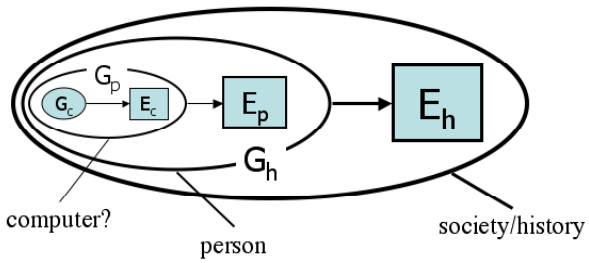



*Figure 1. Context defines the expectations (E) against which the creativity of a generative process (G) is evaluated (from Takala, 2005).*

What are the expectations then? They can be understood as constraints on the product (or process): what it should or is assumed to be (or how it is assumed to be done). They may be hard (defining the domain), such as laws of nature and logic or explicit rules of a game, but they can also be

soft (acquired) constraints: habits, conventions, manners, fashion, social norms, political correctness, etc. These soft constraints are contextual and subject to consideration, applying in one situation but irrelevant in another. But they can be very hard in practice if based on psychological repression. This may serve as an interpretation of Boden's expression that creativity produces "previously impossible ideas". An *idée fixe*, or design fixation (Jansson and Smith 1991) may be the most common obstacle hindering creativity. Such soft constraints form the "box", out of which we are supposed to take a leap.

What makes creativity valuable is that it is a constructive, sense-making act, not just anarchy that randomly defies any rules without a purpose. The new act must in some (novel) way be regular and repeatable. Creativity is search for a constructive and consistent solution assuming some constraints but neglecting or modifying others. By and large, *creativity is management of constraints* for finding a resolution of conflicts among them.

Different degrees of creativity can be identified according to the level of abstraction, or cognitive complexity: (1) Most trivial, though subjectively surprising, is the case when a solution is already known but happens not to be in the current scope of attention: "It just didn't come to my mind". (2) Some effort is required if the solution is not familiar as such but is potentially reachable by known methods or rules. Then essential is the selection of right starting points and methods to proceed with, while neglecting the obvious ones that may distract the process. An example of this is the need to backtrack in order to avoid an obstacle instead of stubbornly pushing straight towards a goal. (3) Yet a higher level comes if the solution is potentially reachable within the hard constraints, but requires constructive actions on the metalevel, i.e. new rules or methods. (4) Finally, even if the product is actually not realizable, we may still act creatively by imagination, neglecting the physical constraints.

The first two degrees, interpreting unexpectedness as changes in the scope of attention (relaxing soft constraints and that way releasing latent possibilities), are demonstrated below using a self-organizing memory as a model. The higher levels, requiring symbolic rules to be changed, are out of the scope of this paper. So is the sometimes required property that creativity should reflect itself, consciously recognizing that something novel and valuable has been formed.

## On Representations

What can be done (consciously acted on) in problem solving, depends on its conceptual representation. This is an important research issue for cognitive science. The main bulk of AI research concentrates on the symbolic level, dealing with logic, language and inference rules. Another end is the subsymbolic sensory area, dealing with neural networks, associative memory and statistical inference. The well-known frame problem, or symbol grounding, calls for connections between the two. In the present work, we are not trying to fill the gap fully, but approach it from bottom up, demonstrating how primitive conceptual representations possibly form from the embodied information.

As the enaction theory (Stewart et al 2011, Rosch et al. 1992) assumes, regularities of the world are learned by receiving repeated stimuli and doing explorative actions. Conditioning and mimicking are two basic psychological principles facilitating this. Later, abstractions of experiences form as subsymbolic concepts. They facilitate more efficient behavior as perceptions are immediately categorized into known classes that may trigger preprogrammed reactions.

Such predefined reactions are of advantage in the world where things are quite predictable. A repeatedly adequate behavior gradually becomes the expected, a rule to be followed. Novel reactions are necessary only if the conditions change – as the proverb says: "necessity is the mother of invention". From evolutionary perspective, however, it may also be of advantage to try out novelties even without a reason, to become prepared for changes. Such tendency is called curiosity, or creative personality.

In neural networks, the sensory information is modeled statistically as conditional distributions and associations. Connecting this to the higher cognitive processes has long been a challenge. Gärdenfors (2000) suggests *conceptual spaces* as a potential bridge between sensory and symbolic levels, a theory of concept formation on supersensory but subsymbolic level. The idea is to describe objects with their properties that act as dimensions of a geometric (metric or topological) space. Individual objects are represented as points in this space, and their generalized conceptual representations as (convex) areas. Inspired by prototype theory (Rosch 1973) Gärdenfors suggests that natural categories may be represented as a Voronoi tessellation around central points representing stereotypical prototypes. This way the extensional (set of experienced samples) is converted into a more efficient intensional (set of constraints) representation.

In this paper, a somewhat similar framework is built, though not relying on a geometric feature space like Gärdenfors (2000) and Chella et al. (2014), but letting the neural cells of a self-organizing network to serve as representative samples of the sensory input. Concepts are not formed explicitly but just as (dynamic) clusters of similar cells. Thus we call it *preconceptual*, resembling the development stage of mind before actual conceptual thinking, in which sensorimotor activity predominates. Pylyshin (2001) uses the term in a compatible manner to describe situated vision, referring to objects that are identified but not defined by their properties. The idea also closely relates to 'proto-symbols' by Brooks and Stein (1994), who use the term for patterns of behavior that represent generalizations but appear rather as signals than formal symbols. Creativity is then demonstrated in primitive form, i.e. problem solving and conflict management using implicit concepts without symbols (Brooks 1991).

## Implementation with Self-Organizing Map

The computational framework we use is based on the Self-Organizing Map (SOM) by Kohonen (2001). It is a widely used clustering device in pattern recognition and data analysis. As a biologically motivated neural network it is an interesting model for cognitive science. It has been suggested by Gärdenfors (2000) as a means of implementing conceptual spaces, though his approach is rather programmatic than an actual implementation.

The SOM is a neural network consisting of an array of cells connected to a vector of input values (Fig.2). The connection weights $w_{ij}$ of a cell are initially random but are changed as follows: Given an input vector $\mathbf{x}$, the cell with best matching weight vector $\boldsymbol{w_j}$ is selected, and its weights are tuned towards the input values. A similar tuning is also done in its neighbor cells.

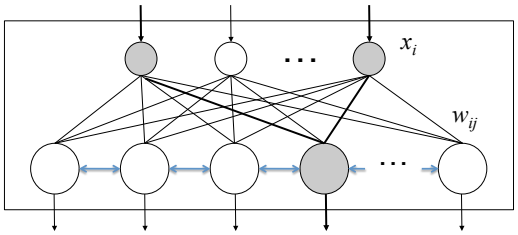



*Fig. 2: The principle of SOM. Input vector $\boldsymbol{X}$ is compared with weight vectors $\boldsymbol{W_j}$ of the cells. The best matching unit is selected and its weight vector tuned towards the input in the training phase. As associative memory, SOM returns $\boldsymbol{W_j}$ as output in response to partial input (an example: active elements emphasized).*

With a large number of input samples, the network organizes itself by unsupervised machine learning instead of using explicitly given concepts. Effectively it builds a model of the training input's statistical distribution, such that each cell represents a collection (a vector) of associated input values, and the number of cells with similar values reflects the density of those value combinations in the input. Usually SOM is implemented with low-dimensional topology (typically a regular 2-D array), and becomes folded if applied to higher dimensional input. An example is given in Figure 3.

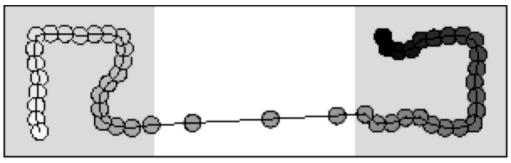


*Fig. 3: One-dimensional SOM (chain of cells) trained with data from a 2-D distribution concentrated in the grey areas (cells are visualized in the input space in locations of their learned values).*

In pattern recognition SOM is widely used as a classification device. It tells efficiently if a given input vector belongs to one category or another. This helps in data compression as complex input vectors can be quantized and represented with a smaller number of dimensions.

In SOM, similar cells emerge close to each other resulting in associations between a cell and its neighborhood. If there are concentrations in the input distribution, similar cells form clusters separated by dissimilar boundaries (Figs. 4a).

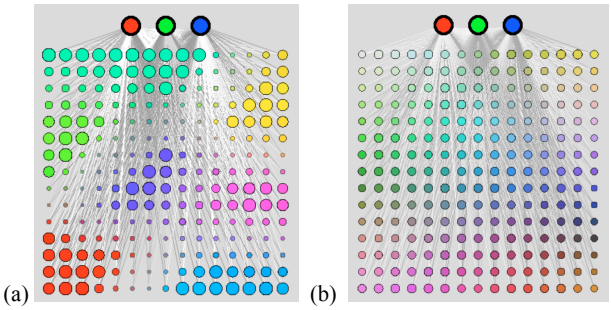



*Fig. 4: A two-dimensional SOM trained with RGB values of (a) discrete colors (b) flat color spectrum. Cell color shows its learned values, cell size indicates similarity with its neighbors.*

As each cell represents a vector of correlated input values, the SOM can act as an associative memory. A partial input (i.e. the values given for some inputs, and the rest undefined) as a stimulus activates the cells according to their similarity with the defined inputs. As result we get for each cell the probability of its value vector to become the output. Then we select the cell best matching the partial input, and take its weight vector as output (see Fig.2). Effectively the associative memory would fill in the undefined values by those from a cell selected by highest probability. Practical applications are found in image completion (Kohonen 2001), or information retrieval (Kohonen et al. 2000), for example.

The separable clusters (as in Fig. 4a) can be interpreted as primitive concept formation ("preconcepts"). When an input activates some cells, their similar neighbors are activated as well in the cluster. Then if the cluster were labeled with semantic information (such as color name), the input would be identified with that. The behavior resembles categorical perception in psychology (Goldstone and Hendrickson 2010) in the sense that the classification of any input within a cluster would get strong support by a group of cells, whereas an input falling to an area boundary would be in "unknown" territory where classification is unreliable. This coincides with the phenomenon in categorical perception that stimuli near category boundaries are more difficult to identify than within categories.

It is not clear if the human perceptual categories are independent of symbolic concepts, nor if they are presented by stereotypical prototypes or area boundaries. We hypothesize that it is possible to form concepts without higher level semantics, if such identifiable areas emerge. Such

does not happen if the input distribution is flat without statistical foci (Fig. 4b).

## A Case Study

In this section, we show how an associative SOM can be used to solve the control problem of a kinematic hand, and demonstrate preconceptual creative behavior in that context.

### Setting the Scene

An articulated kinematic hand mechanism consists of a set of links connected at rotational joints to make a chain. In our case there are two such links (Fig. 5). Using the two joint angles (α and β) as motoric controls, the hand can reach points in the (*x, y*) plane within an area delimited by its physical constraints (i.e. the allowed ranges of control angles, and possible other geometric obstacles). The hand position can easily be calculated by trigonometry from the angles and lengths of the joined links, whereas the inverse is non-trivial. This inverse kinematics (IK) problem, finding control values for angles, given a target position, is generally a hard problem without analytical solution. A simple solution exists for our case with only two degrees of freedom, but it is still interesting due to its non-linearity (including singularities), physical constraints, and non-uniqueness of the solution: the same point can be reached by left or right handed configuration (negative or positive values of β, respectively).

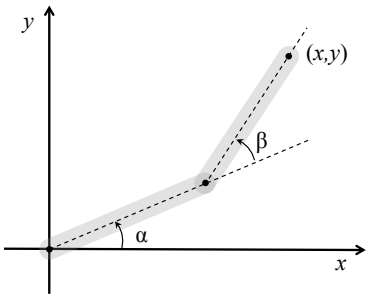
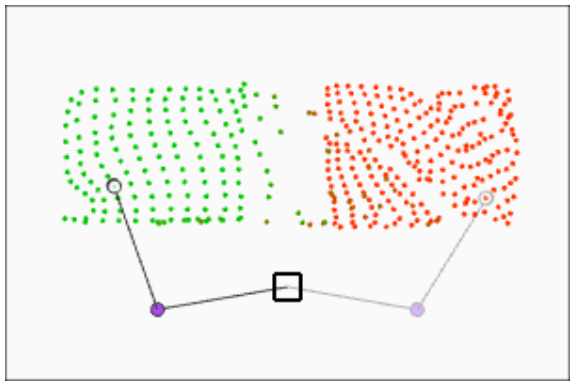



*Fig. 5: Kinematics of a robotic hand*

Among many other techniques, feed-forward neural networks have been proposed to solve the IK problem by training the system with random samples from the configuration space (e.g. Duka 2013). In case the problem is under-constrained (i.e. the robot has redundant degrees of freedom), sampling can be utilized to satisfy additional goals, such as moving in a certain style. Wiley and Hahn (1997) propose building from the given positions a resampled grid that serves as a geometric index, out of which the final angle-target combinations are calculated by interpolation. Our approach is similar to both of these in the sense that a neural network is trained to form a grid-like index, from which candidate starting points are selected for final approach to the target.

Let us assume our humanoid robot has two hands with their physical limits (hard constraints) similar to those of the human left and right hand. Each hand is trained to work in its most natural area (left/right in front of the base) as in Fig. 6a. The system is implemented in one SOM with two inputs for hand position (x, y), two for joint angles (α, β), and one (binary) input for handedness. Then clusters automatically form in SOM corresponding to left and right handed operation (Fig 6b). Their actual shape is random, sometimes bifurcated or consisting of multiple foci, but the areas are clearly identifiable. The clusters are separated by a boundary where the cells are less similar with their neighbors (shown in yellow).

(a)



(b)

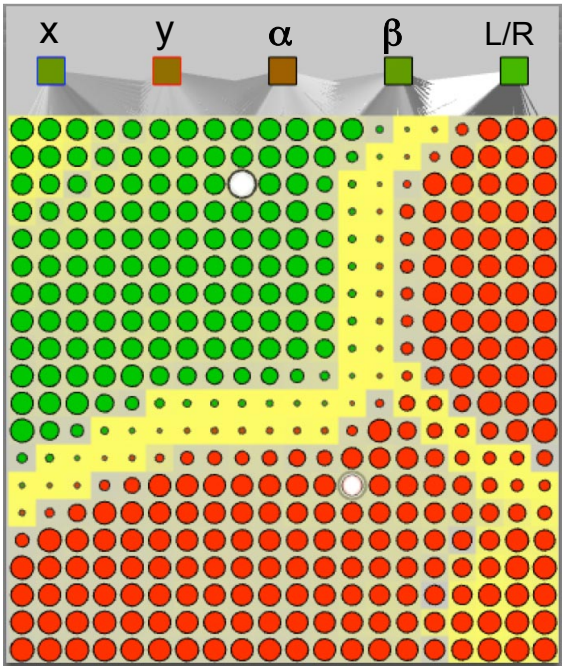



*Fig. 6: Training areas of hands (L=green, R=red) in the experiment. a) in robot space, b) as clusters formed in SOM. Two sample positions shown: white cells in SOM and the corresponding left (solid) and right (shadowed) hand positions.*

The IK problem is solved by association, taking the target's coordinates as partial input, finding the cell(s) that best matches with it, and returning its weight values for the missing inputs (the control angles α and β):

$$f\text{: } (x, y, ?, ?) \longrightarrow (x', y', \alpha, \beta)$$

Although the result as such is not exact, it provides a good starting point for an iterative final approach. The movement direction needed in the iteration phase can be estimated from the cell's neighborhood by differentiation (approximating the Jacobian of parameters). This is a common strategy with actual robots and well grounded by biological action where proprioceptive memory and motor programs (Keele 1968) quickly lead to approximately right position and the final approach is done with the help of sensory feedback. In our implementation this phase is computed explicitly, but the Jacobian differentials could as well be learned by the SOM, if continuous movements instead of random positions were used in the training phase.

Targets within the trained areas are easily reached with the method above, and if the target is not too far out from the trained area, it usually can be reached from the closest starting point by the final iteration (Fig. 7a).

## Acting Creatively

Now let us take a challenge where the simple approach does not work, by setting the target in a place not reachable from the closest point by direct iteration. This may be caused by a limitation of the mechanism itself or happen due to a physical obstacle (such as the box wall in Fig. 7c). Then the final approach gets stuck and we need to find a new starting point.

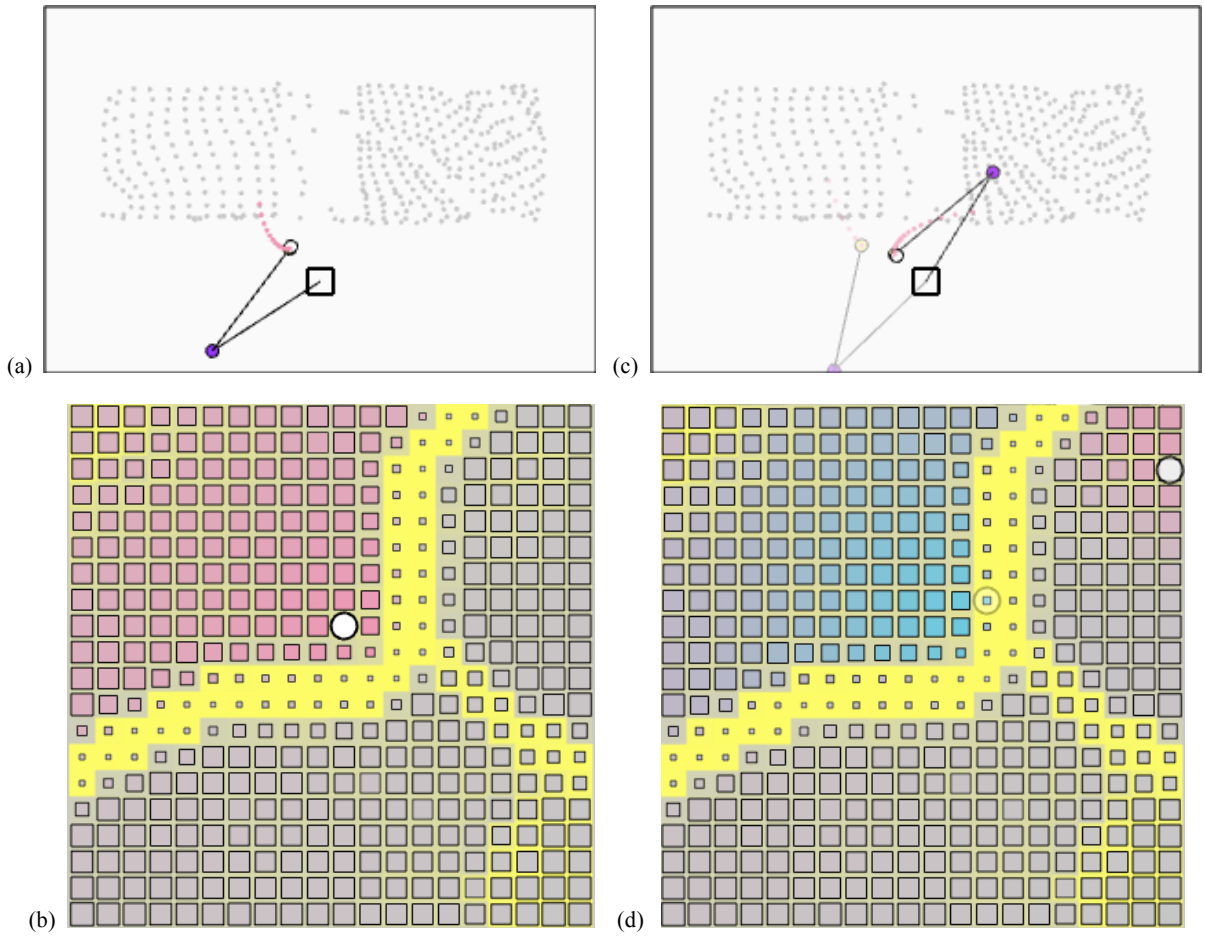



*Fig. 7: Creativity in search for IK solutions. (a) target point reachable with left hand (final iterative approach shown as a sequence of red dots), (b) SOM cell (white circle), found in a recently active cluster (pink), defines the starting point for approach, (c) target appears impossible for the left hand due to the wall obstacle at its "elbow" (shadowed), but a new starting point feasible for the right hand is found (solid), (d) corresponding activity in SOM, where the previous starting point is surrounded by negative feedback effect (blue) due to unsuccessful trials, and the new point gets positive feedback (pink) which propagates to neighboring cells.*

Though in principle any starting point could be considered as a new candidate, a random search is not very effective. Even if a cell's probability to be selected is weighted by its correlation with the input, a random method would mostly suggest candidates near the one which already lead to a dead end. The obvious engineering solution, trying out all candidate points in successive order, is not suitable here because sorting would call for higher level conceptual thinking and a different memory organization. We do not want to give the system any ready-made domain specific heuristics either, but want it to rely on very generic principles. As such an approach we utilize supervised reinforcement learning with a short-term memory (STM).

We implemented a distributed STM as an additional variable in each cell. It modulates the cell's probability to be selected as candidate for a trial. Its value would be increased by positive feedback from a successful case and decreased if the trial fails. Following the self-organization principles, these changes are also propagated to the cell's neighborhood but only among similar cells. To keep the operation dynamic, both positive and negative effects are gradually faded, possibly with different time constants.

The system's behavior now depends on its short-term history, its sensitivity to feedback, and the relative time constants. Let us assume the robot has operated for a while with targets in the left-hand area. Then the cells in the corresponding cluster(s) have been activated a lot, and due to positive feedback their probability to be selected again is high (pink color in Fig. 7a-b). When the target moves to a near but unreachable position (Fig 7c), the same cells continue to be activated as candidates, but a failure to reach the goal from one starting point will make the probability of that cell (and its close neighborhood) low. However, because of recent positive activity, the search will still continue with other cells in the same cluster. Then the further course of action is determined by the system's history and parameters as follows.

If a cluster's temporal activity is high (due to operating long in that area) and fading slower than the effects of negative feedback, the system will continue search within the same cluster despite of being unsuccessful. This corresponds to *mental inertia*, the tendency to keep on temporal preferences, i.e. the agent's expectation that a recently useful concept will continue to be so, an *idée fixe*.

However, if the negative feedback is more persistent and eventually dominates the whole cluster (indicated by blue color in Fig 7d), then a cell in some other cluster (probably one with next best correlation with the target) gets highest probability and will be taken as starting point for a trial. If it does not succeed, negative feedback will make its neighborhood less probable and the search continues somewhere else. Effectively this would implicitly perform an ordered search, though without explicit sorting.

Once a successful case is found (possibly requiring iterative final approach as in Fig. 7c), it will get positive feedback which is diffused to its neighbors in the same cluster, too (pink color in Fig. 7d). If the agent's operation continues with further targets nearby, this neighborhood will provide successful candidates again, and eventually the cluster becomes predominant: a primitive paradigm shift has happened, *heureka!*

## Analysis of system behavior

We can evaluate the system theoretically and get the following qualitative observations, also confirmed by experiments with different parameters and test conditions.

In the above case, the creative leap was required because the left hand was unable to continue operation due to a constraint. Had the system a different history, with the right hand recently used before going to the new target, the new solution would have been obvious because of the predominant right hand: no creative moment, nothing unexpected, although new compared to what had been learned and stored in the long term memory (SOM). This is in alignment with the general observation that mental fluidity is induced by pressures (Hofstadter and Mitchell 1995) and may not happen otherwise.

Sticking with recently used behavior and building expectations is necessary for the system to act creatively, but it is not sufficient alone. Without negative feedback from an unsuccessful trial the system will keep trying the same over and over without getting anywhere.

Without any (positive or negative) feedback the system looses its temporal properties and reacts always the same way in a given situation, governed by the associative memory alone.

An interesting situation is encountered if we neglect the positive feedback but keep the negative. This leads to an "anti-sticking" behavior: once a cell has been used, neither it nor its close neighbors will be used for the next trial, but something loosely associated with the input. As the effect of negative feedback gradually fades away, the system may return to this cell if its association to the input is high, but only temporarily, and then jump to another cell. Overall, this resembles divergent thinking: variable alternatives are tried out, not randomly but guided by associations.

In our case study the robot's handedness was given as an explicit input feature to the SOM. This makes a clear distinction between clusters corresponding to left and right handed operation, respectively. However, this feature appears to be unnecessary, as similar behavior may emerge anyway if there only are two or more separate clusters formed from the distribution of input value combinations.

The ability to act creatively depends on the problem domain and its representation: if there are local optima where one may get stuck, there is a possibility for radical moves – otherwise a too simple route may lead to the solution. In this respect our system can be compared with optimization: Gradient search is a sticky strategy corresponding to the case with positive feedback only. Parallel search methods, such as genetic algorithms and simulated annealing, may lead to unexpected solutions, though in their basic form they have no such concept as surprise. However, the 'temperature' that makes simulated annealing process to look for more random options may well be compared to the negative feedback in our system.

## Discussion

Different degrees of creativity, as mentioned in the introduction, can be demonstrated with our system. The case when a solution is already familiar (or reachable by iteration) but "didn't come to my mind" is modeled if the recent history has built strong temporal preference for a subset of solutions. This manifests itself as the agent's "sticky" tendency to sometimes utilize iterative approach from recently used starting points even if there were a better starting point stored in SOM, but this alternative is in a different cluster.

The more interesting case, target reachable within hard constraints but outside the most obvious trained area, is demonstrated when starting to use the other hand after trying and failing with one (as in Fig. 7). This can be interpreted as transformational creativity on preconceptual level, a change in the predominant cluster (rule) used in the agent's operation. It involves relaxation of soft constraints (giving up accustomed solutions), an essential property of creativity.

Whether this should be called creativity, may be an arguable question. Hristovski et al. (2011) have studied a similar situation of limb movements in the context of boxing. On the one hand, they state that any novel movement that has not been performed previously by an individual can be considered a P-creative act. On the other hand, they note that movement system bistability yields too much predictable behavior to account for creativity. Our case may be interpreted as the latter due to the binary choice of left or right hand in any situation, or the former because the exact hand movement is not predictable. A deeper analysis of the system's dynamics may be needed to take a stance.

Although our model shows qualitative changes in the robot's dynamical behavior, it is missing temporal anticipation, which could be utilized for creative planning of actions. The implementation as such does not support reasoning about an action's consequences that would be needed for goal-oriented behavior and higher-level expectations (Lorini and Falcone 2005). However, similar techniques might be used for learning temporal associations as well, thus making it a platform for further development.

Lorini and Falcone (2005) used formal logic to describe expectations and surprise in symbolic domain. At the other end of the scale, specific neural assemblies have been found that correspond to these phenomena in visual cognition (Egner et al. 2009). This suggests that a neural network model may be feasible. Gabora (2010) presents a schematized associative memory where neural cliques are alternatingly recruited for analytic and associative modes of thought, which is supposed to be essential for creativity. The model does not consider expectations and surprise, nor computational implementation, but the activation function of neurons may be comparable to our feedback mechanism.

The Copycat system (Hofstadter and Mitchell 1994) has a somewhat similar feedback mechanism as our STM. Its global 'temperature' and the 'unhappiness' of objects serve as measures controlling the random choices that facilitate unexpected behavior. The main differences are that it works on textual objects instead of continuous signals, and its architecture is based on a crowd of heterogeneous codelets instead of neural networks. The latter feature makes it more reminiscent to Brooks' robots.

Relaxation of hard constraints, e.g. leaving the physical space and thinking in another context by analogy or metaphor, would call for higher level conceptual models than neural networks, and is out of the scope of this paper. The same applies to reflective thinking. Our poor system itself does not recognize creativity, though it may be possible to detect it from the abrupt changes happening in the STM values during a creative leap.

Had the system a measure of cumulative effort used before a successful trial, or about the time spent without a goal at all, it could model the emotional frustration and boredom that are supposed to control creative behavior on a higher level. In previous work (Takala 2005) these were used to control the recruitment of alternative methods to solve given problems. Combining the mechanisms with the present work may result in interesting behaviors.

Our general approach follows much that suggested in robotics (Brooks 1991, Brooks and Stein 1994). Although the current implementation is based on a single neural network, and a multilevel hierarchical organization of several SOMs may be possible, a more heterogeneous architecture may also be due.

## Conclusion

This work emphasizes the contextual nature of creativity, culminating to expectations and their role as soft constraints that must be violated in order to find novel and surprising solutions to problems. Concentrating on the preconceptual level of cognition, it contributes to an area rarely touched in previous works.

A computational model is presented that implements a primitive form of creativity, which may serve as a basis for further development. Autonomous formation of conceptual spaces is demonstrated with the self-organizing memory, and a learning mechanism proposed that simulates the temporary preferences typical in idea fixations. Though our example case is about kinematics, the model is domain independent and may be applied in many different areas.

The creativity model proposed in this paper is based on various ideas that are not novel as such but presented in multiple previous works. The main contribution appears to be the implementation where a self-organizing neural network is combined with control mechanisms usually applied on the symbolic level. Our system is not using predefined heuristics or encoded algorithms but applies generic learning principles to form (pre)concepts, on which the feedback mechanism operates.

A theoretical conclusion is that creativity cannot happen just anywhere, but requires certain conditions: In order to be surprising, the situation should involve expectations, or temporary preferences, that are violated in a creative act. If

the system acts in a continuous parametric domain, such as movement, the setting (or its representation) should be non-monotonic, such that the system may get stuck in a local optimum. Yet another condition, though mostly overlooked in the present work, is motivation. If the problems to be solved are given from outside, the system acts in a slave mode, whereas a truly creative mind would be curious and willing to set problems, not just to solve them.

An immediate future work is to study the proposed mechanism in more complicated cases, such as a real robot, taking into account physical continuity of movement and not only static positions. Another extension is to facilitate explorative creativity by letting the robot move randomly around and learn continuously. Long term goals include developing the proposed approach towards higher-level cognition and conceptual thinking, including analogical reasoning and emotional self-control.

## Acknowledgements

Thanks to the anonymous reviewers for their comments on the manuscript and pointers to additional sources. This work has been made possible in part by the sabbatical system of Aalto University.

## References

Boden, M. (1990). *The Creative Mind: Myths and Mechanisms*, Weidenfeld & Nicholson (2nd ed. Routledge, 2004).

Brooks, R. (1991). Intelligence without representation. *Artificial intelligence, 47*(1), 139-159.

Brooks, R., & Stein, L. (1994). Building brains for bodies. *Autonomous Robots, 1*(1), 7-25.

Chella, A., Gaglio, S., Oliveri, G., Augello, A., & Pilato, G. (2014). Creativity in Conceptual Spaces. In *5th Intl. Conf. on Computational Creativity*, 306-314.

Csikszentmihalyi, M. (1997). *Creativity: flow and the psychology of discovery and invention*. Harper Perennial.

Duka, A. V. (2014). Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *Procedia Technology*, *12*, 20-27.

Egner, T., Monti, J. & Summerfield, C. (2010). Expectation and Surprise Determine Neural Population Responses in the Ventral Visual Stream. *The Journal of Neuroscience, 30(49)*, 16601–16608.

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), 1-13.

Gärdenfors, P. (2004). *Conceptual spaces: The geometry of thought*. MIT press.

Goldstone, R. L., & Hendrickson, A. T. (2010). Categorical perception. *Wiley Interdisciplinary Reviews: Cognitive Science*, *1*(1), 69-78.

Grace, K., & Maher, M. L. (2014). What to expect when you're expecting: the role of unexpectedness in computationally evaluating creativity. In *4th Intl. Conf. on Computational Creativity*, 120-128.

Hofstadter, D., & Mitchell, M. (1994). The copycat project: A model of mental fluidity and analogy-making. *Advances in connectionist and neural computation theory, 2*(31-112), 29-30.

Hristovski, R., Davids, K., Araujo, D., & Passos, P. (2011). Constraints-induced emergence of functional novelty in complex neurobiological systems: a basis for creativity in sport. *Nonlinear Dynamics, Psychology and Life Sciences*, 15(2), 175-206.

Jansson, D. G., & Smith, S. M. (1991). Design fixation. *Design studies*, *12*(1), 3-11.

Keele, S. W. (1968). Movement control in skilled motor performance. *Psychological bulletin*, *70*(6p1), 387-403.

Kohonen, T. (2001). *Self-organizing maps* (Vol. 30). Springer Science & Business Media.

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., & Saarela, A. (2000). Self organization of a massive document collection. *Neural Networks, IEEE Transactions on*, *11*(3), 574-585.

Lorini, E., & Falcone, R. (2005). Modeling expectations in cognitive agents. In *AAAI 2005 Fall Symposium: From Reactive to Anticipatory Cognitive Embodied Systems*. 114-121.

Maher, M. L., Brady, K., & Fisher, D. (2013). Computational models of surprise in evaluating creative design. In *4th Intl. Conf. on Computational Creativity*, 147-151.

Mumford, M. D. (2003). Where have we been, where are we going? Taking stock in creativity research. *Creativity Research Journal*, *15*(2-3), 107-120.

Newell, A., Shaw, C., & Simon, H. A. (1959). *The processes of creative thinking*. Santa Monica, CA: Rand Corporation.

Pylyshyn, Z. (2001). Visual indexes, preconceptual objects, and situated vision. *Cognition* 80(1), 127-158.

Rosch, E. (1973). Natural categories. *Cognitive psychology*, 4(3), 328-350.

Rosch, E., Thompson, E., & Varela, F. (1992). *The embodied mind: Cognitive science and human experience*. MIT press.

Stewart, J. R., Gapenne, O., & Di Paolo, E. A. (Eds.). (2010). *Enaction: Toward a new paradigm for cognitive science*. MIT Press.

Takala, T. (2005). Creativity as disruptive adaptation – a computational case study. In: Gero and Maher (eds.), *Computational and Cognitive Models of Creative Design VI*, Univ. of Sydney, 95-110.

Wiley, D. J., & Hahn, J. K. (1997). Interpolation synthesis of articulated figure motion. *Computer Graphics and Applications, IEEE*, *17*(6), 39-45.

# Specific curiosity as a cause and consequence of transformational creativity

**Kazjon Grace & Mary Lou Maher**
Department of Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC 28203 USA
{k.grace,m.maher}@uncc.edu

**Abstract**

This paper describes a framework by which creative systems can intentionally exhibit transformational creativity. Intentions are derived from surprising events in a process based on specific curiosity. We argue that autonomy of intent is achieved when a creative system directs its generative processes based on knowledge learnt from within its creative domain, and develop a framework to elaborate this behaviour. The framework describes ways that transformation of the creative domain can arise: from learning, from a serendipitous situation, and as a result of intentional exploration. Examples of each of these kinds of transformation are then illustrated through examples in the domain of recipes.

## Introduction

Significant effort has been devoted to developing computational models that can recognise creative artefacts, on the assumption such a capability could be used to generate creative artefacts if paired with an appropriate search algorithm. However, generate-and-test creative systems lack any kind of *autonomous intent*: they never decide to make a green artefact, or a loud one, or a happy one, unless such qualities are built into their externally-provided objective function. As classically formulated, a search function does not distinguish two points within its space in any way but by the objective, and thus has no intent that can be defined with the representations that define that space, only in how the resulting artefacts perform. Search functions that can modify their goals while searching (Gebser, Kaufmann, and Schaub 2009), or that search based on specific past experiences (Cully, Clune, and Mouret 2014) do exist, but, from a computational creativity perspective, there remains an unanswered question: under what conditions should a system decide to modify its search?

At first this lack of autonomous intentions in our systems' search processes may fail to seem problematic: we are not constrained by cognitive plausibility. There is no inherent reason why intentionality, while clearly a quality of human creators, should be required in their digital analogue. Our goal is systems which produce output that would be considered creative, regardless of the processes involved. On closer inspection, however, autonomy of intention may not be so easily discarded from creativity. Intent is intrinsically tied to definitions of art and creativity (Dewey 2005), where the debated questions concern not whether an artefact's creator had intent, but whether that intent should be privileged over observers' interpretations (Best 1981). Intention is seen among human creators as critical both to the production and consumption of creative artefacts – evidence that argues for its role in appreciative as well as generative computational processes.

Autonomy of intent also provides critical information for use in framing. A creative system's ability to construct framing narratives for its work – considered critical to any computationally creative construct (Charnley, Pease, and Colton 2012) – stems from its ability to provide *justification* for creative decisions. Without autonomy of intent these justifications can only be driven by external objectives (e.g. "I wanted to make the artefact seem brighter"), not intrinsic motivations (e.g. "I was exploring how colour influenced brightness"). Human creators make the decision to explore a particular set of concepts, and follow that exploration to its resolution by way of creative expression. Framing, as the channel by which a creative system can convince its audience of its creative autonomy, should explain such explorations. Previous models of intent in framing have been based on information extrinsic to the creative domain, such as the day's top news stories (Krzeczkowska et al. 2010), but we argue that without learning how to connect such external knowledge to the creative domain (e.g. through analogy), then such intent cannot be autonomous.

How, then, can a creative system derive intent from its knowledge about the creative domain? On what basis should it transform its inspiring set and own past creations into contextual constraints on its search process? For one possible answer we turn to cognitive studies of how human designers think during the process of designing, and how their search for a creative solution affects itself. Human designers do not sequentially analyse a problem, synthesise solutions to it and then evaluate those solutions, but instead switch between those processes iteratively (Schön 1983), finding new problems as frequently as they find new solutions (Weisberg 1993). This co-evolution of problem-framing alongside problem-solving becomes more evident in expert designers (Cross 2004), and – more critically for our purposes – has been shown to produce more valuable output (Getzels and Csikszentmihalyi 1976). A cognitive protocol analysis of

sketching architects found that not only did they regularly unexpected discover features in their own drawings, but that those discoveries often led to reformulation of the design task (Suwa, Gero, and Purcell 1999). These reformulations led in turn to more unexpected discoveries, evidence that this cycle of intentionality and exploration is beneficial, if not central, to human creativity. We seek to capture this cycle in the computational model presented in this paper.

We propose that the inspiration for a computational model of intentional creativity can come from the iterative process of defining the creative task and solving it in parallel. We propose that intentions are not created *de novo*, but that they arise from a drive to explore what the system has observed but not understood, both from its own output and that of other creators. The catalyst for this exploratory behaviour is unexpectedness: a creator being surprised by an artefact, and forming the intention to explore some part of the design space in return. We refer to this as a kind of *specific curiosity*, after the distinction between specific and diversive curiosity first articulated by Berlyne (1966).

We frame our model for specific curiosity as an extension of Wiggins (2006) framework for describing exploratory and transformational creativity. With that symbolic representation we can then describe how transformational creativity leads to surprise, and how surprise can in turn lead to further creativity.

## Transformational creativity, surprise and their effects on behaviour

This paper describes a model of autonomous intent in creative systems, drawing on theories of evaluating creativity, psychological studies of curiosity and cognitive studies of how designers respond to unexpected discoveries. We introduce each of those literatures here.

### Three long-lost cousins: novelty, transformational creativity and surprise

Novelty (Newell, Shaw, and Simon 1959; Saunders and Gero 2001), surprise (Macedo and Cardoso 2001; Grace et al. 2014) and domain transformation (Boden 2003; Wiggins 2006) are three core ideas around which the debate on how to computationally recognise creative artefacts has revolved. In Grace and Maher (2014) we outlined how each of those three could be connected to the notion of unexpectedness, establishing one possible way to compare them in a common language.

Novelty was, to the authors' knowledge, first floated alongside value by Newell, Shaw and Simon (1959), forming the closest thing to a broadly-accepted definition for creativity that we have today. Novelty and value are proposed as necessary and complementary aspects of creativity: a solely valuable artefact is merely good, while a solely novel artefact is merely weird. Novelty is typically conceptualised as difference from that which is known (Sternberg and Lubart 1999), and usually operationalised by a distance measure between a new observation and past experiences. An alternate view of novelty is based on the degree to which observing an artefact helps an agent to understand the world (Schmidhuber 2010), proponents of which criticise the distance-based approach as attributing overly high novelty to noise.

Boden (2003) proposed another solution to the problem of distinguishing meaningful novelty from noise by focusing on impact. *Transformational* creativity is based on the degree to which an artefact changes the creative domain to which it belongs. This is suggested by Boden to be a more significant form of creativity than the combination of "mere" novelty and value, which she considers the result of *exploratory* creativity. Wiggins (2006) formalises Boden's definition of transformational creativity and provides a general description of a creative system that is capable of it, although he questions Boden's strict hierarchical superiority of transformation over exploration.

The authors have previously proposed unexpectedness and surprise as an alternative formulation of novelty (Grace et al. 2014), although we are far from the first to do so (Macedo and Cardoso 2001). Unexpectedness is the degree to which observing an artefact violates (i.e. opposes) an agent's confident predictions about the world. The flexibility of this approach is in the source of predictions, which may be relationships within the artefacts, trends derived from the domain's history, or other sources of knowledge. Novelty can be described from this perspective as a form of unexpectedness based on the predicting that the domain will continue as it has in the past. Surprise is an affective response to unexpectedness: unexpected artefacts induce surprise in their observers. Transformational creativity can be described as a quantification of surprise based on how much a new artefact changed domain knowledge. This connection was described in Baldi and Itti (2010), who used an information theoretic perspective to connect measuring surprise by (un-)likelihood to measuring it by impact on knowledge.

Throughout this paper we adopt the viewpoint that these three notions are intimately connected, constituting complementary perspectives on how a creative artefact can be meaningfully different from those that preceded it. We argue that the evaluative processes of creative systems should possess the ability to detect all of the above aspects of meaningful difference, and that any one of them – in conjunction with value – can indicate creativity.

### Curiosity and the pursuit of novelty

Curiosity is an overloaded term in psychology, referring both to a trait possessed by different people to different degrees, as well as to motivating state that drives its experiencers to seek novel stimuli (Berlyne 1966). The latter definition, curiosity as a state, has been proposed as a motivator for computational creative systems (Saunders and Gero 2001; Merrick and Maher 2009), based on the principle that novelty-seeking (alone or alongside value) will drive exploration towards creative solutions.

Berlyne distinguishes state-curiosity along two axes: perceptual vs epistemic and specific vs diversive. Perceptual curiosity is the drive towards novel sensory stimuli, and has been observed in a variety of animals of different cognitive capabilities. Epistemic curiosity is the drive to acquire novel knowledge This conceptual curiosity can be modelled by

systems that learn a conceptual space and measure novelty within it, rather than measuring between artefacts at the level of sensory input (Saunders and Gero 2001). The distinction between creativity at the sensory and knowledge-levels has been drawn within computational creativity by Smith and Mateas (2011), who refer to the latter as "rational curiosity".

The specific/diversive division has received less attention in computational creativity. Specific curiosity is the search for observations that explain or elaborate a particular goal concept. Diversive curiosity, on which most computational models of curiosity have focussed, is the search for new information without any specific targets. While the search for a specific concept can be modelled by search, the challenge is how to trigger specific curiosity: when and why should a creative system become specifically curious? This is related to the broader issue of creative autonomy (Jennings 2010; Saunders 2011). In this paper we develop a model of specific curiosity that uses surprise as a way to address this challenge.

### How surprises affect designing

Cognitive studies of human creators – particularly in the field of design – have shown that surprise significantly impacts the creative process. Designing has been described as a "reflective conversation with the medium" (Schön 1983), meaning that designers iteratively synthesise new additions to their emerging design and then reflect on their effects. Expressing creative artefacts through rough yet external representations – usually referred to as sketches in the case of human designers – is a critical component of the creative process as it allows designers to observe changes they did not consciously make (Schon and Wiggins 1992; Goldschmidt 1991). Through this externalisation a designer may perceive an emergent shape, discover a new relationship between components, or construct an analogy to past designs. Several computational creativity systems have adopted this cyclical reflective approach in whole or in part, including the search-bias transformation in DeLeNoX (Liapis et al. 2013), the interpretation-driven mapping of Idiom (Grace, Gero, and Saunders 2015) and the expectation-based reinterpretation of Kelly and Gero (Kelly and Gero 2014).

This iterative process of "seeing" (perceiving an emerging design) and "moving" (making a change to it) allows designers to read more off a sketch than they originally put there (Schon and Wiggins 1992). Though the term has since been corrupted beyond recognition, this was the original meaning of *design thinking*: an iterative, reflective, solutions-focussed strategy as opposed to a step-by-step, analytical problem-focused one (Lawson 2006). In a "think aloud" cognitive protocol study where architects were observed designing, unexpected discoveries were bi-directionally causally connected to reformulation of the design goals, i.e.: surprises led to transformation of the problem, and transformation of the problem led to surprises (Suwa, Gero, and Purcell 1999). These results with human creators suggest that surprise-triggered specific curiosity might be useful for encouraging transformative creativity in artificial creative systems. In the remainder of this paper we develop a framework for how that behaviour could be operationalised.

## Unexpectedness-triggered specific curiosity: A model of transformation-seeking behaviour

We adopt the creative systems framework from (Wiggins 2006) to describe our model of unexpectedness and specific curiosity. Wiggins' framework describes a creative system in terms of a search process that traverses a conceptual space to generate artefacts, coupled with a metacognitive search process that traverses the space of all possible conceptual spaces. The resulting system is capable of both exploratory and transformational creativity, with the latter represented as exploration at the meta-level. The following symbols define the core of the framework, although readers are encouraged to familiarise themselves with the original, which affords each definition far greater depth:

- $\mathscr{U}$ is the *universe*, the space of all possible distinct concepts that make up all possible representations of artefacts in the current creative domain.
- $\mathscr{L}$ is the *ruleset language*, the set of all possible rules that act on concepts the creative system can construct.
- $[\![.]\!]$ is the *definition interpreter* that takes a subset of $\mathscr{L}$ and acts on a set of concepts, yielding real numbers in [0,1]. This is used to apply a rule set to a set of concepts, assigning a value to each.
- $\mathscr{R} \subseteq \mathscr{L}$ is a *constraint ruleset*, by which the system defines the scope of the conceptual space (within $\mathscr{U}$).
- $\mathscr{C}$ is a *conceptual space* is the current subset of $\mathscr{U}$ permitted by $\mathscr{R}$. i.e., $\mathscr{C} = [\![\mathscr{R}]\!](\mathscr{U})$.
- $\mathscr{T} \subseteq \mathscr{L}$ is a *traversal ruleset*, by which the system explores $\mathscr{C}$.
- $\mathscr{E} \subseteq \mathscr{L}$ is an *evaluation ruleset*, by which the system evaluates proposed concepts.
- $c_{in}$ is the *input set*, a totally ordered subset of $\mathscr{U}$ that reflects the list of artefacts known to the system, in the order of the system's observation of them.
- $c_{out}$ is the *output set*, a totally ordered subset of $\mathscr{U}$ that reflects the output of the creative system after a particular generative iteration.
- $\langle\!\langle .,.,. \rangle\!\rangle$ is the *generation interpreter* that takes three subsets of $\mathscr{L}$, the rules that define the conceptual space $\mathscr{R}$, the rules that define how to traverse that space $\mathscr{T}$, and the rules that assign value to members of that space, $\mathscr{E}$ and acts on the set of all previously observed artefacts to generate a new set of artefacts. i.e., $c_{out} = \langle\!\langle \mathscr{R},\mathscr{T},\mathscr{E} \rangle\!\rangle (c_{in})$.
- $\mathscr{L}_{\mathscr{L}}$ is the *meta-level ruleset language*, the set of all possible rules that act on rulesets (i.e., on $\mathscr{L}$) the creative system can construct.
- $\mathscr{R}_{\mathscr{L}} \subseteq \mathscr{L}_{\mathscr{L}}$ is a *meta-level constraint ruleset*, by which the system defines the scope of the meta-conceptual space of possible rules that can be part of $\mathscr{L}$.

$\mathscr{T}_{\mathscr{L}} \subseteq \mathscr{L}_{\mathscr{L}}$ is a *meta-level traversal ruleset*, by which the system explores the space of possible rules for $\mathscr{L}$.

$\mathscr{E}_{\mathscr{L}} \subseteq \mathscr{L}_{\mathscr{L}}$ is a *meta-level evaluation ruleset*, by which the system evaluates proposed rulesets for their ability to generate valuable concepts.

The differentiation of $\mathscr{R}$, the rules defining the conceptual space, from $\mathscr{T}$, the rules defining the search process which acts on that space, is a significant addition to Boden's notion of transformational creativity. With this distinction Wiggins can describe two kinds of transformational creativity: $\mathscr{R}$-transformation of the space of possible concepts, and $\mathscr{T}$-transformation of the search process for generating new concepts. $\mathscr{R}$-transformation, closest to Boden's original conceptualisation of transformative creativity, concerns the redefinition of what a creative system considers possible. $\mathscr{T}$-transformation concerns the redefinition of how a creative system creates.

The definition of a creative domain – as captured by Wiggins' $\mathscr{R}$ – is a socially grounded construct. While it is useful from the perspective of defining transformation across a creative domain to think of that construct as stable across all members of a society, in practice this knowledge must be learnt by each member. In Boden's original model, the definition of the creative domain is agreed amongst all participants, and this knowledge is not expected to be constructed through exposure to the domain. Wiggins hints at the social nature of $\mathscr{R}$, but does not distinguish individual and societal transformation of the conceptual space. To model the influence of artefacts created by others on a system's behaviour, we must capture this distinction: we will use $\mathscr{R}$ to refer to an individual creative system's definition of the space, but one could imagine a broader, socially grounded *historical-*$\mathscr{R}$ of the sort Boden describes emerging from the cross-pollination of ideas and norms.

Our intent is to capture specific curiosity – intentional pursuit of further transformation along a search trajectory incited by a particular transformative example – within an expansion of this framework. To achieve this, we need to expand Wiggins' formalisation in four ways:

- To enable a creator to be surprised by its own output, as in Schön (1983), a creative system must externalise and re-perceive its creations as part of the generative process.
- To incorporate the influence of other creators, the input to a creative system's generative process must include all artefacts it has observed, not just its own creations.
- To model the probabilistic nature of expectations, the conceptual space should be a fuzzy set of *probable* concepts, not a crisp set of *possible* concepts.
- To separate unexpectedness from inexplicability, the system should be aware of its confidence in the predicted likelihood of any concept being in the conceptual space.

These changes capture the situated, social, and expectation-based nature of creative systems, allowing us to use Wiggins' formalisation to explore the question of when, where and why transformative creativity occurs.

## Surprise as $\mathscr{R}$-transformation

We now formally describe the above expansion of the framework. The literature on design cognition describes how creators can be surprised by their own creations. For this to be possible in an artificial creative system those creations must be represented in a way that contains additional information not used to create them. To reflect this we add a step to the post-generation process of the creative system. First, $\langle\langle .,.,. \rangle\rangle$ is used to generate a new set of outputs, $c_{out}$, from the current inputs, and then instead of those outputs being directly appended to $c_{in}$ for the next iteration, they are first reified via a function $r$, which maps from a concept to an externalised representation of that concept which we call an "artefact", and then re-perceived by a function $p$, which maps from an artefact back to a concept in $\mathscr{U}$. The nature of perception, reification and the space of possible artefacts is beyond the scope of this paper.

To capture a society of creative systems that influence each others' work, we must amend the generative step of Wiggins' formalisation: instead of applying the interpreter $\langle\langle .,.,. \rangle\rangle$ to just $c_{in}$, the ordered set of that system's own past creations, we must apply it to all an ordered set of all concepts the system has previously observed, regardless of their source. We assume our creative system is part of a society of creative systems that are all producing artefacts within the same domain (by which we mean they share at least $\mathscr{U}$). Each creative system possesses an additional ordered set of concepts, $c_{obs}$ that it has observed but did not create. Different societies may have different structures in which creative systems are exposed to each others' work in more or less selective ways, but $c_{obs}$ is generated by applying the perception function $p$ defined above to some subset of the artefacts externalised by other creative systems. If $c_{obs}$ is non-empty before a creative system has generated any concepts of its own, then those pre-existing known artefacts are the system's *inspiring set* (Ritchie 2001). We can now describe the generation step in our amended formalisation, applying the interpreter to the union of creations and observations, and afterwards reifying and re-perceiving the output, i.e. $c_{out} = p(r(\langle\langle \mathscr{R},\mathscr{T},\mathscr{E} \rangle\rangle(c_{in} \cup c_{obs})))$.

Wiggins suggests that the output of the interpretation function for $\mathscr{R}$ (a real number in $[0,1]$) be converted to a boolean value indicating membership in $\mathscr{C}$. We propose instead that $\mathscr{C}$ be considered a fuzzy set, with the output of the interpreter defining a membership function $l : \mathscr{U} \rightarrow [0,1]$ that indicates the likelihood of observing each concept as part of the domain. This transforms Wiggins' space of *possible* artefacts into a space of *probable* artefacts, and lets us capture all the rich relationships between concepts that influence their mutual likelihoods. We derive this interpretation from our previous work on expectation, novelty and transformation, see Grace and Maher (2014) for details.

We introduce into our framework a notion of confidence. This serves to differentiate unexpectedness (a violation of confident expectations) from ignorance (Ortony and Partridge 1987). To achieve this we replace the $[\![.]\!]$ interpreter from Wiggins with a modified version, $(\!|.|\!)$, which differs only in that it returns a 2-tuple of real numbers in [0,1] for each artefact to which it is applied. The first, as in $[\![.]\!]$, is

the truth value, which becomes the value of the likelihood function $l$ that defines the artefact's membership in the resulting set. The second value is the system's confidence, with 0 indicating a complete lack of confidence and 1 indicating complete certainty. This confidence becomes another function $c : \mathscr{U} \rightarrow [0, 1]$. We use $(\!|.|\!)$ when generating the conceptual space with $\mathscr{R}$, as in:

$$\mathscr{C} = (\!|\mathscr{R}|\!)(\mathscr{U})$$

As a result our $\mathscr{C}$ is a fuzzy set of concepts with a membership function $l$ defining the likelihood of observing each concept in $\mathscr{U}$, as well as a similar confidence function $c$ defining the system's confidence in each artefact's likelihood. These functions are compiled from the first and last elements, respectively, of the tuples output by $(\!|.|\!)$. That is, for each concept $a \in \mathscr{U}$, given $(\!|\mathscr{R}|\!)(\{a\}) = (a_l, a_c)$ and assuming $a_l > 0$:

$$a \in \mathscr{C}, l(a) = a_l, c(a) = a_c$$

From this perspective, Wiggins' $\mathscr{R}$ becomes the creative system's *expectations about the creative domain*. This connection between conceptual space membership and expectation allows us to describe the influence of surprise on creative search. In our amended framework, $\mathscr{R}$-transformation is commonplace and necessary, a natural effect of creative systems acquiring the knowledge they need to competently model the society's rules about the domain through their own experience.

A creative system experiences *expectation failure* when the conceptual representation of a newly observed artefact has a low a-priori likelihood in the conceptual space. We can then distinguish two kinds of artefact that cause expectation failure: inexplicable ones, where the system is not confident of its predicted low likelihood, and unexpected ones, where it is. An unexpected artefact $a_u$ is one for which:

$$a_u \in (c_{in} \cup c_{obs}), l(a_u) \approx 0, c(a_u) \approx 1$$

Complementarily, for an inexplicable artefact $a_i$:

$$a_i \in (c_{in} \cup c_{obs}), l(a_i) \approx 0, c(a_i) \not\approx 1$$

Only in the first case can we say that the agent's expectations were violated – in the absence of a confident prediction the system was merely ignorant. Both inexplicable and unexpected artefacts should by rights induce a transformation of the domain knowledge in $\mathscr{R}$, as well as potentially transformations of $\mathscr{T}$. Those transformations can be considered a result of creativity if the artefact(s) that caused them are valuable under $\mathscr{E}$. Given our definition of unexpectedness in terms of $\mathscr{R}$ we can restate how our expanded formalisation captures the dyad of novelty and value. The rules in $\mathscr{E}$ will be concerned with the evaluation of artefacts' performance, quality, style, and other components of value, and some portion of $\mathscr{T}$ will use those evaluations to direct search. Contrastingly, some other subset of $\mathscr{T}$ will be concerned with novelty seeking: evaluating the dissimilarity of new artefacts to existing ones using measures of novelty, surprise and transformativity. We refer to this novelty-seeking subset as $\mathscr{T}_n \subset \mathscr{T}$. These latter traversal rules will be based on the likelihoods, confidences, and transformations of $\mathscr{L}$ associated with artefacts.

We do not seek to resolve the disputes surrounding the definitions of novelty, surprise or transformation, only suggesting that $\mathscr{T}_n$ could contain metrics for any or all of those, but we do require that for any creative system $\mathscr{T}_n \neq \emptyset$.

Any artefact valued by both $\mathscr{T}_n$ and $\mathscr{E}$ can be considered p-creative. This generative act is *serendipitous* if the search process possessed no specific intent to create that artefact or anything like it. An artefact discovered to be transformative by $\mathscr{T}_n$ after its creation was not the result of a directed search, for the system cannot know how its knowledge will be transformed by new observations. This places limits on a creative system's ability to generate framing about its creative output: serendipity defies satisfying explanation.

In the next section we use our definitions of inexplicable and unexpected artefacts to describe different possible kinds of transformational creativity. We also propose how a system might adopt constraints on its future generation in response to unexpectedness, and thereby intentionally seek out further unexpected discoveries.

## Specific curiosity as a consequence of surprise

A system that has observed inexplicable artefacts will attempt to learn: to improve its (clearly insufficient) knowledge of $\mathscr{U}$. We consider *learning* to be a creative system's response to the inexplicable, and it is our first possible kind of $\mathscr{R}$-transformation. Learning can be expressed as the application of $\mathscr{T}_{\mathscr{L}}$ to produce new $\mathscr{R}$ and/or $\mathscr{T}$ in response to inexplicable artefact(s) in $c_{in}$ or $c_{obs}$. While the mechanisms of learning will be specific to the rules in $\mathscr{L}_{\mathscr{L}}$, we can describe its effects: it attempts to transform $\mathscr{R}$ such that the likelihood of previously observed artefacts increases.

A system that has observed unexpected artefacts will be surprised. We consider *artefact-induced surprise* to be a creative system's response to unexpected artefacts, and it is our second kind of $\mathscr{R}$-transformation. Artefact-induced surprise can be expressed as the application of $\mathscr{T}_{\mathscr{L}}$ to produce new $\mathscr{R}$ and/or $\mathscr{T}$ in response to unexpected artefact(s) in $c_{in}$ or $c_{obs}$. Learning occurs from unexpected objects as it does from inexplicable ones, producing $\mathscr{R}$-transformations that increase the expected likelihood of previous observations.

Inspired by cognitive studies of reflection in human designers by Suwa et al (1999) and others we can now consider how surprise might affect a system's future generative behaviour (i.e. cause transformation of $\mathscr{T}$). Specific curiosity, as introduced earlier, is the deliberate pursuit of specific new knowledge or stimuli through the adoption of goals or constraints on behaviour. In the context of a creative system this is $\mathscr{T}$-transformation with the goal of exploring an unexpected stimulus, based on the hypothesis that (as observed in human designers), surprise begets further surprise. This can also be considered a form of active learning (Cohn, Ghahramani, and Jordan 1996), where the system actively tries to fill the gaps in its knowledge through generation.

To become specifically curious about an artefact is to seek to create more artefacts that embody the interesting things about it. We formalise this as follows: given an unexpected

artefact $a_u$ we can determine the subset of rules that contributed to its confident low-likelihood prediction: $\mathscr{R}_{a_u} \subseteq \mathscr{R}$. These rules embody the domain knowledge that was violated by the perception of the new artefact, in that they produced a confident prediction that was proven wrong. This subset forms the basis of the system's specific curiosity, in that the system can use them to pursue artefacts that are unexpected according to just those rules. To define this we induce $r$, a relevance function over concepts that measures the complement of the likelihood of a concept occurring in a conceptual space defined exclusively by $\mathscr{R}_{a_u}$. Accordingly $r(a) \approx 1$ for any artefact $a$ that would be considered unexpected according to the same rules as was $a_u$, including $a_u$ itself. Conversely, any artefact that is not unexpected, or is unexpected due to other rules not in $\mathscr{R}_{a_u}$, would produce a lower value of $r$. We can then define specific curiosity about $a_u$ as replacing $\mathscr{T}_n$ with a single rule that seeks artefacts for which $r(a) \approx 1$. This (temporarily) redirects the system's general (i.e. diversive) search for novel artefacts towards those that are unexpected according to the same rules as the one that caused the surprise.

By constructing a relevance function from the rules violated by the unexpected artefact we focus the system upon the parts of its own knowledge that produced the unexpected result. The results of this specific curiosity will vary based on the structure of the knowledge that was violated. If the rules define boundaries of the domain, the relevance function will value artefacts that break the same boundaries as the focus of curiosity. If the violated rules placed the focus in a new or rare category, the relevance function will value artefacts in that category. If the violated rules define an expected relationship between components of the artefacts' representation, the relevance function will value artefacts that break the same relationship in the same way as the focus. In each case the relevance function will value artefacts that are in some way similar to the one that caused surprise, but with that similarity determined by the system's knowledge.

The hypothesis driving this specific curiosity is that regions of the conceptual space that generate one unexpected artefact likely have the potential to generate more, and searching nearby has a greater chance to yield further unexpected (and therefore potentially creative) artefacts than searching elsewhere in the space. This behaviour aligns with the concept of creative autonomy and situational adaptation of goals described in (Jennings 2010).

In the following section we illustrate the above kinds of $\mathscr{R}$-transformation with examples from the domain of recipe generation.

## A worked example of unexpectedness-triggered specific curiosity

As a hypothetical example of our unexpectedness-triggered reformulation approach, consider the creative domain of recipes. Culinary creativity has recently attracted attention in the computational creativity community (Morris et al. 2012; Varshney et al. 2013), and we draw upon it as a way of illustrating our model of specific curiosity.

Assume a hypothetical recipe generation system integrated with a large online recipe repository. The system has access to all the recipes posted by humans, and is tasked with supplementing that database with its own creations. Each recipe is an artefact represented by its ingredients and their quantities, the preparation steps, and metadata such as cooking time and user-applied tags. This is supplemented by behavioural information for each recipe: the full text and ratings of its set of user reviews. The system's task is to generate novel and valuable recipes, and submit them for human consumption and review. $\mathscr{E}$ is based on aggregated user ratings. $\mathscr{R}$ is based on domain knowledge represented by a set of predictive models that describe the likelihood of various combinations of ingredients, quantities, tags, categories, reviews and ratings occuring. We can now describe three ways that this implementation of our framework could encounter transformative creativity.

The first cause of $\mathscr{R}$-transformation is encountering an *inexplicable* recipe. This would be commonplace while the system developed its knowledge about the domain (as the pre-existing human-created recipes that form its inspiring set were added to its database). For example, assume that the system, early into its learning, encountered its first slow-cooked dish. The existing rules in $\mathscr{R}$ would assign a very low a-priori likelihood to a recipe with an eight hour cooking time, but having seen so few previous recipes of any kind it would also assign a low confidence to that prediction. The result would be learning – transforming $\mathscr{R}$ to incorporate the new range of observed cooking times. No surprise or specific curiosity would result – the system's understanding of the conceptual space improved as a result of observing new kinds of artefact that had been produced by others, a necessary and commonplace step of acquiring competency in a creative domain.

The second cause of $\mathscr{R}$-transformation is an *unexpected* recipe. This occurs when the system makes confident predictions of the likelihood of observed recipes, but is still wrong, possibly as the result of a change in the behaviour of the other creative systems in the society (which, in this case, are the human submitters of recipes). Consider what would happen to the system's knowledge about the ingredient "ginger" if its inspiring set (i.e. the recipes in $c_{obs}$ it used to populate $\mathscr{R}$ before generating any artefacts of its own) contained mostly Western recipes, and it developed confident predictions about that ingredient before being exposed to Eastern-inspired recipes. It would confidently expect that ginger was found mostly in sweet baked goods, alongside ingredients like butter, sugar and flour. Encountering a recipe for ginger-and-soy chicken would be highly surprising, causing it to adapt its domain knowledge to fit the new recipe. In this case the creative system had a robust, but incomplete model of the creative domain, and observed an artefact that it would consider p-creative, even though that artefact's creator may have considered it novel.

The third cause of $\mathscr{R}$-transformation is as a result of specific curiosity caused by an earlier surprising recipe. As another example, consider "chicken paprikash", a Hungarian-inspired dish that combines a roux-based sauce with curry-like spices (cumin, paprika and chili). This is an incongruous combination of ingredients and instructions, as the ma-

jority of roux-based sauces are flavoured with herbs, stocks and/or cheeses. Our creative system encounters this recipe, becomes surprised as in the ginger-and-soy chicken example, and uses that surprise to trigger specific curiosity. The rules in $\mathscr{R}$ that confidently assign a low likelihood to a recipe containing both the steps for a roux and the ingredients for a curry are extracted as $\mathscr{R}a_u$. A relevance function is then constructed from those rules that evaluates the degree to which a recipe violates them, and this function replaces the novelty-seeking rules in $\mathscr{T}_n$. The system begins generating recipes that violate these specific rules, such as a roux-based sauce with other unexpected ingredients (such as chocolate), or curries with unusual preparation steps (such as being baked into a pie). The authors feel compelled to mention that they are not chefs, but encourage readers to assume for the sake of argument that those new recipes are both novel and valuable. The observation of these new recipes would lead to additional $\mathscr{R}$-transformation, and this time that transformation can be said to have a deliberate cause. These artefacts were not created serendipitously, they were intentionally generated as the result of a targeted exploration of a specific region of the creative domain, and their discovery further transformed the conceptual space.

Specific curiosity can be triggered both from a creative system's own creations, or from those of the other creative systems within its society (here the human user-base of the recipe website). In the case of the chicken paprikash above, the specific curiosity episode was triggered by the observation of a surprising creative artefact generated by a human – other likely external curiosity-triggers in this domain could include the addition of bacon to sweet foods, the inclusion of leafy greens in smoothies, the rise of a new and novel "superfood", or a seasonally resurgent ingredient.

The creative system could trigger its own specific curiosity episodes by generating recipes that, once reified and reperceived, were considered surprising. Consider, for example, rules in our creative system's $\mathscr{T}$ that use computational analogy-making to map between two recipes and then transfer a new ingredient from the source to the target. An analogy could be constructed between a calzone and an omelette, as both consist of a base layer to which toppings are added before the base is folded over to create a filled final product. The rules for analogical transfer in $\mathscr{T}$ identify that the tomato paste spread on the calzone is missing from the omelette, and create a new recipe in which a tomato sauce is spread over the omelette before folding. This would be considered unexpected by the rules in $\mathscr{R}$ that pertain to omelettes, which would make confident predictions that a tomato-based sauce would be unlikely to be involved in an omelette recipe. The authors again remind the reader that we are definitely not chefs, but let us assume that the resulting sauced omelette was also considered valuable. Specific curiosity about that unexpected combination of ingredients and cooking methods would result in a transformation of $\mathscr{T}_n$ to specifically seek out further recipes involving unusual ingredients being added to omelettes during cooking. Generating new artefacts under this transformed search trajectory could lead to the recipes with further unexpected mid-omelette additions such as spices or fruits. These new creative artefacts would further transform the rules in $\mathscr{R}$ that pertain to omelette creation, and if they were also considered valuable according to $\mathscr{E}$ then they would constitute intentional transformative creativity.

## Conclusions

We have described an extension to Wiggins' (2006) framework that captures the notions of unexpectedness, surprise and specific curiosity. This approach is motivated by the need for creative systems that can make autonomous evaluative decisions and exhibit intentional behaviour (Jennings 2010; Saunders 2012). The solution proposed in our framework draws on literature from design cognition which suggests that human creators are not only capable of self-surprise but that it is a significant driver of creative output. Based on this inspiration from cognition we model surprise based on violation of a creative system's learnt model of the conceptual space, and describe specific curiosity behaviours that explore surprising stimuli.

Within our framework we can distinguish three causes of transformational creativity: inexplicable artefacts, unexpected artefacts, and specific curiosity. If found in an artefact that was also valuable the first would not be creative (as the transformation resulted from a lack of sufficient knowledge to make predictions), the second would be serendipitous creativity (as the system stumbled upon it without any deliberate goal), and the last would constitute intentional creativity. Specific curiosity describes the iterative cycle between the $\mathscr{R}$-transformation that occurs when observing or creating an unexpected artefact, the $\mathscr{T}$-transformation that facilitates the resulting search for more, similarly surprising artefacts, and the resulting $\mathscr{R}$-transformation that heralds the success of that deliberate search. Our future work involves the development of systems like the one presented here as an example: creative machines capable of surprise, specific curiosity and autonomous intent.

## References


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.

Berlyne, D. E. 1966. Curiosity and exploration. *Science* 153(3731):25–33.

Best, D. 1981. Intentionality and art. *Philosophy* 56(217):349–363.

Boden, M. A. 2003. *The creative mind: Myths and mechanisms*. Routledge.

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.

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *Journal of artificial intelligence research*.

Cross, N. 2004. Expertise in design: an overview. *Design studies* 25(5):427–441.

Cully, A.; Clune, J.; and Mouret, J.-B. 2014. Robots that can adapt like natural animals. *arXiv preprint arXiv:1407.3501*.

Dewey, J. 2005. *Art as experience*. Penguin.

Gebser, M.; Kaufmann, B.; and Schaub, T. 2009. Solution enumeration for projected boolean search problems. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer. 71–86.

Getzels, J. W., and Csikszentmihalyi, M. 1976. *The creative vision: A longitudinal study of problem finding in art*. Wiley New York.

Goldschmidt, G. 1991. The dialectics of sketching. *Creativity research journal* 4(2):123–143.

Grace, K., and Maher, M. L. 2014. What to expect when youre expecting: the role of unexpectedness in computationally evaluating creativity. In *Proceedings of the 4th International Conference on Computational Creativity, to appear*.

Grace, K.; Maher, M.; Fisher, D.; and Brady, K. 2014. A data-intensive approach to predicting creative designs based on novelty, value and surprise. *International Journal of Design Creativity and Innovation* (Ahead of Print).

Grace, K.; Gero, J.; and Saunders, R. 2015. Interpretation-driven mapping: a framework for conducting search and re-representation in parallel for computational analogy in design. *AI EDAM* 29(2):185–201.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Kelly, N., and Gero, J. S. 2014. Interpretation in design: modelling how the situation changes during design activity. *Research in Engineering Design* 25(2):109–124.

Krzeczkowska, A.; El-Hage, J.; Colton, S.; and Clark, S. 2010. Automated collage generation-with intent. In *Proceedings of the 1st international conference on computational creativity*, 20.

Lawson, B. 2006. *How designers think: the design process demystified*. Routledge.

Liapis, A.; Martınez, 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.

Macedo, L., and Cardoso, A. 2001. Modeling forms of surprise in an artificial agent. *Structure* 1(C2):C3.

Merrick, K., and Maher, M. 2009. Motivated reinforcement learing.

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.

Newell, A.; Shaw, J.; and Simon, H. A. 1959. *The processes of creative thinking*. Rand Corporation.

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.

Ritchie, G. 2001. Assessing creativity. In *Proc. of AISB01 Symposium*.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. *Computational and Cognitive Models of Creative Design V, Key Centre of Design Computing and Cognition, University of Sydney, Sydney* 113–139.

Saunders, R. 2011. Artificial creative systems and the evolution of language. In *Proceedings of the Second International Conference on Computational Creativity*, 36–41.

Saunders, R. 2012. Towards autonomous creative systems: A computational approach. *Cognitive Computation* 4(3):216–225.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *Autonomous Mental Development, IEEE Transactions on* 2(3):230–247.

Schon, D. A., and Wiggins, G. 1992. Kinds of seeing and their functions in designing. *Design studies* 13(2):135–156.

Schön, D. A. 1983. *The reflective practitioner: How professionals think in action*, volume 5126. Basic books.

Smith, A. M., and Mateas, M. 2011. Knowledge-level creativity in game design. In *Proc. of the 2nd International Conference in Computational Creativity (ICCC 2011)*.

Sternberg, R. J., and Lubart, T. I. 1999. The concept of creativity: Prospects and paradigms. *Handbook of creativity* 1:3–15.

Suwa, M.; Gero, J.; and Purcell, T. 1999. Unexpected discoveries and s-inventions of design requirements: A key to creative designs. *Computational Models of Creative Design IV, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia* 297–320.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schoergendorfer, A.; and Chee, Y.-M. 2013. A big data approach to computational creativity. *arXiv preprint arXiv:1311.1213*.

Weisberg, R. W. 1993. *Creativity: Beyond the myth of genius*. WH Freeman New York.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

# Computational Poetry Workshop: Making Sense of Work in Progress

**J. Corneli**[*] Goldsmiths College
**A. Jordanous** University of Kent
**R. Shepperd** Goldsmiths College
**M. T. Llano** Goldsmiths College
**J. Misztal** Jagiellonian University
**S. Colton** Goldsmiths College
**C. Guckelsberger** Goldsmiths College

**Abstract**

Creativity cannot exist in a vacuum; it develops through feedback, learning, reflection and social interaction with others. However, this perspective has been relatively under-investigated in computational creativity research, which typically examines systems that operate individually. We develop a thought experiment showing how structured dialogues can help develop the creative aspects of computer poetry. Centrally in this approach, we ask questions of a poem, inviting it to tell us in what way it may be considered a "creative making."

**Keywords**: computer poetry, social creativity, flowcharts, Writer's Workshops

'We *can* talk,' said the Tiger-lily: 'when there's anybody worth talking to.'

Through the Looking Glass, Lewis Carroll

## Introduction

We are writing in a large part to champion Alan Turing's proposal that intelligent machines should "be able to converse with each other to sharpen their wits" (Turing, 1951). The formalism that we propose builds on the notion of social cybernetics that flows from the following propositions of Heinz von Foerster's, which he uses to theorise systems in which participants can responsibly specify their own roles in relationship to other system participants:

"Anything said is said *by* an observer."
"Anything said is said *to* an observer."

(Von Foerster, 2003 [1979])

According to Jaako Seikkula and Tom Arnkil, who draw on the philosophical and literary analysis of Mikhail Bakhtin (Bakhtin, 2010 [1986], 1984 [1963]) in their approach to psychosocial work,

"Dialogues could be called 'the art of crossing boundaries'. Instead of trying to control others, the parties reach out towards each other to hear their views better, to generate shared languages and to join resources."

(Seikkula and Arnkil, 2014, p. 23)

[*]Corresponding author. Email: j.corneli@gold.ac.uk

This paper outlines a study of social creativity with a dialogical emphasis, taking computer poetry as our working domain. It uses the Writer's Workshop model (Gabriel, 2002) as the virtual laboratory in which to conduct a thought experiment. The findings of our study are applied to the FloWr system (Charnley, Colton, and Llano, 2014). We focus on the following questions in turn:

- How has the social dimension of creativity been explored in CC to date?
- How can a created artefact tell us about its making, and what can this contribute to CC?
- How can computer poetry contribute to developing a process-based theory of poetics?
- What would have to change about the FloWr system to implement the computational poetry workshop approach?
- What are the pros and cons of the workshop approach?
- What might be the future role of dialogue in CC?

## Background

### Social creativity in CC

Minsky noted that computers need to be social if they are to deal with problems of any great complexity (Minsky, 1967, 1988). We believe that this is particularly true for challenges in computational creativity, since the essence of creativity lives in its appreciation by the creative entity itself and its audience. With creativity in 'the eye of the beholder' (Cardoso, Veale, and Wiggins, 2009), the ability to respond to evaluation during the creative process (Poincaré, 1929 [1908]; Csikszentmihalyi, 1988) becomes pivotal. Social creativity expands this paradigm by introducing co-creators to the process, and creating works that rely on dialogue, reflection, and multiple perspectives (e.g. the stages suggested by (Gervás and Leon, 2014)). 'Results' may be steeped in process and are not always based on consensus.

The Four Ps of creativity – the creative Person, Product, Process and Press (i.e. environment) (Rhodes, 1961) – have been emphasised in general creativity research. *Pluralising* these terms (Persons, Products, Processes) calls further attention to a social dimension of creativity, and would emphasise the way the "Press" accommodates multiple multidirectional perspectives akin to a social network in both the modern and original senses. The Pluralised Ps remind us

that in order to understand creativity it is not sufficient to model a lone creator or to generate an attractive artwork.

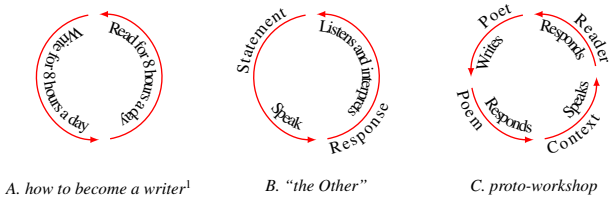
To date, computational creativity research has achieved many successes in computational generation of creative products, but the question of how these systems could adapt and learn from feedback to improve their creativity is less-explored in computational creativity (Jordanous, 2015). Evaluation has been advanced as a pivotal contributory part of the creative process, but researchers often give priority to generating artefacts that could be seen as creative over the task of incorporating feedback and evaluation within the processing of a creative system (Jordanous, 2011).

At the previous year's International Conference on Computational Creativity (ICCC 2014) the opening session had the theme "co-creation." However in the main proceedings of the conference, 36 out of 49 papers (approximately 3 in 4 papers) do not appear to mention social interaction or the ability to respond to feedback. Some notable exceptions highlight the usefulness of interaction and feedback for creative systems (McGraw and Hofstadter, 1993; Colton, Bundy, and Walsh, 2000; Sosa, Gero, and Jennings, 2009; Pérez y Pérez, Aguilar, and Negrete, 2010; Pease, Guhe, and Smaill, 2010; Saunders, 2012). Some of this work is influenced by the DIFI (Domain-Individual-Field-Interaction) framework (Csikszentmihalyi, 1988). However, social interaction between creative agents and their audience is often overlooked or relatively simplified: some examples in the domain of computer poetry presented below give the flavour. Increased development of the interactivity of creative systems, especially where this affects the way these systems works, has been highlighted as deserving more attention (Colton and Wiggins, 2012).

FloWr is a framework for implementing creative systems as scripts over processes that can be manipulated visually as flowcharts (Charnley et al., 2014). Its general approach consists of linking the inputs and outputs of code modules, called *ProcessNodes*, together to create a linear flow of data. The resulting Flowcharts can be constructed and executed visually through a GUI; however, they are ultimately represented as scripts, which are the main medium of FloWr. Experiments with automatic process generation in FloWr, reported in (Charnley et al., 2014), highlight the ability of the tool to do meta-programming and modify its own flowcharts. This suggests that FloWr has potential as an environment for modelling social creativity, where the observers are nodes and flowcharts, and their languages are, respectively, programming and meta-programming instructions.

### . . . and in computer poetry

In the domain of poetry-generation, there have already been several attempts to simulate social creativity by incorporating multi-agent systems. In WASP (Gervás, 2010), social behavior is simulated by incorporating a cooperative society of readers/critics/editors/writers consisting of specialized families of experts that cooperate during the poetry-generation process. The McGONAGALL system (Manurung, Ritchie, and Thompson, 2012) incorporates diverse modules as operators in evolutionary algorithms that produce poems fulfilling the constraints on grammar, meaning and poeticity. This approach facilitates the pursuit of several alternative solution paths in parallel, focusing on more promising results or coming back to former ideas. However McGONNAGALL does not provide any communication between modules. In the MASTER system for computer-aided poetry generation (Kirke and Miranda, 2013) a society of agents in various emotional states influences each other's moods with their pieces of poetry. The poetry-generation process is based on social learning as the agents interact by reciting their own pieces of poetry to each other. The generated poems are based on repeated words and sounds, and are closer in some ways to music than to typical language. Montfort, Pérez y Pérez, Harrell, and Campana (2013) and Misztal and Indurkhya (2014) use blackboard approaches to poetry-generation, in which independent specialized modules cooperate via a shared global workspace, *à la* (Baars, 1997). "Experts" exchange information using the blackboard, but without direct communication and without feedback about the reception of their created artifacts.




*A. how to become a writer*[1] *B. "the Other"* *C. proto-workshop*

Figure 1: (A) gives a simple recipe for the growth and development of a writer; (B) *response* always has dimensions that goes beyond the utterance that is overheard; (C) adds a *reader* who shares the context with the writer and responds.

In connection with our work in the current paper, we did a limited proof-of-concept reimplementation of some of the core methods of blackboard poetry system inside of FloWr; we include one of the generated poems and the corresponding flowchart.

## Methods

### "What are the proposed 'lab rats'?"

The generative side of the cycles in Figure 1 has been studied more than the reflective side. Our "lab rats" are, accordingly, not poems per se, but rather, *instances of reading and responding to poetry*. Naturally, such responses could be more or less "canned" (as with Michael Cook's humorously non-specific AppreciationBot[2]), so the question becomes: what constitutes an interesting and useful response, and how will these be developed? The idea of responses is useful at various levels. We focus here on staging an encounter between writer and reader.

### Writer's Workshops

Quoting (Gabriel, 2002, pp. 2–3):

> The original idea behind the writers' workshop was to do a *close reading* of a work... looking at the words on

[1] According to (King, 2000).

[2] https://twitter.com/appreciationbot

> the page rather than the intentions of the author or the historical and aesthetic context of the work. Under this philosophy, the workshop doesn't care much what the author feels about what he or she wrote, only what's on the page.

Framing and any other contextualisation of the work *as it is intended to be presented* is permitted, and receives critical attention. We define a *Workshop* closely following Gabriel's outline, to be an activity consisting of these steps: `presentation`, `listening`, `feedback`, `questions`, and `reflections`. The first and most important feature of `feedback` is for the listener to say what they heard; in other words, what, for them, is in the work. In some settings this is augmented with `suggestions`. After any `questions` from the author, the commentators may make `replies` to offer clarification. In related recent work, we have shown how the Workshop framework can help foster serendipitous discovery and invention (Corneli, Pease, Colton, Jordanous, and Guckelsberger, 2015; Corneli and Jordanous, 2015).

### Content as creative process

Giving agency to the poem rather than the poet's intentions, the poem illuminates its own creative process. This informs our approach to Workshop interactions, which are focusing on the poem observing its own construction. We're interested in context not in the literary or historical sense but in the micro-history of the poem's creative evolution. The originary and therefore unpredetermined nature of the creative process means that the outcome represents a more accurate and objective evidence of the process than the poet's attempt to explain the process. Moreover, to the extent that a creator knows what is expressed through the creative process, even he or she learns this only in the course of doing the work. Observers are only able to consider after the fact how a creator may have selected and rejected various possibilities. The *content* of the poem is no more and no less than how the poem was made.

> "In a poem, objective material becomes the content and the matter of the emotion and not just its evocative occasion." (Dewey, 1958 [1934], p. 69)

P. G. Whitehouse writing on Dewey's *Art as Experience* suggests that Dewey joins Collingwood in separating aesthetic emotion from any notion of inspiration that could be considered to be something like raw materials. An emotion is *aesthetic* when it "adheres to an object formed by an expressive act" (Whitehouse, 1978, pp. 149–156). However, "the art object does not have emotion for its significant content"; rather, the emotion "belongs to the self that is concerned in the movement of events toward an issue that is desired or disliked" (Dewey, 1958 [1934], p. 14).

### Aspects of the creative process

Doug Anderson and Carl Hausman take Collingwood's study further and map the creative process roughly as follows (Anderson and Hausman, 1992, pp. 299-305):

> Disturbance → aesthetic emotion → response → artist's decision on components of expression → feeling of easement plus a simultaneous emerging of a unique imaginative expression → alleviation → realising and converting prior psychical emotion → unique aesthetic experience including new conscious emotion

The poem is a *work of progress* before it is a *work in progress*. The purpose of a poetry workshop that attends to the content of the poem as process is to illuminate what the poet is exploring through his/her creative process and through the poem. The process of reading a poem is also a process of *poiesis* – and in the Workshop, the reader joins the writer in the process of creation. Asking questions like those listed in in Table 1 tells us what the constituent parts of the poem are doing.

### Relevance for CC research

From a CC standpoint, asking what the work tells us about the creative process gives an objective and critical focus on "creative evolution" (Bergson, 1911 [1907]) and provides an antidote to the seductions of mere generation. A poetry workshop gives participants the opportunity to read the drafts and final versions of poems by other Workshop participants, a shared culture of critique that can be applied to previously existing poems, and a structured way to gather feedback on one's own work in progress. These analyses, unbiased by the explanations of the (software) creator, will allow participants to explore and extend the conceptual space around poetry, or in practical terms, the toolbox the agents can access. "Extending" expresses both a refinement of the tools used and the introduction of entirely new tools. Moreover, reverse-engineering of the creative process from artefacts will help to teach agents participating in the workshop at which stage of *their* creative process these new tools or extensions could potentially be used. Dialogue in the workshop involves "respecting the voices of each of the participants" (Seikkula and Arnkil, 2014), be they agents, poems, or individual words – and suggests that we look at the "art of boundary crossing" that is to be found *inside* poems.

## Bridges between 'theory' and 'practice'

Our *ansatz* is that the Workshop could serve as a way to develop a process-based theory of poetics. There are certain prerequisites: in particular, an underlying context is required, shared (with respect to differing points of view) by the poet and the reader/listener (see Figure 1). Participants are assumed to have relatively stable, enduring but evolving, identities – either might be able to ask "Who am *I*?" and "Who are *you*?" (Bakhtin, 1984 [1963], p. 251). Answers would acknowledge a prepared mind with certain prior questions, abilities, involvements, and so on. However, within the Workshop dialogues, the discussion focuses solely on the work itself. Persistent identities allow participants to learn from these exchanges.

Table 1 contains a list of questions that a reasonably sophisticated poetry reader might ask about poems. This is complemented by a list of questions that could be addressed, in a

| Question | Examples |
|---|---|
| What are the register(s) of the poem? | cliched, instructive, imperative |
| Who is addressed? | reader, poet, friend, rival, confidante |
| What position(s) are present in the poem? | pleading, remonstrating, ephemeral |
| What is the poem doing with the reader? | accuse, bewilder, alienate |
| Who are the characters in the poem? | "the falconer", "you", narrator, "two men" |
| What is the role of image(s) in the poem? | "the sea", "a bicycle"; multiple meanings |
| What functions, mechanics, and paradigms are present for the reader to engage with? | communication, subverted cliche |
| What problems, discomforts, or diseasements are invoked in the poem? | horror, self-loathing, rejection, desire |
| How do these evolve? | E.g. an image may start to take over from a register |
| What is the world of the poem, and how does the poem distinguish between this and its perception of this? | "Surely", "must"; sacred vs mundane; perspectival vs surreal; tale vs telling |
| What are the overlaps, transitions, implicit dialogues? | "twinned" lines/ideas, juxtaposed parts of the poem |
| What role does the chronology of reading play, versus references to chronology and chronological positions within the poem? | flagged development, evolution, movement, stasis |
| How are lexical categories used? | solid nouns, tortuous adjectives, indistinct adverbs |
| Are there discernible allusive effects? | illustrating the literary apprenticeship of the author (or reader) |
| Where is the poet presented with respect to the poem? | Confidence, determination, exploration |

Table 1: Questions that we ask when reading a poem

straightforward programmatic manner (Table 2). Each of the examples listed in the right-hand column of Table 1 (and a plethora that are not listed) present a way of thinking about the poem. We can see these as roughly analogous to the agents in Table 2 (Minsky, 2006).

To illustrate, in response to a computer-generated poem:

*Oh dog the mysterious demon*
*Why do you feel startle of attention?*
*Oh demon the lonely encounter*
*ghostly elusive ruler*
*Oh encounter the horrible glimpse*
*helpless introspective consciousness*

A human critic might offer the following feedback:

1. The use of the word *mysterious* in the first line has no resolution, real or attempted, or quest to find one.
2. The use of the word *attention* is not being interrogated or acknowledged for its importance. Its qualifying word is *startle*, used here as an adjective; acknowledging the fact that the attention is noted, but is not yet part of the transformative of the poem.
3. This is repeated in the next references to the aesthetic experience as a *lonely encounter*, *exclusive ruler*, *horrible glimpse* and *introspective consciousness*.
4. The contact made between the poem and its own construction is qualified in negative terms attached to the words *demon*, *encounter* and *consciousness*.
5. This poem does not welcome the intimacy of bringing anything to aesthetic consciousness so that it might be expressed. Why do I say that? Because the words are generalised and *horribly* imprecise.
6. The poem does not move toward a better understanding of the ideas it alludes to. The vocabulary seems to associate exploration with fear and isolation and this is (paradoxically) quite an interesting acknowledgment of the poem's refusal to go anywhere i.e. to become a thing transformed by a creative process.

| Question | Agent concerned |
|---|---|
| ***Word level*** | |
| What is are dictionary definitions of this word? | WordNet expert |
| What are its etymological roots? | Etymology expert |
| Where did this word come from? | Provenance expert |
| What pronouns are used in the poem? | Pronoun expert |
| ***Phrase level*** | |
| What are the components? | Keywords expert |
| Do the components have a negative or positive connotation? | Association expert |
| What are the modifiers attached to the components? | Modifer expert |
| ***Sentence level*** | |
| What is the parse tree? | Grammar expert |
| ***Line level*** | |
| How long is the line? | Counting expert |
| Where does it break? | Breathing expert |
| Where is there white space? | Position expert |
| ***Poem level*** | |
| How are terms that exhibit emotion distributed within the poem? | Distribution expert |
| Where is there alliteration (rhyme, consonance) in the poem? | Phonics expert |
| Does the poem have a metrical structure? | Rhythm expert |
| How repetitive is the poem? | Repetition expert |
| Does the poem cohere? | Thematic expert |
| Does the poem have a progression? | Narrative expert |
| Where are the various elements of the poem concentrated? | Entropy expert |

Table 2: Questions we imagine a computer would currently be capable of answering when reading a poem

Each of these six points is *dual-voiced* in the sense that the critic is relaying the words of the poem with a new emphasis. Each such statement is one side of a micro-dialogue (Bakhtin, 1984 [1963], p. 73). The challenge is, of course, to bring the observations into the awareness of the computer poet, across the "analogue divide." Care should be taken not just to blythely program the computer with more rules, but rather to give attention to facilitating the process of learning new rules contextually. We continue with the example from this point of view in the following section.

First we will consider a reversal of roles, with the computer in the position of critic, looking at a passage from an historical piece of poetry. We have selected a passage from Robert Burns that might have – but in fact did not – serve as a model for the poem generated above.

*I'm truly sorry man's dominion*
*Has broken Nature's social union,*
*An' justifies that ill opinion*
*Which makes thee startle*
*At me, thy poor, earth born companion*
*An' fellow mortal!*

Naturally, the first problem is for the computer to *read* the poem. One of the approaches that is most appealing from our point of view is the automatic generation of a semantic network from the input text (Harrington and Clark, 2007). We could straightforwardly extend the methods of Harrington and Clark with notions drawn from Table 2.

1. The passage begins with *I*/*me*, locating the *poor, earth born* poet
2. *thee*/*thy* is another person, possibly the reader, who becomes *startled*
3. Singular *I* contrasts with the class *man*
4. *sorry* is a sad emotion
5. *truly* exaggerates sorry
6. *dominion* is large
7. *broken* and *union* are opposites
8. *sorry* and *justifies* are opposites
9. *union*, *companion*, and *fellow* are positive words about relationship and joining
10. *broken*, *ill*, *poor*, *startle* and *mortal* are related to frailty
11. *born* and *mortal* are related
12. There are a lot of rhymes in the poem, at the end of the lines, enjambed.

These comments are very different from the other reading above, and are differently interesting.

We've demonstrated that the computer is capable of asking objective questions of a poem. A similar semantic network approach would allow it to listen to feedback and take it on board, even when it doesn't understand the ways of thinking that generate this feedback. Again, this links the process of reading and writing poetry to a process of dialogue.

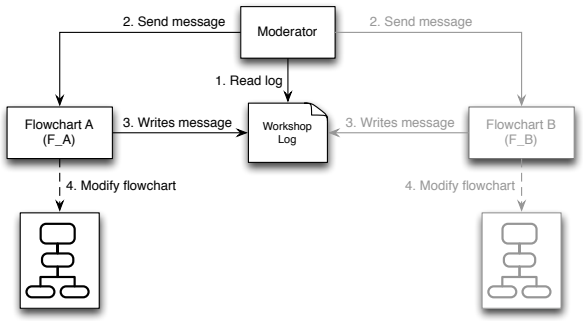



Figure 2: Schematic diagram for a workshop built in the FloWr system

## Seeds for a FloWr Garden

Keeping in mind the current limitations of FloWr – no looping or conditionals, only running one flowchart at a time and in one direction – a *conversation* between ProcessNodes or flowcharts is not immediately feasible. Figure 2 represents a hypothetical design in which a Workshop could take place with a minimally-altered version of FloWr. As shown in Figure 2, each participant in the Workshop would be represented by a single node. One of these nodes is a *moderator* in charge of dictating the interaction between the participants of the Workshop, while the rest represent *flowcharts* that have the ability to modify their own connections according to the discussions from the Workshop – this can be achieved by exploiting the scripting mechanism of FloWr and dynamically loading the new structure of the flowchart. Moreover, a shared *log* would contain the history of the messages exchanged during the Workshop and a queue of messages waiting to be delivered. We define four different types of messages that can be exchanged:

- ***comments*** about specific elements of a poem, or more general statements about how the poem affects this reader.
- ***questions*** to facilitate comprehension of this commentary; for instance, the questions can vary from simple requests of sources of information (e.g. files, input from another node, which resources a flowchart uses, etc.) to process-specific details (e.g. current conditions, purpose, other outstanding questions, etc.)
- ***answers*** would be associated to previous questions and may contain simple text such as an url for the source of information, or a piece of script representing a node used by a flowchart.
- ***suggestions*** are changes proposed by one participant to another. Similar to the answer, this can be as simple as suggesting the change of an information source, or more complex, such as suggesting the replacement of a node for an alternative node.

A Workshop session follows this communication protocol:

1. The moderator initialises an empty log and sends a message to the flowcharts to indicate that the session has started.
2. The flowcharts start writing messages in the log.
3. The moderator checks the current state of the workshop by reading the log.
4. The moderator selects the next message in the queue and passes it to the target flowchart.
5. The flowchart reads the message and acts accordingly, by either (*i*) modifying its connections or; (*ii*) sending a message back, i.e., writing to the log.
6. Step 3 is repeated until no further message are left in the queue.

**Example.** Figure 3 shows the poetry generator flowchart that generates the poem about the "demon dog" presented above. The flowchart uses two linguistic resources: ConceptNet (Liu and Singh, 2004), a semantic network of common knowledge, and Disco (Kolb, 2008), a semantic similarity words retrieval system. Let us assume the human critic **A** has access to the system through a "UI flowchart" like a Read-Eval-Print Loop (REPL), and the poetry generator **B** is mainly concerned with maintaining a generative flowchart like the one shown in the figure. The following exchange of messages can occur:

1. **Comment from participant A to participant B**: *The words "lonely encounter" and "elusive ruler" in lines 2 and 3 are generalised and imprecise.*
2. **Question from participant B to participant A**: *I identify the processes* Disco3 *and* Disco4 *as the source of the problem. Can you suggest an alternative to Disco?*
3. **Suggestions from participant A to participant B**: *Use WordNet or the Historical Thesaurus to find more expressive and specific terms for the core concepts in the poem; try to link the core concepts together by chaining together related concepts in ConceptNet or WordNet.*
4. **Action executed by participant B**: Receives suggestions, creates several alternative versions of the script, executes them and decides which one is most coherent and which conveys a sense of narrative.

From this exchange, the computer might learn (without ever being explicitly told) that expressive terms and narratives are related, and it might begin to discover a way to produce coherent poems with a narrative structure.

Since the computer has source code instead of a brain, we can use it to do control studies with process. However, in general source code does not uniquely determine process: contextual effects are what make an experiment an experiment. As described in (Cook and Colton, 2013), code may include hints about its expected operating context. This is related to our theme of embedding process within an artefact. In this connection, one extension to FloWr that would help to facilitate dialogue between flowcharts would be to add machine-readable *commentary* to ProcessNodes. Commentaries would label a node's inputs and outputs, describe its basic purpose, and provide information about procedure, conditional behaviour, mappings between processes and elements of a generated poem (like the mapping between Disco3 and "lonely encounter").

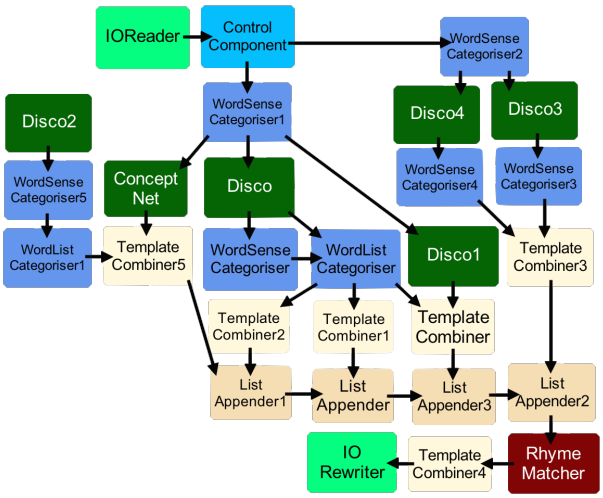



Figure 3: The flowchart that created the "demon dog" poem

Altered versions of a flowchart (Charnley et al., 2014) can be seen as parallel solutions that could be executed and compared on a population basis with respect to some prespecified metrics in order to make an informed decision on which suggestion(s) to follow, as hinted in the last step of the example. In (Colton, Pease, Corneli, Cook, and Llano, 2014) we explored the related idea of modelling system progress over time. Learning new rules contextually would offer one clear measure of progress. Caveat lector: considerable work would be required to realise the ideas we've described in FloWr or any other platform we're aware of.

## Discussion

**Potential applications.** The paradigm advanced in this paper would not remove the "generation" aspects of CC, but would pair them more closely with reflection. The same skills that support learning in a writers workshop may support a form of dialogue with the work itself, leading to richer creative artefacts that show us more about how creativity works. Focusing on social creativity does not suggest that we should devalue works from lone creatives, but it does suggest that we think about how we knit individuals together in the social fabric of the CC community. The current model at the International Conference for Computational Creativity (ICCC) is similar to many other academic conferences: we present *our* work to one another and build *our* sense of community in that way. But what about a track for computers to present their work? The idea of computers interacting in a workshop-like setting is not unprecendented. As Turing (1951) foresaw, computational software has become highly competent at Chess and reasonably competent at Go, partly through continuous practice pitting programs against each

other. Poetry could be approached in a similar way, reviving the *floral games* of the troubadours. Other creative arts may also be amenable to the same sort of approach. Gabriel mentions "brainstorms, critiques, charrettes, pair programming, open-source software projects, and even master classes" (Gabriel, 2002, p. 11). The sort of thinking we have developed here might be adapted to contexts like these.

**Potential criticisms.** It can be challenging and time-consuming to invite and process feedback, and the Workshop would often be seen as unnecessary for standardised production cycles that can already produce artefacts that are "good enough." Furthermore, since we often seem to *get* the computer to do just what we have in mind when we're programming, it might not make sense to treat it as a distinct other and invite it to participate in a dialogue. (Some REPL users may disagree, and may already think of programming as a dialogue.) From our read/write perspective on computational creativity, the most immediate problem is that appreciation of works of art is rather hard. Consider the difference between creating a video game (for example) and playing a video game. In the first case, the designer has full control over the rule-set, game mechanics, interaction devices and so forth. At least one computational video game designer can play its own games (Cook, Colton, Raad, and Gow, 2013), and an experiment shows that it is possible for an artificial game player to learn how to play classic video games using reinforcement learning, starting from raw pixels (Mnih et al., 2013) – but both are quite far from general-purpose game playing. This is itself a topic of contemporary research, and it serves to illustrate that coping with feedback is a major challenge for AI research. Finally, we are not in a position to make strong claims about the quality of workshopped artefacts, although our experience with poetry has shown us that high-quality poems are often exactly the ones which teach us about the creative process. We hope future research will explore this connection further.

## Conclusions

The ideas of social interaction, feedback, and evaluation have frequently been discussed in CC, but implementation and theorisation around these topics have been more limited. In the current paper, we suggest giving artefacts more agency, designing computer programs with more autonomy, and focusing research effort on creative evolution. We have shown that in principle computers can engage in dialogue about poems, which points to a theory of poetics rooted in the making of boundary-crossing objects and processes. In order to move from thought experiment to computational simulation, FloWr could be helpfully extended with further programmer facilities including loops, subroutines, and commentaries, along with the ability to generate-and-test in a population-based manner, and the ability to learn over time. Workshops and related approaches are suitable for autonomous learning and development of the creative process, but they face technical and also some theoretical limitations. Dialogue may offer a way to creatively push these limits, empowering both programs and programmers.

## Acknowledgements

This research has been supported by EPSRC grants EP/L00206X and EP/J004049, and the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant numbers: 611553 (COINVENT) and 611560 (WHIM).

## References

Anderson, D. R., and Hausman, C. R. (1992). The Role of Aesthetic Emotion in RG Collingwood's Conception of Creative Activity. *Journal of Aesthetics and Art Criticism*, 299–305.

Baars, B. J. (1997). In the Theater of Consciousness: The Workspace of the Mind. *Journal of Consciousness Studies*, *4*.

Bakhtin, M. (1984 [1963]). *Problems of Dostoevsky's poetics*. Translated by Caryl Emerson. University of Minnesota Press.

Bakhtin, M. (2010 [1986]). *Toward a Philosophy of the Act*. University of Texas Press.

Bergson, H. (1911 [1907]). *Creative evolution*. Henry Holt and Company.

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.

Charnley, J., Colton, S., and Llano, M. T. (2014). The FloWr framework: automated flowchart construction, optimisation and alteration for creative systems. In D. Ventura, S. Colton, N. Lavrac, and M. Cook (Eds.), *Procs. of the 5th International Conference on Computational Creativity*.

Colton, S., Bundy, A., and Walsh, T. (2000). Agent based cooperative theory formation in pure mathematics. In *Procs. of AISB symposium on creative and cultural aspects & applications of AI and cognitive science*.

Colton, S., Pease, A., Corneli, J., Cook, M., and Llano, T. (2014). Assessing Progress in Building Autonomously Creative Systems. In D. Ventura, S. Colton, N. Lavrac, and M. Cook (Eds.), *Procs. of the 5th International Conference on Computational Creativity*.

Colton, S., and Wiggins, G. A. (2012). Computational Creativity: The Final Frontier? In *Procs. of 20th European Conference on Artificial Intelligence (ECAI)* (pp. 21–26). Montpellier, France.

Cook, M., and Colton, S. (2013). From Mechanics to Meaning and Back Again: Exploring Techniques for the Contextualisation of Code. In *Procs. of the AIIDE Workshop on Artificial Intelligence and Game Aesthetics*.

Cook, M., Colton, S., Raad, A., and Gow, J. (2013). Mechanic Miner: Reflection-Driven Game Mechanic Discovery and Level Design. In *Procs. of EvoGames Workshop*.

Corneli, J., and Jordanous, A. (2015). Implementing feedback in creative systems: a workshop approach. arXiv:

1505.06850 [cs.AI]. Retrieved from http://arxiv.org/abs/1505.06850

Corneli, J., Pease, A., Colton, S., Jordanous, A., and Guckelsberger, C. (2015). Modelling serendipity in a computational context. arXiv: 1411.0440 [cs.AI]. Retrieved from http://arxiv.org/abs/1411.0440

Csikszentmihalyi, M. (1988). Society, culture, and person: a systems view of creativity. In R. J. Sternberg (Ed.), *The Nature of Creativity* (Chap. 13, pp. 325–339). Cambridge, UK: Cambridge University Press.

Dewey, J. (1958 [1934]). *Art as experience*. Capricorn Books.

Gabriel, R. P. (2002). *Writer's Workshops and the Work of Making Things: Patterns, Poetry...* Addison-Wesley Longman Publishing Co., Inc.

Gervás, P. (2010). Engineering Linguistic Creativity: Bird Flight and Jet Planes. In *NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*. Los Angeles: Association for Computational Linguistics.

Gervás, P., and Leon, C. (2014). Reading and writing as a creative cycle: the need for a computational model. In D. Ventura, S. Colton, N. Lavrac, and M. Cook (Eds.), *Procs. of the 5th International Conference on Computational Creativity*.

Harrington, B., and Clark, S. (2007). ASKNet: Automated Semantic Knowledge Network. In A. Howe, and R. Holte (Eds.), *Procs. of the Twenty-Second AAAI Conference on Artificial Intelligence* (Vol. 2, pp. 889–895). AAAI Press.

Jordanous, A. (2011). Evaluating Evaluation: Assessing Progress in Computational Creativity Research. In *Procs. of the Second International Conference on Computational Creativity (ICCC-11)*. Mexico.

Jordanous, A. (2015). Four PPPPerspectives on computational creativity. In *Procs. of the AISB Symposium on Computational Creativity*.

King, S. (2000). *On writing*. Simon and Schuster.

Kirke, A., and Miranda, E. (2013). Emotional and Multiagent Systems in Computer-aided Writing and Poetry. In *Procs. of the Artificial Intelligence and Poetry Symposium (AISB'13)* (pp. 17–22). Exeter University, Exeter, UK.

Kolb, P. (2008). DISCO: A Multilingual Database of Distributionally Similar Words. In *Procs. of the 9th Conference on Natural Language Processing*.

Liu, H., and Singh, P. (2004). Commonsense Reasoning in and Over Natural Language. In *Procs. of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems* (Vol. 3215). LNCS. Springer.

Manurung, R., Ritchie, G., and Thompson, H. (2012). Using Genetic Algorithms to Create Meaningful Poetic Text. *Journal of Experimental and Theoretical Artificial Intelligence*, *24*, 43–64.

McGraw, G., and Hofstadter, D. (1993). Perception and creation of diverse alphabetic styles. *AISB Quarterly*, *85*, 42–49.

Minsky, M. (1967). Why programming is a good medium for expressing poorly understood and sloppily formulated ideas. *Design and Planning II-Computers in Design and Communication*, 120–125.

Minsky, M. (1988). *The Society of Mind*. New York: Simon and Schuster.

Minsky, M. (2006). *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. New York: Pantheon.

Misztal, J., and Indurkhya, B. (2014). Poetry generation system with an emotional personality. In *Procs. of 5th International Conference on Computational Creativity*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv: 1312.5602 [cs.LG]. Retrieved from http://arxiv.org/abs/1312.5602

Montfort, N., Pérez y Pérez, R., Harrell, D. F., and Campana, A. (2013). Slant: A Blackboard System to Generate Plot, Figuration, and Narrative Discourse Aspects of Stories. In *Proceedings of the fourth international conference on computational creativity*.

Pease, A., Guhe, M., and Smaill, A. (2010). Some Aspects of Analogical Reasoning in Mathematical Creativity. In *Procs. of the International Conference on Computational Creativity* (pp. 60–64). Lisbon, Portugal.

Pérez y Pérez, R., Aguilar, A., and Negrete, S. (2010). The ERI-Designer: A Computer Model for the Arrangement of Furniture. *Minds and Machines*, *20*(4), 533–564.

Poincaré, H. (1929 [1908]). Mathematical Creation. In *The Foundations of Science: Science and Hypothesis, The Value of Science, Science and Method* (Chap. III of Book I. Science and the Scientist, Vol. Science and Method, pp. 383–394). New York: The Science Press.

Rhodes, M. (1961). An analysis of creativity. *Phi Delta Kappan*, *42*(7), 305–310.

Saunders, R. (2012). Towards Autonomous Creative Systems: A Computational Approach. *Cognitive Computation*, *4*(3), 216–225.

Seikkula, J., and Arnkil, T. E. (2014). *Open Dialogues and Anticipations – Respecting Otherness in the Present Moment*. Helsinki: National Institute for Health and Welfare.

Sosa, R., Gero, J., and Jennings, K. (2009). Growing and destroying the worth of ideas. In *Procs. of the 7th ACM Creativity and Cognition conference* (pp. 295–304). Berkeley, California.

Turing, A. (1951). Intelligent machinery, a heretical theory. A lecture given to '51 Society' at Manchester.

Von Foerster, H. (2003 [1979]). Cybernetics of cybernetics. In *Understanding understanding* (pp. 283–286). Springer.

Whitehouse, P. (1978). The Meaning of 'Emotion' in Dewey's Art as Experience. *Journal of Aesthetics and Art Criticism*, 149–156.

# Interaction Evaluation for Human-Computer Co-creativity: A Case Study

**Anna Kantosalo, Jukka M. Toivanen, Hannu Toivonen**
Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
anna.kantosalo@helsinki.fi, jukka.toivanen@cs.helsinki.fi, hannu.toivonen@cs.helsinki.fi

**Abstract**

Interaction design has been suggested as a framework for evaluating computational creativity by Bown (2014). Yet few practical accounts on using an Interaction Design based evaluation strategy in Computational Creativity Contexts have been reported in the literature. This study paper describes the evaluation process and results of a human-computer co-creative poetry writing tool intended for children in a school context. We specifically focus on one formative evaluation case utilizing Interaction Design evaluation methods, offering a suggestion on how to conduct Interaction Design based evaluation in a computational creativity context, as well as, report the results of the evaluation itself. The evaluation process is considered from the perspective of a computational creativity researcher and we focus on challenges and benefits of the interaction design evaluation approach within a computational creativity project context.

## Introduction

Evaluation is vital for guiding the development and measuring progress in computational creativity methods (Jordanous 2012). Especially formative feedback is needed to guide practical development work (Jordanous 2012). This is also true for interactive systems based on computational creativity methods, including human-computer co-creative systems – systems in which both the human and the computer take creative responsibility of the output of the program. As new human-computer co-creative systems are created we will need to address issues in their evaluation.

Bown (2014) argues for a more contextually based evaluation of creative systems within their cultural environments. We consider this to be true especially for human-computer co-creative systems as an evaluation focusing merely on the computational system's creativity is not sufficient to evaluate the success and progress of the system with regard to the user's creative process or the co-creative experience itself. Methods incorporating the user's perspective are needed for incorporating these aspects. Bown (2014) suggests learning from contextually and culturally aware evaluation methods intended for end-user evaluation established in the field of Interaction Design.

In this study paper, we first briefly discuss the similarities and differences between human-computer co-creativity evaluation and computational creativity evaluation. We then proceed to view Interaction Design in the context of computational creativity: We see how Interaction Design currently connects to computational creativity and view previous human-computer co-creation and creativity support system evaluation projects in the light of the DECIDE framework (Rogers, Sharp, and Preece 2011). Then, we move on to discuss our own case study of the Poetry Machine evaluation and illustrate how the DECIDE framework works in practice in the context of a human-computer co-creativity system evaluation. Next, we present the results of our evaluation case study and finally discuss our findings and the usefulness of this evaluation with regard to computational creativity development.

## Evaluating Computational Creativity and Human-Computer Co-Creativity

Evaluation of computationally creative systems may focus on different levels of the system: According to Colton and Wiggins (2012), a distinction is often made between evaluating the "cultural value of the artefacts produced by systems, and tests which evaluate the sophistication of the behaviours exhibited by such systems". Jordanous (2012) supports a similar idea in her analysis of existing evaluation frameworks. According to Yannakakis et al. (2014), this characterization of evaluation also applies for the evaluation of co-creativity. Yannakakis et al. continue that the evaluation of the final outcomes of a co-creative process may utilize same approaches as the evaluation of the outcomes of an independent computationally creative process but the process itself is more difficult to evaluate because of the unknown nature of the human creativity process itself. In this paper, we have focused on the evaluation of the process aspects and left out the evaluation of the artefacts. However, the evaluation of artefacts can also factor into evaluating the effects and benefits of the co-creative system to its users.

Jordanous (2012) notes that computational creativity evaluation has traditionally favored expert evaluation, although the evaluation of computational creativity systems with target users has been discussed. There are still few practical examples describing the end-user-evaluation of either autonomously creative or co-creative systems. In this paper, we hope to provide the field with a practical example of how

end-user evaluation of computational creativity software involving users can be conducted in practice at early development stages.

One important difference between evaluating autonomous computational creativity systems and human-computer co-creative systems seems to be that the subjective experience of the human user of a co-creative system becomes an interesting evaluation target. Therefore, the focus of evaluating co-creative systems can not be only on evaluating the creativity of the system, but also in part on the effects the system has on the user. Yannakakis et al. (2014) conclude that the interaction between the human and the computer fosters the creativity of the tool, but the claim cannot be thoroughly evaluated with current frameworks.

Finally, Jordanous (2012) divides the evaluation of computational creativity systems to summative and formative evaluation. The purpose of the former is to provide a summary of a system's creativity, while the latter aims to provide constructive feedback on the system. A similar distinction is made by Hartson et al. (2003) for Interaction Design evaluation methods, with the distinction that formative evaluation is usually done iteratively during product design and summative evaluation is usually reserved for finished designs or comparisons between designs. Jordanous (2014) seems to consider formative evaluation a more important goal for current computational creativity evaluation procedures, as she regards the usefulness of evaluation results as an evaluation criteria for evaluation methods themselves. This paper focuses on the formative evaluation of an on-going project, aiming to produce results that are useful for guiding the future development of the poetry writing tool.

## Interaction Design and Evaluation in Computational Creativity Contexts

The field of Interaction Design studies how to best design interactive products to facilitate human interaction and communication. As such, it seems ideal for designing human-computer co-creative tools. Interaction Design covers a multitude of design fields and approaches, such as user-centered design (Rogers, Sharp, and Preece 2011). As a methodological framework it offers iterative processes and methods for designing and evaluating interaction in specific contexts. Some Interaction Design methods have already been used in designing software based on Computational Creativity methods (Kantosalo et al. 2014).

Bown (2014) argues that the wide range of robust Interaction Design methods for observing and measuring user experience could help build a thorough empirical grounding for Computational Creativity evaluation. He continues that Interaction Design would also help to establish commonly used evaluation concepts – 'value' and 'novelty' – as constructs immediately related to the goals of the individual user. This new human-centered approach would shift the nature of the enquiry very slightly "by asking not how creative a system is, or whether it is creative by some measure, but how its creative potential is practically manifest in interactions with people."

In this section, we provide a brief review of Interaction Design evaluation in creative contexts. We cover human-computer co-creativity projects STANDUP (Waller et al. 2009), Scuddle (Carlson, Schiphorst, and Pasquier 2011), Evolver (DiPaola et al. 2013), and the Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014). They all have used evaluation methods that can be seen to fall within the scope of Interaction Design. To learn more about how the creative context should be considered in Interaction Design evaluation, we include six creativity support systems that have been evaluated in the literature: the IdeaManager (Shibata and Hori 2002), a Virtual Musical Environment (VME) (Johnston, Amitani, and Edmonds 2005), the Envisionment and Discovery Collaboratory (EDC) (Warr and O'Neill 2007), the Choreographer's Notebook (Singh et al. 2011), Ugobes Pleo (Ryokai, Lee, and Breitbart 2009), and Parallel Pies (Terry et al. 2004).

We structure the review, and our subsequent description of how we evaluated the Poetry Machine, according to the DECIDE framework by Rogers et al. (2011). The DECIDE framework is a checklist with the following six items:

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

Each step of the framework guides the next step: Determining goals helps designers to ask relevant study questions, and questions guide the selection of methodologies. Then again, the selected methods predict some of the practical issues, which may be related to ethical questions. Finally, all previous factors are relevant to deciding how the data is best evaluated, analyzed, interpreted, and presented.

### Determining Evaluation Goals

Choosing what to evaluate is often a challenge in the creative domains. Some projects attempt to measure the increase in creativity of the user, some discuss the creativity of the system, while some focus on user experiences and feedback. Carroll (2011) has noted that because creativity is difficult to define, it is often difficult to say if tests designed to measure creativity of an interactive system actually measure creativity or some other construct. Additionally, aspects of creativity may be domain specific (Carroll 2011).

It is surprising that only two of the reviewed human-computer co-creativity evaluation projects state their goals explicitly: Waller et al. (2009) investigated if their target group is capable of using the STANDUP system, and how they use it. Yannakakis et al. (2014) studied if the Sentient Sketchbook fostered the designer's creativity, specified as aspects of lateral thinking and diagrammatic reasoning. In evaluations of creativity support tools, goals have included gathering initial feedback (Johnston, Amitani, and Edmonds 2005), evaluating if the tool supports specific aspects of a creative process (Warr and O'Neill 2007; Singh et al. 2011), or what is the role of the tool in a creative process (Ryokai, Lee, and Breitbart 2009).

## Exploring the Questions

Exploring the questions means the redefinition and focus of the goals to more operational questions (Rogers, Sharp, and Preece 2011). Among the Human-Computer Co-Creativity evaluation examples, only Yannakakis et al. (2014) further explain their evaluation targets as the degree and quality of use of the suggestions of a computational partner. As a type of elaboration for their implicit goals DiPaola et al. (2013) provide the set of actual questions used in their study. Among the creativity support systems, Singh et al. (2011) provide a similar list of questions asked from their users and Johnston et al. (2005) list the specific behaviors of the system they want to investigate.

## Choosing Methods

There is a wide range of Interaction Design evaluation methodologies, including formal vs. informal testing methods, thinking aloud vs. observation, and summative vs. formative testing (Lewis 2006). It is common for designers to combine different methods to gather rich data (Rogers, Sharp, and Preece 2011). Mixed-methods approach combining quantitative and qualitative data is also the evaluation recommendation of the NSF Workshop on Creativity Support Tools (Carroll 2011).

The selection of Interaction Design methods is affected by multiple factors: Firstly, the purpose of the evaluation, context of use, and type of data to be gathered matter (Rogers, Sharp, and Preece 2011). Secondly, practitioners must consider the reliability, thoroughness and validity of methods (Hartson, Andre, and Williges 2003). Finally, a number of case based issues contribute to the selection, such as cost efficiency and the target group.

All of the studied projects described the methods used in the study but not necessarily the rationale behind their selection. Notably three of the projects, including the Sentient Sketchbook, the Idea manager, and Choreographer's Notebook, used remote methods, including the collection of usage logs to determine the quantity of use or usage patterns. Shibata and Hori (2002) explained they needed longitudinal remote data collection because creativity is dependent on the context and environment of the users and thus impossible to study in a laboratory setting. Nearly all laboratory studies seem to have strived to simulate creative situations for the users, with the exception of STANDUP and Evolver.

Methods have also been applied to creative contexts in different ways. For example, the tasks used in the evaluation are very differentiated, some evaluations having more explorative tasks with only a general goal (e.g. Pleo and Parallel Pies), while others used more specific tasks with scripted roles for the participants (e.g. EDC).

## Identifying Issues

Regardless of the chosen methods, all methods require representative participants, representative tasks and representative environments in which participants are observed (Lewis 2006). These dimensions define most of the practical issues related to any Interaction Design evaluation, and were not absent from the example projects either. For example, finding suitable users was difficult for the STANDUP project (Waller et al. 2009).

The creative context also proposes some additional issues to evaluation: Experiences from creativity support tool evaluation show that errors in the interfaces may sometimes provide additional opportunities for the users, and that spending significant times at a task may indicate immersion, not poor quality of interaction (Carroll 2011). Therefore, some metrics loaned from Interaction Design may not suit the evaluation in the creative setting (Carroll 2011).

The novelty and value of artefacts produced by creative systems become highly dependent on user and context in creativity contexts, as suggested by Bown (2014): For instance, Shibata and Hori (2002) studied a creativity support tool intended to catalyze idea generation. They had their users to evaluate the novelty and practicality of the ideas for themselves, instead of trying to assign objective values to the produced ideas.

## Ethics

As with all human studies, ethical issues require specific care with Interaction Design evaluations involving users. Very few specific ethical issues were reported in the example studies, and in general they were unrelated to creativity: Waller et al. (2009) report issues related to child participants and Warr and O'Neill (2007) note the use of consent forms and stress to users that they are evaluating the software, not the users.

## Analysis and Presentation

The chosen methods define the type of data collected to a great extent but researchers still have to choose how to analyze and present the data, as well as account for its validity, generalizability and scope (Rogers, Sharp, and Preece 2011). Many of the sample cases focus on the creative process and key interactions related to it in their analysis: Yannakakis et al. (2014) analyzed use patterns from log files and identified important process milestones from them with the help of the user provided qualitative data. Singh et al. (2011) also analyzed logs noting key changes in the creative processes by presenting rationale for the use. Warr and O'Neill (2007) recognized different sub-activities and key interactions in the idea generation process of their users based on video logs. Ryokai et al. (2009) illustrated the process through a detailed example and Carlson et al. (2011) focused on process related user quotes. As a semi-process oriented reporting approach Waller et al. (2009) focused on analyzing interaction paths and Terry et al. (2004) analyzed how well the interaction model enhanced the workflow of their users.

Feedback plays a great part in most of the evaluation projects; Waller et al. (2009), Johnston et al. (2005), Shibata and Hori (2002), Warr and O'Neill (2007), and Terry et al. (2004) report new ideas for improvement. Most projects also used user quotes to illustrate key findings or feedback; only Yannakakis et al. (2014), Warr and O'Neill (2007), and Terry et al. (2004) do not use user quotes at all.

## Evaluation of the Poetry Machine

The Poetry Machine (Kantosalo et al. 2014) aims to solve the problem of the empty paper for its users, school children studying poetry or just exercising writing. The user selects a theme (in the tested version one out of 8 options), and the Poetry Machine provides a draft poem consisting of poetry fragments. The editing interface simulates a set of fridge magnets. The user can edit the draft by dragging words and rows around, removing them, or entering new ones. The user can also ask for further assistance from the computer, by using a feature called the "robot". By dragging words or rows on the robot, the robot provides the user with similar fragments or rhyming words.

The Poetry Machine has been developed at the University of Helsinki, based on the poetry generation methods developed earlier in the group (Toivanen et al. 2012). However, the version evaluated in this paper does not utilize the full functionality of these methods. Instead we decided to use simple fragment based approaches to provide pieces of poetry and rhyme candidates that can be expanded to full poems by users of the system. The Finnish poetry fragments and rhyme dictionaries are automatically extracted from a corpus containing children's literature from Project Gutenberg. This simplistic setting makes it easier to assess the effectiveness of the current interface of the system and also provides a basic setting for further iterative testing.

### Planning the Evaluation

In the next paragraphs we describe the evaluation process of the Poetry Machine through the DECIDE framework.

**Determining Evaluation Goals** We selected three goals for the evaluation of the Poetry Machine: (1) discovery of usability problems, (2) evaluation of its usefulness, and (3) evaluation of its enjoyability. The first goal is a typical Interaction Design evaluation goal, yielding concrete remarks on how to improve the interface. In this case, eliminating usability problems is a vital step before conducting additional, comparative testing on the contents of the co-creation. The second goal, usefulness, is defined here as the system's ability to make creative writing easier for children. Finally the last goal, enjoyability, is related to the ISO-9241-11 (ISO/IEC 2010) satisfaction parameter, but combined with fun, which with child users correlates with usability (Sim, MacFarlane, and Read 2006).

**Exploring the Questions** In the question exploration phase, each goal was elaborated with a set of sub-questions, which could be more easily approached with specific Interaction Design evaluation methods. Our primary study questions were:

1. Usability
   (a) Are children able to use the program?
   (b) Is the interface graphically pleasing to children?
2. Usefulness
   (a) What features of the program are the most useful for children?
   (b) Does the program make creative writing easier for children?
3. Enjoyability
   (a) Do children exhibit negative signs, such as signs of boredom or frustration, when using the program?
   (b) Do children exhibit positive signs, such as smiling, or willingness to continue the activity for a longer period of time?
   (c) What activities do children name when asked about the most fun/boring features in the program?

Most of the questions can be further divided into sub-sub-questions, such as "Do children use all of the features or only few?".

We intentionally excluded questions, such as "Does the tool promote learning or creativity?". These questions were considered outside the scope of the first evaluation, but more experiments are planned for evaluating the pedagogical potential of the tool, and alternatives for promoting creativity.

**Choosing Methods** In order to gather a wide range of feedback, we decided to use a mixed-methods approach with two methods: Peer Tutoring and a small group session we call Group Testing. We chose the paired Peer Tutoring composition proposed by Edwards and Benedyk (2007) in which two users work as a pair – the first participant first learns the use of the tool and then teaches it to his or her partner. In the Group Testing we simulated a small group teaching scenario with one teacher teaching a group of five pupils on how to write a poem with the Poetry Machine. By using the methods in a school environment, we attempted to imitate some culturally and contextually aware conditions.

Peer Tutoring was selected as it has been previously used with young children in usability tests organized at school. It offers a natural context for using the tool with a friend, diminishing biases resulting from an unbalanced adult-child relationship between the users and the researchers administering the test (Höysniemi, Hämäläinen, and Turkki 2003). It is also good for eliciting comments from children (Edwards and Benedyk 2007), as well as for fostering creativity, experimentation and problem solving-skills within the test situation (Höysniemi, Hämäläinen, and Turkki 2003). Group Testing allowed us to observe the use of the system in a more authentic, teacher driven learning situation.

Observation of behavioral signs is considered more trustworthy than self reports in the case of children (Hanna, Risden, and Alexander 1997), and it is used in both methods to provide both quantitative and qualitative data. To collect more qualitative data, both methods were coupled with an appropriate background questionnaire and a post task debriefing. With Peer Tutoring we used a paired interview. For the Group Testing we developed a group-based, game-like, feedback gathering method called Feedback Game (Kantosalo and Riihiaho 2014).

Each of our six Peer Tutoring sessions started with *tutor introduction*: The researchers presented themselves to the tutor pupil, and the facilitating researcher helped him or her to fill a background questionnaire. During the next step, *tutor training*, the tutor was encouraged to explore the tool and

write a poem with it. Next, during *tutee introduction*, the tutee was introduced to the test setting and filled the background questionnaire, while the tutor read a book. This was followed by the actual *peer tutoring* phase, during which the tutor guided the tutee in writing a poem with the tool. Finally the tutor and the tutee were interviewed in what we call the *pair interview* phase.

Both Group Testing sessions started with an *introduction* phase, during which the participating children filled in the same background questionnaire as the Peer Tutoring participants. This was followed by *instruction by the teacher*, during which the teacher shortly composed a poem at the front of the classroom explaining the use of the tool. We then moved on to the *poem writing* phase, during which each child composed a poem, the teacher instructing them when necessary. Feedback from the children was then gathered in the *the Feedback Game* phase. In the game children answered questions like "Was it fun to use the poetry tool?" on a five step Likert scale turned into a gameboard. Each question was followed by a round of arguments. Finally a separate *teacher interview* was conducted to learn how the teacher perceived the effects of the tool on his class.

**Identifying Issues** As a sensitive user group children require specific care in selecting and applying test methods. Both, the Peer Tutoring test and the Group Testing were conducted on site, in a small classroom at a local Finnish school. To gather enough material to make for possible test session failures, we decided to work with a fairly large group of children. We recruited a class of 9-10-year-old pupils. Their teacher selected 22 participants (12 for Peer Tutoring, 10 for Group Testing) according to criteria provided by us. The sample is large, but narrow, which is somewhat typical for Interaction Design evaluation with children (see e.g. the sample sizes in (Sim, MacFarlane, and Read 2006) or (Höysniemi, Hämäläinen, and Turkki 2003)). Further testing with more varied users is planned.

To ensure unintrusive data collection we videotaped each session, and the researcher acting as the main facilitator in charge of interviewing and helping was accompanied by one or two additional observers, who were present at all times. Additionally we performed automatic data collection of the artefacts produced by the children, including recording which words in each poem were computer generated.

To promote creative thinking, we decided to use a very generic test task — the general goal of "writing a poem". In Peer Tutoring, this proved very difficult for some of the tutors, who were unfamiliar with poetry and required thus more guidance, such as suggesting a topic in one case. The tutees seemed to respond to the task more positively, possibly due to peer presence. We were also worried the tutors might try to push tutees to a specific creative direction during testing and discouraged this by allowing only the tutee direct access to the mouse and keyboard during the peer tutoring. We were happy to see the tutors seldom did anything to affect the creative content of their tutee's poem. The same open task worked well with the Group Testing participants.

**Ethics** As the participants of the study were all underaged, we requested a permission from the guardians of each pupil with a letter sent to them through the school. Additionally, we emphasized the volunteer nature of the study at the beginning of each session, explained the secrecy of all raw material, and noted we were there to recruit the pupils' help, not to evaluate them. During two of the Peer Tutoring sessions we held longer pauses to allow the tutor pupils to take a recess or have lunch before continuing with their tutee.

**Analysis and Presentation** All sessions were analyzed from videotaped material. All peer tutoring session videos were analyzed by two researchers; the facilitator and one observer. Each Group Testing session video was analyzed by the facilitator. Additionally field notes were used to note important factors during testing. The facilitator counted instances of use for each feature from the Peer Tutoring videos, as well as positive and negative gestures. Both facilitator and observer additionally observed the tape for interesting comments, actions and usability problems. The problem listings obtained were combined and duplicates were merged into single problems. Each problem was rated by frequency and assigned a severity rating. It was not possible to conduct an equally robust analysis of the Group Testing sessions, because of limitations in taping each participant individually. More general observations were made instead. The pair interviews and Feedback Game sessions were transcribed and the transcripts analyzed for common elements and improvement ideas.

## Evaluation Results

The analysis revealed a number of interesting issues related to the evaluation goals and user ideas for improving the tool. Additionally we were able to find some interesting elements related to the use patterns and creative processes of the users.

**Usability** We found 82 unique usability problems through the Peer Tutoring tests. The problems ranged from practical interface problems, such as how to move words, and aesthetic problems, such as the appearance of buttons on screen, to more conceptual problems including for example misunderstanding of what publishing a poem means. A solution for each problem was suggested based on the problem's manifestation during testing and improvements are being carried out to allow further testing.

**Enjoyability** The enjoyability of the tool was evaluated based on gestures recorded from the Peer Tutoring videos and user comments. All of the six girls, who participated in the Peer Tutoring tests, seemed to show more negative gestures than positive when composing a poem. Four of the six boys however showed more positive signs. This could be taken as an indication of a generally negative reception for the prototype, however there is some ambiguity in interpreting gestures of children: Hanna et al. (1997) consider frowning a negative sign, but during testing this seemed rather to be a sign of concentration, which according to Read et al. (2002) should be considered as a positive sign. Also, as Carroll points out (2011), these signs may have to be interpreted differently due to the creative context. If we interpret these possible signs of concentration as positive, only one pupil displayed more negative gestures during testing. Most of the

negative comments heard during testing had to do with the ambiguity of the task: some children were unsure of what poems are and how to write one. Other negative comments heard during the Peer Tutoring indicated usability problems, and in one case disapproval of the concept itself. Less negative comments were heard during the Group Testing, where children received more clear instructions from their teacher.

The interview and Feedback Game results support a more positive response to the tool: All Peer Tutoring participants gave great scores for the prototype (4 or 5 stars out of 5), 5 out of 12 stating reasons related to the perceived fun of the tool. Additionally two pupils would recommend the tool to their peers based on fun. All Feedback Game participants agreed the tool was fun, and four of them specifically indicated they were willing to participate in a similar test because writing poems during the test was so fun. Enjoyability is also supported by anecdotal evidence provided by the teacher after the testing, during a later visit to the school, and the general reception children gave to the tool. This includes one child mentioning after a test that she had actually stayed after school as she was so enthusiastic to try the tool out.

**Usefulness** The tool was found useful by both the pupils and their teacher: The pupils clearly responded positively to writing poems with the tool. 12 out of 22 pupils indicated that poem writing with it was fun. Six pupils out of 22 also considered that writing poems with the tool was easier than writing otherwise. One pupil specifically mentioned that existing words given by the computer helped his writing process. The teacher highlighted motivation issues: He considered that the pupils were faster to get to work and more engaged with the program than in a typical lesson. He specifically mentioned that one of the pupils, who usually had difficulties with coming up with ideas for creative writing worked very autonomously throughout the session. The teacher also reported later that one of his pupils had been inspired by the tool to start poem writing as a hobby.

All pupils were able to write a poem during testing, however two of them seemed to reproduce one written before the test session. Also, some of the tutors required some ideation help for writing their poem and the facilitator suggested a theme for them, helping the process along with some open questions.

No formal evaluation of the educational value of the tool was made and children were not asked to specifically evaluate the learning potential of the tool, but many of the children considered the tool useful for learning: Seven pupils wanted to recommend the tool to others as they saw it as useful for learning. Three pupils considered autonomously that they had themselves learned to write poems with the tool. The teacher was also able to see the tool as a useful part for future lessons.

**Use Patterns and Creative Process** To gain a better understanding of the use of the tool, we recorded how many times each feature was used by the children during testing. While some of the users were writing with no apparent pattern, the data showed two clear strategies utilized by some of the pupils. The first strategy was to use one of the row-boxes, originally intended to note the row structure in the final poem, as a storage-unit. A pupil using this storage-strategy would shift words within the interface from the operational area to the storage-unit and back according to his or her poem idea. The final poem would consist in a large part of words suggested by the computer. Four participants in the Peer Tutoring test were seen using this strategy. The second strategy, robot-induced-ideation, was seen specifically in one of the pupils. He would primarily engage with the robot, looking always first through its suggestions and only then added a word either invented by the robot or himself.

By looking at the usage data recorded during the use, Peer Tutoring participants wrote shorter poems than the Group Testing participants. The average length of Peer Tutoring participants' poems was 11.6 words (median 11, minimum 6 and maximum 23), while the Group Testing participants wrote 25.4 word poems on average (median 19, minimum 12 and maximum 55). On average, 28% of the final words in the poems written by Peer Tutoring participants were provided by the computer (either in the initial draft, or suggested by the robot tool), while 34% of the words used by Group Testing participants originated from the computer. In both test setups two pupils decided not to use any of the suggestions provided by the computer, while in the Group Testing one participant relied entirely on words suggested by the computer, acting as a sort of an editor. However, the logs do not record all of the effects of the tool to the writing of the children – for example one child said during a Peer Tutoring session that "something came to my mind from this" and pointed to one of the robot's suggestions.

We did not attempt to evaluate the quality of the poems and the possible effect of Poetry Machine on them. A larger sample would be needed, as well as a comparative set of poems, either from the same age group or from earlier poems written by these pupils.

**User Ideas** The user ideas collected during testing are summarized in table 1. Peer Tutoring and Group Testing produced different kinds of ideas. On average one Peer Tutoring session produced one idea, whereas each Group Testing session managed to produce two. The ideas gathered during Group Testing are also more immediately related to the conceptual level of the system, while the Peer Tutoring ideas also address more specific interaction ideas. We discuss the main ideas below.

| | |
|---|---|
| 1 | Inputting multiple words together should be easy |
| 2 | Users should be able to remove all words easily |
| 3 | Proposed words should be more familiar |
| 4 | Proposed words should be more tightly linked to words pointed out by the user |
| 5 | Proposed words could be displayed under the word to be replaced |
| 6 | A quick way to add punctuation is needed |
| 7 | Drafts should have more familiar words |
| 8 | Proposed words should be more related to the topic |
| 9 | Proposed words should have better rhymes |
| 10 | Drafts should have more rhymes |

Table 1: Ideas collected from users during testing

## Using the Results in Developing the Poetry Machine

The usability evaluation results are already used to enhance the interface in order to support test situations in which we focus more on the content of the interactions instead of their fluidity. The initial results will guide our research into the pedagogical potential of the tool, and we will further focus in the development of the tool as a motivating agent.

The use patterns collected show potential principles on the base of which further interaction in the tool can be build to support human-computer co-creativity. For example the storage-strategy should be investigated further as an interaction paradigm in the system. The relationship between the robot-induced-ideation and the quantity of computer provided words in the system should be investigated further in the tests, and means for promoting it could include a more active computational participant.

The feedback provides many possibilities for further development of the computational creativity methods used in the system. Especially the ideas give concrete suggestions as to how the system should be developed further.

(1) Instead of just providing simple fragments without any cohesion between them, methods for adding more coherence between the proposed fragments should be investigated. Here the computer could propose fragments that are well suited to the fragments already proposed and also written by the user. Methods of textual coherence based on vector space models of words and linguistic fragments (Mikolov et al. 2013) or corpus word statistics could be used here to enhance the results.

(2) The quality of the rhymes have room for improvement. Methods for improving the quality of rhymes are many, including metrics based on word length etc. Also many different kinds of rhymes like syllabic rhymes, half rhymes, assonances, consonances, and alliteration could be used to add more variation.

(3) Words suggested by the system could be more familiar to the users. However, the users were not unanimously supporting the use of only familiar words. During group testing, one pupil noted that "there were these words you use more seldomly, so there were a couple I could select for my poem". Therefore, tying the words better to the context, proposing synonyms and antonyms for the words pointed out by the user, and using a mix of more and less typical words a good balance between vocabulary enhancing and supporting words could be attained.

In the future, the system could also be used for teaching metrical systems prevalent in traditional poetry. The computer could, for instance, propose that the user could write a sonnet and then track the number of syllables on each line of the poem. If the number of syllables on some line did not fit the metrical structure of a sonnet the computer could propose changing, for instance, one word on the line to satisfy the metrical constraints.

## Conclusions

We have shown how to conduct an interaction design method based evaluation on a human-computer co-creativity tool called Poetry Machine. The evaluation conducted in this case study has similarities to other evaluation cases of human-computer co-creative tools and creativity support tools. Especially interesting is the varied set of evaluation goals that can be supported through Interaction Design methodologies. In creative contexts however, the selection of methodology seems to be especially important: Mixed methods should be used to gain a varied set of data. Also specific care has to be taken to create a test situation that allows the flow of creativity by either using remote study methods, methods that have been found to suit creative contexts, or setting up the evaluation in a creative environment. Tuning methods for creative contexts also requires selecting suitable tasks for the users to do within the test situation.

A very interesting aspect to Interaction Design evaluation planning and practice within the creative context are the issues faced during testing. It seems that some traditionally used interaction design evaluation measures, such as time, or facial gestures are not useful within a creative context, as some negative signs, such as frowning, may actually indicate positive aspects, such as concentration or immersion instead. Most of the issues related to human-computer co-creativity testing with interaction design evaluation methods still seem to be concerned with typical interaction design evaluation problems, such as selecting suitable users.

The analyzed sample cases revealed that typically the analysis of human-computer co-creativity evaluation results is similar to that of Interaction Design evaluation. For example, quotes are frequently used to illustrate key issues. Interestingly many projects have also focused on how the creative process of the user is supported by the interface. A large part of the cases also provided feedback and improvement ideas.

We have illustrated here how such formative evaluation results can be applied to practical computational creativity development work by providing a list of gathered user ideas and presenting concrete ideas on how to use them for further development. However, a simple listing of the ideas is not enough – to defend design decisions and to tune solutions to actual user needs, we need to look at the qualitative data as a whole.

Based on the projects studied for this paper, it seems interaction design evaluation methods have already taken a place within human-computer co-creativity evaluation and the philosophical foundations of this work are also being laid in the computational creativity community. Through our case study, we have demonstrated in a formalized manner, how to plan and conduct Interaction Design method based evaluation for a human-computer co-creativity tool and how the results can be applied in practice. With this we have shown how interaction design evaluation practices offer an interesting, complementary evaluation approach to human-computer co-creation tools, providing results that can be put to practical development use.

## Acknowledgments

This work has been supported by the Academy of Finland (decision 276897, CLiC) and by the European Commission (FET grant 611733, ConCreTe). We wish to thank the pupils

and teachers who participated in this research, and K. Tiuraniemi and M. Hynninen for participating in the data collection.

## References

Bown, O. 2014. Empirically grounding the evaluation of creative systems: Incorporating interaction design. In *Proceedings of the Fifth International Conference on Computational Creativity*, 112–119.

Carlson, K.; Schiphorst, T.; and Pasquier, P. 2011. Scuddle: Generating movement catalysts for computer-aided choreography. In *Proceedings of the Second International Conference on Computational Creativity*, 123–128.

Carroll, E. A. 2011. Convergence of self-report and physiological responses for evaluating creativity support tools. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 455–456. ACM.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: the final frontier? In *ECAI 2012 : 20th European Conference on Artificial Intelligence*, 21–26.

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*, 40–47.

Edwards, H., and Benedyk, R. 2007. A comparison of usability evaluation methods for child participants in a school setting. In *Proceedings of the 6th International Conference on Interaction Design and Children*, 9–16. ACM.

Hanna, L.; Risden, K.; and Alexander, K. 1997. Guidelines for usability testing with children. *interactions* 4(5):9–14.

Hartson, H. R.; Andre, T. S.; and Williges, R. C. 2003. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction* 15(1):373–410.

Höysniemi, J.; Hämäläinen, P.; and Turkki, L. 2003. Using peer tutoring in evaluating the usability of a physically interactive computer game with children. *Interacting with Computers* 15(2):203–225.

ISO/IEC. 2010. Iso 9241-210 ergonomics of human-system interaction – part 210: Human-centered design for interactive systems.

Johnston, A.; Amitani, S.; and Edmonds, E. 2005. Amplifying reflective thinking in musical performance. In *Proceedings of the 5th Conference on Creativity & Cognition*, 166–175. ACM.

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. 2014. Stepping back to progress forwards: Setting standards for meta-evaluation of computational creativity. In *Proceedings of the Fifth International Conference on Computational Creativity*, 129–136.

Kantosalo, A., and Riihiaho, S. 2014. Let's play the feedback game. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, 943–946. ACM.

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*, 1–8.

Lewis, J. R. 2006. Sample sizes for usability tests: Mostly math, not magic. *Interactions* 13(6):29–33.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Read, J.; MacFarlane, S.; and Casey, C. 2002. Endurability, engagement and expectations: Measuring children's fun. In *Interaction design and children*, volume 2, 1–23. Shaker Publishing Eindhoven.

Rogers, Y.; Sharp, H.; and Preece, J. 2011. *Interaction Design: Beyond Human Computer Interaction.* Wiley, 3rd edition.

Ryokai, K.; Lee, M. J.; and Breitbart, J. M. 2009. Children's storytelling and programming with robotic characters. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, 19–28. ACM.

Shibata, H., and Hori, K. 2002. A system to support long-term creative thinking in daily life and its evaluation. In *Proceedings of the 4th Conference on Creativity & Cognition*, 142–149. ACM.

Sim, G.; MacFarlane, S.; and Read, J. 2006. All work and no play: Measuring fun, usability, and learning in software for children. *Computers & Education* 46(3):235–248.

Singh, V.; Latulipe, C.; Carroll, E.; and Lottridge, D. 2011. The choreographer's notebook: A video annotation system for dancers and choreographers. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 197–206. ACM.

Terry, M.; Mynatt, E. D.; Nakakoji, K.; and Yamamoto, Y. 2004. Variation in element and action: Supporting simultaneous development of alternative solutions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 711–718. ACM.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *International Conference on Computational Creativity*, 175–179.

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.

Warr, A., and O'Neill, E. 2007. Tool support for creativity using externalizations. In *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, 127–136. ACM.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proceedings of the 9th International Conference on the Foundations of Digital Games*.

# Impact of a Creativity Support Tool on Student Learning about Scientific Discovery Processes

**Ashok K. Goel and David A. Joyner**
Design & Intelligence Laboratory, School of Interactive Computing
Georgia Institute of Technology
85 Fifth Street NW, Atlanta, GA 30338
ashok.goel@cc.gatech.edu; david.joyner@gatech.edu;

**Abstract**

Science education nowadays emphasizes authentic science practices mimicking the creative discovery processes of real scientists. How, then, can we build creativity support tools for student learning about scientific discovery processes? We summarize several epistemic views of ideation in scientific discovery and find that the ideation techniques provide few guarantees of correctness of scientific hypotheses, indicating the need for supporting hypothesis evaluation. We describe an interactive tool called MILA−S that enables students to elaborate hypotheses about ecological phenomena into conceptual models and evaluate conceptual models through agent-based simulations. We report on a pilot experiment with 50 middle school students who used MILA−S to discover causal explanations for an ecological phenomenon. Preliminary results from the study indicate that use of MILA−S had a significant impact both on the creative process of model construction and the nature of the constructed models. We posit that the computational support for model construction, evaluation and revision embodied in MILA−S fosters student creativity in learning about scientific discovery processes.

## Introduction

Scientific discovery in general is a creative task (Carruthers, Stitch & Siegal 2002; Clement 2008; Darden 1998; Magini, Nersessian & Thagard 1999; Nersessian 2008). Thus, computational modeling of scientific discovery processes has received significant attention in AI research on creativity (Chen et al. 2009; Davies, Nersessian & Goel 2005; Griffith, Nersessian & Goel 2000; Langley 2000; Langley et al. 1987; Lindsay et al. 1980). Science education nowadays emphasizes authentic science practices mimicking the creative discovery processes of real scientists (Clement 2008; Edelson et al. 1999). Thus, interactive tools for supporting authentic science practices in science education have received significant attention in AI research on education (Bridewell et al. 2006; De Jong & van Joolingen 1998; Jackson, Krajcik, & Soloway 2000; Novak 2010; vanLehn 2013).

The goal of supporting creative discovery processes in science education raises several issues for research on computational creativity. We briefly three questions:
*(1) What specific tasks in creative discovery processes should we automate in supporting science education?* We focus on ideation in scientific discovery, and summarize five epistemic views of ideation in the literature. We find that most epistemic views provide few guarantees of the correctness of ideas. This indicates a need for supporting hypothesis evaluation in student learning about creative discovery processes.
*(2) What computational tools may support evaluation of hypotheses in science education?* We focus on conceptual modeling in scientific discovery. We summarize an interactive technology called MILA−S for first elaborating explanatory hypotheses into conceptual models and then evaluating a hypothesis through simulation.
*(3) What is the impact of creativity support tools such as MILA−S on student learning about scientific discovery processes?* We summarize an educational intervention in a middle school engaging MILA−S for modeling ecological phenomena. We find that the use of MILA−S had substantial impact on the discovery processes of middle school students in modeling the ecological phenomenon.

## Epistemic Views of Scientific Discovery

Idea generation is a core element of the creative process in scientific discovery (Clement 2008; Nersessian 2008). However, the task of ideation is complex. The question for us is what specific subtasks of ideation should we automate in supporting student learning about scientific discovery processes? To answer this question, we examine several epistemic views of ideation in scientific discovery.

## Conceptual Classification

One common view of ideation in scientific discovery is classification of data into known categories.. We know about Linneas' classic work on classification in biology. Classification continues to be important in modern biology (e.g., Golub et al. 1999). Classification has been extensively studied in AI (e.g., Duda, Hart & Stork 2001) and ML (e.g., Bishop 2007). The classic DENDRAL system (Lindsay et al. 1980) classified mass spectroscopy data into chemical molecules. Chandrasekaran & Goel (1988) trace the evolution of early AI theories of classification. We have studied both top-down hierarchical classification in which a concept is incrementally refined based on data (Goel, Soundarajan & Chandrasekaran 1987), and bottom-up hierarchical classification in which features of data are incrementally abstracted into a concept (Bylander, Goel & Johnson 1991).

## Abductive Explanation

Abductive inference, i.e., inference to the best explanation for a set of data, is another common view of ideation in scientific discovery. AI has studied abduction from multiple perspectives (e.g., Charniak & McDermott 1985; Josephson & Josephson 1996). The classic BACON system (Langley et al. 1987) abduced physical laws from data. Bylander et al. (1991) have analyzed the computational complexity of the abduction task. Goel et al. (1995) describe a computational technique for abductive explanation based on the RED system for identifying red-cell antibodies in a patient's serum (Fischer et al 1991): the technique assembles composite explanations that explain a set of data from elementary explanations that explain subsets of the data.

## Conceptual Modeling

Conceptual modeling is ubiquitous in science (e.g., Clement 2008; Darden 1998; Nersessian 2008). Conceptual models are abstract representations of the elements, relationships, and processes of a complex phenomenon or system. AI has extensively studied conceptual models (e.g., Davis 1990; Lenat 1995; Stefik 1995). We have developed conceptual models of complex systems that specify how a system works, i.e., the way the system's structure produces its behaviors that achieve its functions (Goel, Rugaber & Vattam 2009). We have used structure-behavior-function modeling for both engineering systems (Goel & Bhatta 2004) and natural systems (Goel et al. 2012) for supporting a variety of reasoning processes in design and invention.

## Analogical Reasoning

Scientific discovery often engages analogical reasoning (Clement 2008; Dunbar 1997; Nersessian 2008). We know about Neil Bohr's famous analogy between the atomic structure and the solar system. Analogical reasoning engages retrieval of an analogue useful for addressing the scientific problem of interest and transfer of the relevant relational knowledge from the retrieved analogue to the scientific problem. AI research has developed several theories of analogical reasoning (e.g., Bhatta & Goel 2004; Falkanehainer, Forbus & Gentner 1989; Hofstader 1996; Thagard et al. 1990). We have studied analogical reasoning in scientific problem solving (Griffith, Nersessian & Goel 2000). Starting from verbal protocols of physicists addressing problems with spring systems (Clement 1988), we developed an AI system called Torque that emulates the problem solving behavior of the physicists.

## Visual Reasoning

Scientific discovery often engages visual representations and reasoning (Clement 2008; Magnini, Nersessian & Nersessian 1999; Nersessian 2008). Although some AI research has explored visual representations and reasoning (e.g., Glasgow, Narayanan & Chandrasekaran 1995), AI research on visual representations and reasoning is not as robust or mature as on, say, classification. We have developed a language for representing visual knowledge and a computational technique for reasoning about visual analogies (Davies, Goel & Yaner 2008), and to understand the use of visual analogy Maxwell's construction of the unified theory of electromagnetism (Davies, Nersessian & Goel 2005).

## The Evaluation Task

It is noteworthy that in general the above methods of idea generation in scientific discovery provide few guarantees of correctness of their results. Further, while these methods help generate hypotheses for a given situation, in general they do not by themselves evaluate their results. This indicates a need for supporting hypothesis evaluation in student learning about creative discovery processes. That is, there is a need for developing interactive tools that automate the evaluation task in the context of supporting creativity in student learning about scientific discovery processes. Thus, we decided to focus on automating the evaluation task in supporting student learning as described below.

# Model Construction and Evaluation

In this work, we elected to automate the evaluation task in the context of supporting creativity in student learning about conceptual modeling. Cognitive science theories of scientific discovery describe scientific modeling as an iterative process entailing four related but distinct phases: model construction, use, evaluation, and revision (Clement 2008; Nersessian 2008; Schwarz et al. 2009). Thus, a model is first constructed to explain some observations of a phenomenon. The model is then used to make predictions about other aspects of the phenomenon. The model's predictions next are evaluated against actual observations of the system. Finally, the model is revised based on the evaluations to correct the errors and improve the model's explanatory and predictive efficacy.

Scientific models can be of several different types, with each model type having its own unique affordances and constraints, and fulfilling specific functional roles in scientific inquiry (Carruthers, Stitch & Siegal 2002; Magnini, Nersessian & Thagard 1999). In this work, we are specifically interested in two kinds of models: conceptual models and simulation models. Conceptual models allow scientists to specify and share explanations of how a system works, aided by the semantics and structures of the specific conceptual modeling framework. Conceptual models tend to rely heavily on directly modifiable representations, languages and visualizations, enabling rapid iterations of the model construction cycle.

Simulation models capture relationships between the variables of a system such that as the values of input variables are specified, the simulation model predicts the temporal evolution of the values of other system variables. Thus, the simulation model of a system can be run repeatedly with different values for the input variables, the predicted values of the system variables can be compared with the actual observations of the system, and the

simulation model can be revised to account for discrepancies between the predictions and the observations. A main limitation of simulation models is the complexity of the setting up a simulation, which makes it difficult to rapidly iterate on the model construction cycle.

AI research on science education has used both conceptual models (e.g., Novak 2010; vanLehn 2013) and simulation models (e.g., Bridewell et al. 2006; de Jong & van Joolingen 1998; Jackson, Krajcik, & Soloway 2000) very extensively and quite productively. However, AI research on science education typically uses the two kinds of models independently from each other: students use one set of tools for constructing, using, and revising conceptual models, and another tool set for constructing and using simulation models. However, cognitive science theories of scientific inquiry suggest a symbiotic relationship between conceptual modeling and simulation modeling (e.g., Clement 2008; Magnini, Nersessian & Thagard 1999; Nersessian 2008): scientists use conceptual models to set up the simulation models, and they run simulation models to test and revise the conceptual models. Thus, we developed an interactive system called MILA–S that enables science students to construct conceptual models of ecosystems, to directly and automatically generate simulation models from the conceptual models, and then execute the simulations.

## MILA−S: A Tool for Model Construction and Evaluation

MILA (**M**odeling & **I**nquiry **L**earning **A**pplication)_ is a family of interactive tools for supporting student learning about scientific discovery. The core MILA tool enables middle school students to investigate and construct models of complex ecological phenomena. MILA–S also allows students to simulate their conceptual models (Joyner, Goel & Papin 2014). In this paper, we focus on the impact of using MILA-S on students' creativity in conceptual modeling.

MILA builds on a line of exploratory learning environments including the Aquarium Construction Toolkit (ACT; Vattam et al. 2011) and the Ecological Modeling Toolkit (EMT; Joyner et al. 2011). ACT and EMT were shown to facilitate significant improvement in students' deep, expert-like understanding of complex ecological systems. For conceptual modeling, ACT used Structure-Behavior-Functions models that were initially developed in AI research on system design (Goel, Rugaber & Vattam 2009). In contrast, EMT used Component-Mechanism-Phenomenon (or CMP) conceptual models that are variants of Structure-Behavior-Function models adapted for modeling ecological systems. Both ACT and EMT used NetLogo simulations as the simulation models (Wilensky & Reisman 2006; Wilensky & Resnick 1999). Like most interactive tools for supporting modeling in science education (vanLehn 2013), both ACT and EMT provided one set of tools for constructing and revising conceptual models and another tool set for using simulations.

Like EMT, MILA–S uses Component-Mechanism-Phenomenon (or CMP) conceptual models that are variants of the Structure-Behavior-Function models used in ACT.. In CMP models, mechanisms explain phenomena such as fish dying in a lake. Mechanisms arise out of interactions among components and relations among them. Components are parts of the physical structure of system, and are classified as either biotic or abiotic; oxygen, for example, is an abiotic component while fish are biotic components. The representation of each component in CMP includes a set of variables such as population, age, birth rate, and energy for biotic components, and amount for abiotic components. The representation of each component is annotated by a set of parameters specifically for setting up a simulation, such as the appearance of the component and ranges for each variable associated with the component.

In the CMP model of a system, representations of components (and their variables) are related together through different kinds of relations. MILA–S provides the modeler with a set of prototype relations. For example, interactions between a biotic component like 'Fish' and an abiotic component like 'Oxygen' could be 'consumes', 'produces', or 'destroys'. Connections have directionality; a connection from 'Oxygen' to 'Fish' would have a different set of prototypes, including 'poisons'. Representations of relations are also annotated with parameters to facilitate the simulation, such as energy provided for 'consumes' and rate of production for 'produces'.

Like ACT and EMT, MILA–S too uses the NetLogo simulation infrastructure. After constructing a CMP conceptual model, a student clicks a 'Run Sim' button to initialize MILA–S and pass their model for simulation generation. MILA–S iterates through some initial boilerplate settings, then gathers together all the components for initialization along with their individual parameters. After this, MILA–S writes the functions based on the relations specified in the CMP model. A key part of this is a set of assumptions that MILA–S makes about the nature of ecological systems. For example, MILA–S assumes that if a biotic component consumes a certain other component, then it must need that other component to survive. A model with 'Fish' that contains 'consumes' connections to both 'Plankton' and 'Oxygen' would infer that fish need both Plankton and Oxygen to survive. MILA–S also assumes that species will continue to reproduce to fulfill their carrying capacity rather

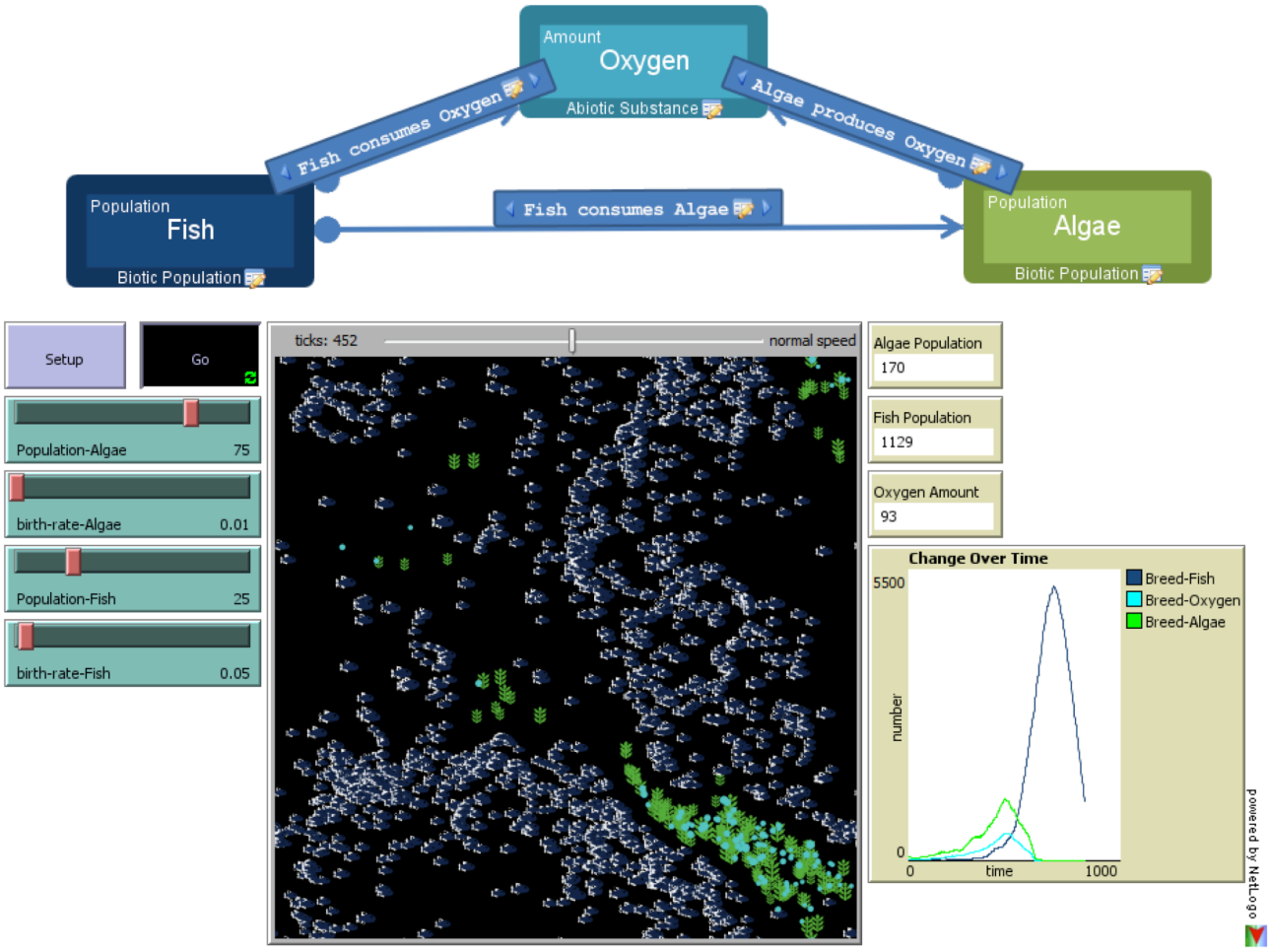



Figure 1: A model in MILA–S (top) showing a set of simple relationships between fish, algae, and oxygen, and the NetLogo simulation (bottom) generated by MILA–S to simulate the model. This model was constructed by the team described in the third case study below; the simulation was generated and run from their model by research staff to obtain this screenshot.

than hitting other arbitrary limitations. These assumptions do limit the range of simulations that MILA–S can generate, but they also facilitate the higher-level rapid model revision process that is the learning objective of this project. Figure 1 illustrates a simple conceptual model constructed by a middle school student team (on the top of the figure) and the results of simulating it (at the bottom).

## Educational Intervention

The present intervention had two main parts. In the first part, 10 classes with 237 students in a metro Atlanta middle school used MILA for two weeks. During this time, students worked in small teams of two or three to investigate two phenomena: a recent massive and sudden fish death in a nearby lake and the record high temperatures in the local area over the previous decade. In the second part, two classes with 50 of the original 237 students used MILA-S to more deeply investigating the phenomenon of massive, sudden death of fish in the lake.

Prior to engagement with MILA–S, the 50 students in our study received a two-week curriculum on modeling and inquiry, featuring five days of interaction with CMP conceptual modeling in MILA. In the first part of the study using MILA, students also used pre-programmed NetLogo simulations that did not respond to students' models, but nonetheless provided students experience with the NetLogo interface and toolkit. Thus, when given MILA–S, students already had significant experience with CMP conceptual modeling, NetLogo simulations, and the interface of MILA−S. The question now becomes what was the impact of using MILA−S on students' creativity?

## Impact on Students' Creativity

An initial examination of the processes and results of model construction by the student teams in our study provides two insights. Firstly, there exists a fundamental difference in the conceptual models that students constructed with MILA–S compared to the earlier models they constructed with MILA: while earlier models were

retrospective and explanatory, models constructed with MILA–S models were prospective and dynamic. Secondly, the model construction process when students were equipped with MILA–S was profoundly different from their earlier process using MILA: whereas previously, conceptual models were used to guide investigation into sources of information such as existing theories or data observations, once equipped with MILA–S the students used the conceptual models to spawn simulations that directly tested the implications of their hypotheses and models thereof.

## The Constructed Models

During engagement with MILA, students produced models that can be described as retrospective and explanatory. Students started from an observable phenomenon, the aforementioned fish kill, and recounted a series of events that led to that result. Causal relationships were captured throughout the model, but only those that lay directly in the causal path leading to the observed phenomenon, and only in the specific way in which the chain occurred in the phenomenon. For example, one team modeled multiple feedback cycles to explain the phenomenon. In their model, a heat spike caused algae populations to grow out of control, then die off due to a lack of carbon dioxide to breathe and a lack of sunlight to produce energy (due to the thick algae clouding the lake). This led to a spike in algae-decomposing bacteria who suddenly had an ample food supply, as well as a drop in the population of oxygen-producing algae. These bacteria, then, consumed an enormous quantity of oxygen, causing the fish population to suffocate. This led to more dead matter in the lake, thus encouraging more bacteria reproduction, exacerbating the fish kill further.

This model presented a complete explanation for why and how the fish kill occurred in the lake; however, the model only captured a retrospective view of the series of events applicable in this situation. Although students could use mental simulation to imagine the results, these models do not explicitly capture dynamic relationships in the system, and thus are of limited use describing what would have happened under different circumstances. For example, had the temperature changed more slowly and allowed the algae to grow steadily rather than exploding and plummeting in quick succession, could the lake have sustained the increased algae population? Would the increased algae population have produced sufficient oxygen to allow the fish population to grow and thrive as well? Thus, models constructed with MILA were explanatory and retrospective.

With MILA–S, students constructed fundamentally different kinds of models that aimed not to capture the series of events that occurred, but rather to capture the dynamic relationships that gave rise to that series of events. Thus, the models constructed in MILA–S invoked a layer of abstraction and generalization away from the models constructed in MILA. For example, one team constructed an initial model that captured the three relationships they considered most pertinent in the system. These students already believed that the fish kill was caused by a sudden drop in oxygen, thus suffocating the fish. Thus, their first relationship was that fish consume oxygen. They similarly knew that oxygen is produced from sunlight, and thus included the relationship between sunlight and oxygen. These connections differed fundamentally from those modelled in MILA, such as accounting for trends in multiple directions (i.e. oxygen production varies directly, up or down, with sunlight presence). The model was not constructed to directly explain the phenomenon, but rather to provide the relationships necessary so that under the right conditions, the phenomenon may arise on its own.

## Model Construction Process

During prior engagement with MILA, model construction occurred as students constructed their initial hypotheses, typically connecting only a cause to a phenomenon with no mechanism in between. This was then used to guide investigation into other sources of information such as observed data or other theories to look for corroborating observations or similar phenomena. The conceptual model was then evaluated according to how well it matched the findings; in some cases, the findings directly contradicted the model, while in other cases, the findings lent evidence or mechanism to the model. Finally, the conceptual models were revised in light of this new information (or dismissed in favor of stronger hypotheses, reflecting revision at a higher level) and the process began again.

During engagement with MILA–S, however, we observed a profound variation on the model construction process. The four phases of model construction were still present, but the nature of model use and evaluation changed. Students started by constructing a small number of relationships they believe to be relevant in the system, the model construction phase. After some initial debugging and testing to become familiar with the way in which conceptual models and simulations fit together, students generated simulations and used them to test the implications of their conceptual models. After running the simulation a few times, students then evaluated how well the results of the simulation matched the observations from the phenomenon. This evaluation had two levels: first, did the simulation accurately predict the ultimate phenomenon (in this case, the fish kill)? Once this basic evaluation was met, an advanced evaluation followed: did other variables, trends, and relationships in the simulation match other observations from the phenomenon? For example, one team successfully caused a fish kill by causing the quantity of food available to the fish to drop, but evaluated this as a poor model nonetheless because nothing in the system indicated a disturbance to the fish's food supply. Finally, equipped with the results of this evaluation, students revised their models to more closely approximate the actual system.

Thus, students still constructed and revised conceptual models, but through the simulation generation framework of MILA–S, the model use and evaluation stages took on the practical rigor, repeatable testing, and numeric analysis

facilitated by simulations. Rather than speculating on the extent to which their model could explain a phenomenon, students were able to directly test its predictive power. Then, when models were shown to lack the ability to explain the full spectrum of the phenomenon, students were able to quickly return and revise their conceptual models and iterate through the process again.

## Three Illustrative Case Studies

We present three case studies from our experiment to illustrate the above observations about the model construction process. These case studies were chosen to demonstrate variations in the process and connections to the underlying model of construction and revision.

### Case 1

The first team posited that pollution from dangerous chemicals played a significant role in the system. Specifically, this team speculated that chemicals were responsible for killing the algae in the lake, which then caused the fish population to drop. They began this hypothesis by constructing a model suggesting that algae produces oxygen, fish consume oxygen, and harmful chemicals destroy algae populations. They then used MILA–S to generate and use a simulation of this model to mimic the initial conditions present in the system (i.e. a fish population, an algae population, and an influx of chemicals). This simulation showed the growth of fish population continuing despite the dampened growth of algae population from the harmful chemicals. The team evaluated this to mean that the death of algae alone could not cause the massive fish kill to occur. The team then revised their model to suggest chemicals directly contributed to the fish kill by poisoning the fish directly, as well as killing the algae.

The team then used MILA–S to generate another simulation. This time, when the team used the simulation under similar initial conditions, the fish population initially grew wildly, but the chemicals ate away at both the fish and algae. Eventually, the harmful chemicals finished eating away at the algae, the oxygen quantity plummeted, and the fish suffocated. Students evaluated that this simulation matched the observed phenomenon, but also evaluated that their model missed a relevant relation: based on a source present in the classroom, students posited that fish ought to consume algae. They revised their model to account for this error uncovered during evaluation, used their simulation again, found the same result, and evaluated that they had provided a model that could explain the fish kill.

### Case 2

A second team started off by creating a simple set of relations that they believed was present due to their biology background and prior experience with MILA. First, they speculated that sunlight "produces" oxygen, and then that fish, in turn, consume the oxygen. Following these two initial relationships, they generated their first simulation through MILA–S and used it to mimic the believed initial conditions of the lake (i.e. a population of fish, available oxygen, available sunlight). Sunlight was inferred to be continuously available, and thus, at first, the population of fish expanded continuously without any limiting factor. However, when the population of fish hit a certain threshold, it began to consume oxygen faster than it was being produced. This led to the quantity of oxygen dropping, and subsequently, the population of fish dropping. However, rather than depleting completely, the fish and oxygen populations instead began to fluctuate inversely, with oxygen concentration rebounding sufficiently when fish population dropped, allowing the fish to rebound.

The team ran this simulation multiple times to ensure that this trend repeated itself. In one instance, the fish population crashed on its own simply due to the suddenness of the fish population growth and subsequent crash. However, the team evaluated that this was not an adequate explanation of what had actually happened in the lake. The team posited that if this kind of expansion and crash could happen without outside forces, it would be more common. Second, the team observed that their model contained faulty or questionable claims, such as the notion that sunlight "produces" algae. This evaluation based on both the simulation results and reflection on the model led to a phase of revision. An 'Algae' component was added between sunlight and oxygen, representing photosynthesis. Students then used MILA–S to generate a new simulation, and used this new simulation to test the model. This time, students found that their model posited that an oxygen crash would *always* occur in the system, and evaluated that while this successfully mimicked the phenomenon of interest, it failed to match the lake on other days.

### Case 3

The third team began with an interesting hypothesis: algae serves as both the food for fish and the oxygen producer for fish. The team, thus, started with a simple three-component model with fish, algae, and oxygen: fish consume algae, fish consume oxygen, and algae produces oxygen. The team further posited that in order for algae populations to grow, they must have sunlight to feed their photosynthesis process. Sunlight, therefore, was drawn to produce algae. The team reasoned that if the fish population destroys the source of one type of 'food' (oxygen) in search for another type (actual food), it could inadvertently destroy its only source for a necessary nutrient.

The team used MILA–S to generate a simulation based on this model and ran it several times under different initial conditions. Each time, algae population initially grew due to the influx of sunlight. As a result, fish populations grew, due to the abundance of both algae (as produced via sunlight) and oxygen (as produced by the algae). As the fish population spiked, the algae hit a critical point where it began to be eaten faster than it reproduced, and the rate of sunlight entering the system was insufficient to maintain steady, strong growth. This caused the algae population to

plummet, and in turn, the fish population to plummet as the fish suddenly lacked both food and oxygen. Sometimes, the algae population subsequently bounced back even after the fish fully died off, while in others both species died entirely.

Unlike the second team, this third team evaluated this to mean their model was accurate: under the initial conditions observed in the lake, their model predicted an algal bloom every single time. Thus, the third team provided two interesting variations on the model construction process observed in other teams: first, they overloaded one particular component, demonstrating an advanced notion of how components can play multiple functional roles. Second, they posited that a successful model would predict that the same events would transpire under the same initial conditions every time, as opposed to the second team's notion that this phenomenon ought to only occur sometimes.

## Summary, Conclusions, and Future Work

Scientific discovery in general is a creative task. Our goal in this work was to enable science students to mimic the scientific modeling practices of real scientists and thus help make learning about scientific discovery as authentic as possible. Our analysis of several epistemic views of idea generation in scientific discovery indicated a need for automating the task of hypothesis evaluation. Therefore, we developed an interactive system called MILA–S that enables science students to construct conceptual models of ecosystems, to directly evaluating the conceptual models by automatically generating simulation models from the conceptual models and then execute the simulations. Our hypothesis was that the computational support for model construction and evaluation embodied in MILA–S would foster student creativity in scientific modeling.

Initial results from a pilot study with 50 students in a middle school provide preliminary evidence in favor of the hypothesis (although a controlled study is needed to conclusively verify these claims). Firstly, students approached the modeling process from a different perspective from the outset, striving to capture dynamic relationships among the components of the ecological system. These dynamic relationships then promoted a more abstract and general perspective on the system. Secondly, the process of model construction, use, evaluation, and revision presented itself naturally during this intervention, with the simulations playing a key role in supporting the cyclical process of constructing conceptual models. By using the simulations to test their predictions and claims, and by subsequently evaluating the success of their conceptual models by matching observations from the actual phenomenon, students engaged in a rapid feedback cycle that saw rapid model revision and repeated use for continued evaluation. MILA–S empowers science students to evaluate the conceptual models through simulation, allowing them to focus on idea generation, and model construction and revision.

Note that in addition to conceptual modeling, this project entails some of the other processes of scientific discovery we briefly mentioned in the introduction. Thus, it engages abductive explanation as students explore multiple hypotheses for explaining an ecological phenomenon, and construct the best explanation for the given data about the phenomenon. It also engages visual representations and reasoning: students construct a visual representation of their conceptual model of the ecological phenomenon (top of Figure 1) and generate visualizations of simulations directly from the conceptual models (bottom of Figure 1).

We are presently engaged in a full-scale investigation to test these theories, techniques and tools with college-level biology students. The objective of this investigation is to examine the use of creativity support tools for scientific modeling of ecological phenomena in college-level introductory biology courses.

## Acknowledgements

We thank the anonymous reviewers of this paper: their comments helped improve the articulation of this work.

## References

Bishop, C. (2007) *Pattern recognition and Machine Learning.* Springer.

Bridewell, W., Sanchez, J., Langley, P., & Billman, D. (2006) An Interactive Environment for Modeling and Discovery of Scientific Knowledge. *IJHMS* 64(11): 1099-1114.

Bylander, T., Allemang, D., Tanner, M., & Josephson, J. (1991) The Computational Complexity of Abduction. *Artificial Intelligence* 49: 25-60.

Bylander, T., Johnson, T., & Goel, A. (1991) Structured Matching: A Task-Specific Technique for Making Decisions. *Knowledge Acquisition* 3(1):1-20.

Carruthers, P., Stitch, S., & Siegal, M. (editors, 2002) *The Cognitive Basis of Science*, Cambridge University Press.

Chandrasekaran, B., & Goel, A. (1988) From Numbers to Symbols to Knowledge Structures: Artificial Intelligence Perspectives on the Classification Task. *IEEE Transactions on Systems, Man, and Cybernetics* 18(3): 415-424.

Charniak, E., & McDermott, D. (1985) *Artificial Intelligence.* Addison-Wesley.

Chen, C., Chen, Y,, Horowitz, M., Hou, H., Liu, Z., & Pellegrino, D. (2009) Towards an Explanatory and Computational Theory of Scientific Discovery, Infometrics 3(3): 191-209.

Clement, J. (1988) Observed Methods of Generating Analogies in Scientific Problem Solving. *Cognitive Science* 12: 563-586.

Clement, J. (2008). *Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Simulation*. Dordrecht: Springer.

Darden, L. (1998). Anomaly-driven theory redesign: computational philosophy of science experiments. In T. W. Bynum, & J. Moor (Eds.), *The Digital Phoenix: How Computers are Changing Philosophy* (pp. 62–78). Oxford: Blackwell.

Davies, J., Goel, A., & Yaner, P. (2008) Proteus: A Theory of Visual Analogies in Problem Solving. *Knowledge-Based Systems* 21(7): 636-654.

Davies, J., Nersessian, N., & Goel, A. (2005) Visual Models in Analogical Problem Solving. *Foundations of Science* 10(1): 133-152.

Davis, E. (1990) *Representations of Commonsense Knowledge.* Morgan Kauffman.

De Jong, T., & Van Joolingen, W. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research, 68*(2), 179-201.

Duda, R., Hart, P., & D. Stork. (2001) *Pattern Classification.* 2nd Edition. John Wiley.

Dunbar, K. (1997) How scientists think: Online creativity and conceptual change in science. In Ward, Smith, & Vaid (Editors), *Conceptual structures and processes: Emergence discovery and change* (pp. 461–493). Washington, DC: American Psychological Association.

Edelson, D., Gordin, D., & Pea, R. (1999). Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the Learning Sciences, 8*(3-4), 391-450.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989) The Structure-Mapping Engine: Algorithms and Examples. *Artificial Intelligence* 41(1): 1-63.

Fischer, O., Goel, A., Svirbely, J., & Smith, J. (1991) The Role of Essential Explanations in Abduction. *Artificial Intelligence in Medicine* 3: 181-191.

Glasgow, J., Narayanan, N.H., Chandrasekaran, B. (Editors, 1995) *Diagrammatic Reasoning: Cognitive and Computational Perspectives*, MIT Press.

Goel, A., & Bhatta, S. (2004) Use of Design Patterns in Analogy-Based Design. *Advanced Engineering Informatics* 18(2):85-94.

Goel, A., Josephson, J., Fischer, O., & Sadayappan, P. (1995) Practical Abduction: Characterization, Decomposition and Distribution. *Journal of Experimental and Theoretical Artificial Intelligence* 7: 429-450.

Goel, A., Rugaber, S., & Vattam, S. (2009). Structure, Behavior & Function of Complex Systems: The SBF Modeling Language. *AI for Engineering Design, Analysis and Manufacturing,* 23: 23-35.

Goel, A., Soundararajan, N., & Chandrasekaran, B. (1987) Complexity in Classificatory Reasoning. In *Proc. Sixth National Conference on Artificial Intelligence (AAAI-87),* Seattle, July 1987, 421-425.

Goel, A., Vattam, S., Wiltgen, B., & Helms, M. (2012) Cognitive, Collaborative, Conceptual and Creative - Four Characteristics of the Next Generation of Knowledge-Based CAD Systems: A Study in Biologically Inspired Design. *Computer-Aided Design*, 44(10): 879-900.

Golub, T., Slonim, D., Tamayo, P. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 289(5439): 531-537.

Griffith, T., Nersessian, N., & Goel, A. (2000) Function-follows-Form: Generative Modeling in Scientific Reasoning. In *Proc. Twenty Second Conference of the Cognitive Science Society*, Philadelphia, pp. 196-201.

Hofstadter, D (Editor, 1996) *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*, Basic Books, New York.

Jackson, S., Krajcik, J., & Soloway, E. (2000) Model-It: A Design Retrospective. In M. Jacobson & R. Kozma (editors), *Innovations in Science and Mathematics Education: Advanced Designs for Technologies of Learning* (pp. 77-115). Lawrence Erlbaum.

Josephson, J., & Josephson, S. (editors, 1996) *Abductive Inference: Computation, Philosophy, Technology.* Cambridge University Press.

Joyner, D., Goel, A., Rugaber, S., Hmelo-Silver, C., & Jordan, R. (2011). Evolution of an Integrated Technology for Supporting Learning about Complex Systems. In *Proc. 11th IEEE International Conference on Advanced Learning Technologies*, Athens, GA.

Joyner, D., Goel, A., & Papin, N. (2014). MILA–S: Generation of Agent-Based Simulations from Conceptual Models of Complex Systems. In *Procs. 19th International Conference on Intelligent User Interfaces*, Haifa, Israel.

Langley, P. (2000) The Computational Support of Scientific Discovery. *IJHMS* 53(3): 393-410.

Langley, P., Simon, H., Bradshaw, G., & Zytkow, J. (Editors, 1987) *Scientific Discovery: Computational Explorations of the Creative Process*. Cambridge, MA: MIT Press, 1987.

Lenat, D. (1995) CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of ACM* 58(11): 32-38.

Lindsay, R., Buchanan, B., Feigenbaum, E., & Lederberg, J. (Editors, 1980) *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project.* McGraw-Hill.

Magnini, L., Nersessian, N., & Thagard, P. (editors, 1999) *Model-Based Reasoning in Scientific Discovery*. Kluwer.

Nersessian, N. (2008). *Creating Scientific Concepts*. Cambridge, MA: MIT Press.

Novak, J. (2010) *Learning, Creating and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. New York: Routledge.

Schwarz, C., Reiser, B., Davis, E., Kenyon, L., Achér, A., Fortus, D., Shwartz, Y., Hug, B., & Krajcik, J. (2009). Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners. *Journal of Research in Science Teaching, 46*(6), 632-654.

Stefik, M. (1995) *Knowledge Systems.* Morgan Kauffman.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence*, 46, 259-310.

VanLehn, K. (2013). Model construction as a learning activity: a design space and review. *Interactive Learning Environments*, *21*(4), 371-413.

Vattam, S., Goel, A., Rugaber, S., Hmelo–Silver, C., Jordan, R., Gray, S, & Sinha, S. (2011) Understanding Complex Natural Systems by Articulating Structure- Behavior-Function Models. *Journal of Educational Technology & Society, 14*(1): 66-81.

Wilensky, U., & Reisman, K. (2006). Thinking Like a Wolf, a Sheep, or a Firefly: Learning Biology Through Constructing and Testing Computational Theories-An Embodied Modeling Approach. *Cognition and Instruction, 24*(2), 171-209.

Wilensky, U., & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology*, 8, 3-19.

# Intentionally Generating Choices in Interactive Narratives

**Michael Mateas** and **Peter Mawhorter** and **Noah Wardrip-Fruin**
Computer Science Department
University of California Santa Cruz
Santa Cruz, CA 95064 USA
{michaelm, pmawhorter, nwf}@soe.ucsc.edu

### Abstract

Interactive stories face a famous "authorial bottleneck." Two existing approaches to this problem are story management systems, such as drama managers, and interactive narrative generators. Existing work leverages well-understood qualities of linear narrative such as suspense to generate content, but interactivity brings new capacities, like the ability to make a player experience regret. These interactive poetics arise from the player's ability to make choices, and depend heavily on the structure of the choices that are presented to the player. This system description paper presents a system that creates choices by reasoning about their structure, and describes the architecture that enables it to do so.

## Introduction

Since the 1970's, researchers in artificial intelligence have been making systems that can creatively generate stories (Klein et al. 1971). With the rise of digital games, and in particular, interactive narratives[1], this research has found a new application: generating and managing the complexities of interactive narratives. One approach to this problem is to manage players' experiences. A managed experience lets authors create a diverse array of content while letting players experience a coherent narrative that includes different parts of the content depending on their choices. Another approach is to create systems that generate content, letting authors work at a more abstract level (perhaps writing re-combinable actions or events) which the system can then use to generate a wide variety of possible stories. Both approaches are proposed solutions to the fact that the work necessary to create a truly open world is overwhelming for human authors (Orland 2011). Existing systems have demonstrated the viability of reasoning about traditional narrative qualities for both experience management and story generation. Qualities unique to interactive narratives have not yet been widely used for reasoning in such systems, however. For example, the ability to make a player regret their own actions is unique to interactive contexts, and it depends on aspects of the narrative (such as which actions the player intended, and which outcomes were consequences of player actions) that go beyond traditional narrative qualities.

Interactive narrative systems thus stand to gain by reasoning about interactive as well as traditional poetics. Presented here is a system called *Dunyazad* that attempts just that: It dynamically builds choices with the goal of achieving specific poetic effects. *Dunyazad* focuses on choice poetics as a subset of interactive poetics, attempting to structure the choices that it gives the player so that they evoke feelings like safety or confusion (Mawhorter et al. 2014). As an operationalization of choice poetics, *Dunyazad*'s successes and failures can also inform the theory that drives it.

Liapis, Yannakakis, and Togelius recently stated that games were an ideal domain for computational creativity, and listed interactive narrative as an important part of that domain (Liapis, Yannakakis, and Togelius 2014). Human authors are now exploring the full potential of interactive narrative: Many independent games have earned praise for their stories, and communities that produce innovative interactive narratives have formed around tools like *Inform 7* (`http://inform7.com/`) and *Twine* (`http://twinery.org/`).[2] If generative narrative systems want to leverage the potential of interactive narrative, they will need to reason about interactive poetics, and in particular, how the choices they present to players are perceived.

## Prior Work

In computational narrative systems, there has been a recent trend towards explicit poetics. Szilas' 2003 *IDTension* first proposed the idea of creating an interactive narrative by "simulating the laws of narrative" (Szilas 2003), much as one can produce a wide range of gameplay by simulating the laws of physics. This direction of work naturally proceeds by identifying the mechanism of specific poetic effects and building computational systems to produce those effects. El-Nasr's 2007 *Mirage* (El-Nasr 2007) is another example of this approach; it attempts to apply a range of dramatic techniques to increase engagement in an interactive narrative. In contrast, systems such as *Suspenser* (Cheong and Young 2006) and *Prevoyant* (Bae and Young 2008) have focused on specific poetic effects (suspense and surprise respectively).

*Dunyazad* as described here can be viewed as continuing this line of research because it reasons explicitly about poet-

[1] The authors are aware that interactive narratives predate digital games in several forms, but digital games have popularized interactive narrative as a medium.

[2] These are both examples of tools not explicitly designed to encourage creativity which nonetheless support it by making authoring faster and easier.

ics; it emphasizes interactive poetics, and in particular, the poetics of discrete choices. Whereas *IDTension* and *Mirage* incorporate traditional poetics into interactive experiences, *Dunyazad* focuses on interactive poetics, leveraging choice structures to create affect. In some respects *Dunyazad* is designed as much to illuminate interactive poetics as to exploit them: because it uses declarative code to construct poetic choices, its successes and failures can be traced to concrete parts of its theory, and that theory can thus be informed by the system's performance.

More recently, several studies have attempted to formally investigate and model poetic effects in interactive narrative contexts, with a focus on choices. In 2011, Thue, Bulitko, Spetch, and Romanuik measured players' perceptions of agency and found that they often differed from what one might expect based on the choices available to the player (Thue et al. 2011). Their system did manipulate an interactive narrative to achieve a poetic effect (give the player a sense of agency), but it focused on manipulating events in a way that was invisible to the player, rather than on changing a player's perceived options at any particular choice. In a study of agency which did not involve a generative system, Fendt, Harrison, Ware, Cardona-Rivera, and Roberts were able to create an illusion of agency, albeit in the context of an extremely simple interactive narrative (Fendt et al. 2012). A follow-up to the Fendt et al. study by Cardona-Rivera, Robertson, Ware, Harrison, Roberts, and Young linked players' perceptions of differences between outcomes to their perceptions of agency (Cardona-Rivera et al. 2014). In another paper focusing on choices in interactive narratives, Yu and Riedl were able to predict player choices using collaborative filtering (Yu and Riedl 2013).

This active research surrounding choices in interactive narratives shows that authors are interested in the poetic effects of choices. However, systems that actually reason about the poetics of the choices they generate are scarce– most existing systems reason about different options and outcomes independently. Barber and Kudenko's 2007 work on dilemma-based interactive narrative is a notable exception (Barber and Kudenko 2007). Their work focuses on a single type of choice, generating interactive experiences where each choice is a dilemma.

Ideally, a system that took choice poetics into account would dynamically construct each choice that it offers the player for maximum poetic impact. Of course, just as *IDTension* and *Mirage* reason about a range of classical poetics, such a system could take into account a range of interactive poetics (including aspects beyond choice poetics). But even a system that only considers choice poetics is a step in the right direction.

## Choice Poetics

The theory of choice poetics described by Mawhorter, Mateas, Wardrip-Fruin, and Jhala in (Mawhorter et al. 2014) provides a framework for reasoning about choices, which is crucial for an interactive narrative system which must generate them. When analyzing the poetics of a choice, the first consideration is the player's mode of engagement: how is the player approaching the game, and what do they hope to achieve through their play? Common modes of engagement include power play (playing to achieve ludic goals like scoring points), avatar play (playing by projecting yourself into the game and making the choices you would make in a character's situation) and role-play (playing to express a particular role through the actions of one or more characters you control). There are also other less common modes of engagement like critical play, and players can (and usually do) engage with multiple modes at once. Taking modes of engagement into account does not require reading the player's mind, however: just as with any other element of a game, designers can make decisions based on their intuitions about how players will play, and they can refine their designs through playtesting.

*Dunyazad* directly encourages avatar play, and assumes that this will be players' primary mode of engagement when it constructs choices. Role play is also supported to some extent, but because there are only minimal game mechanics, *Dunyazad*'s stories do not lend themselves to power play. The game mechanics that do exist (skills which affect outcomes) are deployed in such a way that favorable outcomes from an avatar play perspective (which are favorable for the diegetic protagonist) are aligned with favorable gameplay outcomes (those in which the action attempted is successful, generally leading to successful endings).

Once a choice is considered in terms of a particular mode of engagement, it may fall into one of several classes of recognizable choice idioms, such as the dilemma or the false choice. Recognizing these idioms is based on an analysis of the framing, options, and outcomes of a choice (for example, a classic dilemma must have exactly two options, and the options' outcomes should each thwart a different player goal). Besides classifying choices as examples of choice idioms, (Mawhorter et al. 2014) does not say much about how to construct choices, although it does list some aspects of player experience that can be manipulated through the use of different choice structures. More analysis of existing interactive fictions within the framework of choice poetics would likely yield more specific methods for choice construction, however, and there is some existing advice on choice construction in the form of authoring advice for human authors of interactive narrators (e.g., (Choice of Games LLC 2010)). The choice construction methods in *Dunyazad* are currently based on this latter body of work, as described in the Choice Generation section on page 5.

## *Dunyazad*

Although not yet complete, *Dunyazad* is a novel story generation system that is intended to generate interactive narratives in the style of *Choose Your Own Adventure* books using second-person narration and explicit choices. *Dunyazad* treats choices as first-class objects, and reasons about their structures. In particular, it has rules for constructing a variety of choice types based on the player's estimated expectations and evaluations in its choice structure module.

*Dunyazad* ultimately produces natural language narratives. Each story consists of sections of text followed by choices, where each choice leads to another section of text or to an ending. *Dunyazad* is not interactive, but instead generates

entire interactive narratives that players can interact with separately (importantly, this allows players to re-play parts of the narrative).

*As you journey onwards, a leviathan rises majestically up from the ocean, tentacles curling. It is threatening you.*

→ *You try to flee from it.*
– *You attempt to pacify it with music.*

*You flee from it and escape. You travel onwards.*

Figure 1: A minimal example vignette

Because *Dunyazad* is focused on operationalizing choice poetics, its default domain is simple travel/adventure stories in a fantasy setting, made up of sequences of relatively independent "vignettes" or scenes. Each vignette is made up of a setup plus a few basic actions, some of which may be player-initiated choices ("choice" nodes) and some of which may be events dictated by the system ("event" nodes). Each story node thus represents a single event or choice, including a context, one or more actions that might happen, and any outcomes of those actions. Although most world state is reset at the beginning of each vignette, the state of the player's party is not, allowing for some overall continuity. Figure 1 shows the full text of a minimal example vignette composed of a single choice (at which the player chose to flee), and a single event (the default vignette-ending event "travel onwards"). This vignette also introduces some new context at its choice node (an attacking monster) which is described in the text. Of course, when presented to the player, the text stops at the choice until the player has selected an option.

Although more complex vignettes are possible, the system is designed to create a stream of simple, direct choices, imitating the game *Spent* (`http://playspent.org/html/`) (McKinney 2011). By limiting the complexity of vignettes and refreshing most of the world state between vignettes, the user experience is directed towards shallow and playful interaction, and at the same time, the system has fewer opportunities to accidentally create plot holes. This also creates an environment in which the poetics of individual choices (e.g., was the last choice relaxing?) are important to the feel of the story overall, as opposed to merely supporting a dramatic arc defined by traditional narrative elements.[3] The system accordingly assumes that players will mainly engage in avatar play, and perhaps also light role play, and attempts to provide choices that enable these modes of engagement.[4]

From a technology standpoint, *Dunyazad* combines imperative Python code with declarative answer set programs to iteratively grow a branching story. The imperative code manages the iteration, at each step filling in single story node and adding new blank child nodes for each new option created. Filling in a node is accomplished by using the Potassco Labs tools `gringo` and `clingo` to ground and solve an answer set program (Gebser et al. 2011). The answer set program for each node includes predicates that represent the entire current story state, but facts from the solution to the program are only used to modify the currently-focused node.

After a complete story structure is created, *Dunyazad*'s imperative code uses a set of text templates to render the story into natural language. This module takes care of verb conjugation and pronominalization where necessary, and the text templates form a generative grammar which adds extra variation to the story. This variation doesn't change the underlying sequence of events, and mostly consists of word choice and sentence-structure variation that reduces literal repetition when similar events are described multiple times.

While *Dunyazad*'s hybrid iterative/declarative approach does limit the kinds of constraints that the system can easily place on multi-node story structures, it is necessary to keep the answer set problems tractable: Asking the solver to produce a complete story with hundreds of nodes in a single step is not a task that many modern computers could handle (if any), whereas just solving a single node can be accomplished in seconds. At the same time, being able to use answer set programming for the creation of individual nodes provides two benefits. First, answer set programming reasons simultaneously about all of its constraints, which means that building some logic which detects a certain condition also allows direct control over that condition (by e.g., prohibiting it or requiring that it hold). This means that there is little distinction between writing code which *recognizes* a phenomenon and writing code which *produces* it: the answer set solver does the hard work of figuring out what has to happen in order for the phenomenon to occur.

The second main benefit of using answer set solving is that it *directly* encodes constraints. *Dunyazad* as a project aims to apply choice poetics to the generation of interactive narrative, but it should also be able to push back on choice poetics when constructing choices based on theory fails to produce the expected results. Setting aside the difficult issue of blame assignment between the system and the theory, using answer set programming enables the system to better inform the theory because the constraints responsible for producing behavior can be directly translated into theoretical statements. For example, a rule like `regret(Choice) :- consequence(Choice, Outcome), bad_for_player(Outcome)` translates directly to a theoretical statement "When the player chooses an outcome that leads to something which is bad for them, they will feel regret." If testing reveals that players do not feel regret when the system thinks they should, the rule can be refined, and because it is a direct encoding of the theory, such refinement can directly inform the theory.

## Representation

Although the output of *Dunyazad* is natural language, it has an underlying predicate representation of the stories it generates. Each story node describes either a choice or an event, and the structure of the two is the same, the only difference being events have only one option. Story nodes have a rich predicate representation of their initial state, which can encode arbitrary properties of and relations between story elements, including characters and items. Story nodes also

[3] In this case, the overarching plot of a journey to an exotic destination constrains little in terms of tension, narrative developments, etc.

[4] For more details about of modes of engagement, refer to (Mawhorter et al. 2014)

1. `st(root, inst(actor, monster_76)).`
2. `st(root, property(name, inst(actor, monster_76), "leviathan")).`
3. `st(root, relation(threatening, inst(actor, monster_76), inst(actor, you))).`
4. `at(root, action(option(1), flee)).`
5. `at(root, outcome(option(1), o(success, escape))).`
6. `at(root, outcome(option(1), o(get_injured, safe))).`
7. `at(root, arg(option(1), fearful, inst(actor, you))).`
8. `at(root, arg(option(1), from, inst(actor, monster_76))).`
9. `at(root_1, action(option(1), travel_onwards)).`

Figure 2: Some example predicates describing parts of fig. 1

have some number of options, each of which has an action associated with it, along with argument bindings for that action. Figure 2 shows some of the predicates that describe the example vignette in fig. 1.

Story states are sets of state predicates each of which takes one of four forms:

1. `st(root, inst(Type, ID)).`
   Declares the existence of a particular instance, which has a `Type` of either `actor` or `item`.
2. `st(root, state(State, Inst)).`
   Assigns a unary state such as `injured` to an instance.
3. `st(root, property(Prop, Inst, Value)).`
   Associates a property with an instance and specifies its value. For example, an actor can have the `has_skill` property with a value of `music` indicating that they possess the `music` skill. Properties can be multi-valued.
4. `st(root, relation(Rel, From, To)).`
   Asserts a relation between two instances. For example, an actor can have the `has_item` relation with an item. Some constraints (like exclusivity of the `has_item` relation) are enforced.

Frame axioms dictate that state changes only occur when specified by actions. Actions are defined by arguments, outcome variables, skill links, preconditions, and post-conditions as follows:

1. `argument(Action, Arg, Type).`
   Specifies an argument `Arg` which must bind an instance of type `Type` in the current state.
2. `outcome_val(Action, Var, Val).`
   Specifies that outcome variable `Var` can take on value `Val`. Each variable has multiple possible values.
3. `skill_link(Skill, Type, NeedsTool, Action, Arg, o(OutVar, OutVal)).`
   Skill links specify how character skills influence action outcomes. The four link types are `required`, `promotes`, `avoids`, and `contest`. These indicate player expectations. For example, the `healing` skill is linked to the `healed` value of the `success` outcome variable for the `treat_injury` action via a `required` link that also specifies that a tool is needed. Thus if the player lacks the `healing` skill and an option for them to take the `treat_injury` action is presented, the system assumes that the player will expect the action to fail.
4. Pre- and post-conditions. These have no fixed form, but instead are arbitrary logical constraints. For example, it is an error for the `treat_injury` action to be performed on a patient who is not injured. Most depend on outcome variables having specific values. Another example: if the `success` variable of a `treat_injury` action has a value of `healed`, then the `injured` state is removed from the patient, but if the `success` variable is either `still_injured` or `killed` this doesn't happen.

As an example, the text "You try to flee from it," in fig. 1 is a rendering of an action "flee" with the player character and the monster as arguments. The flee action has two outcome variables: `success` which has values `escape` and `failure`, and `get_injured`, which has values `injured` and `safe`. In fig. 2, facts 4 to 8 describe the action, outcome, and arguments of this option (the node that it is part of is `root`, and it is the first option at that node). Of course, each option at one node leads to another story node, and any consequences of the outcome associated with that option are reflected in the starting world state of the linked node. In fig. 1, the initial choice story node links to two successor nodes, only one of which is displayed.[5] In this case, the consequence of the successful flee action is that you have escaped the threat of the monster (which was part of the initial world state). The "travel onward" action's initial world state thus does not include the threat of the monster attack, which is actually a precondition for the "travel onwards" action–it requires an absence of "problems."

"Problems" and more generally "potentials" (which are either "problems" or "opportunities") are an important part of how the system builds stories. *Dunyazad* represents a "setup" as a partial world state which is added to the current world state when a new vignette begins. In fig. 1, the "monster attack" setup is used, which introduces a monster (in this case a leviathan) which is threatening the player-character.

[5] From a developer's perspective, all of the linked nodes are part of the same vignette, but of course without re-play, a player will only see one of them.

The fact that the player is being threatened (which is encoded as a relation) is explicitly recognized by the system as a "problem"–in fact all instances of the "threatening" relation are considered "problems," even when the player is not the target. This explicit representation of both problems and opportunities drives the basic consideration of what actions are appropriate in a given situation, and also plays into the rules about choice structures.

## Reasoning

*Dunyazad* uses answer set programming to create the individual events and choices that make up a story. It is thus governed by a set of logical constraints which dictate what event configurations are acceptable. As already mentioned, solving for dozens of story nodes simultaneously is infeasible, because solving time is exponential in the number of nodes considered. As a compromise, *Dunyazad* iteratively solves individual story nodes.

Thus *Dunyazad*'s reasoning revolves around the construction of a single event or choice node. The rules governing node construction can be divided into three categories:

- Constructive rules–rules that help create the basic structure of facts, such as the rule that stipulates that each option has an action associated with it.
- Sense rules–rules that disallow nonsensical story structures, such as the rule that says that no choice should have two identical options or the rule that disallows trading items with oneself.
- Content rules–rules that discard some valid stories as uninteresting or otherwise undesirable, such as the rule that requires successive vignettes to use different setups.

Without constructive rules, core facts like those that assign values to arguments would be missing from result answer sets, and the system would crash. Without sense rules, all of the basic components of a story would be there, and the story could be rendered to natural language by the text generation system, but the result would be at best surreal and at worst gibberish. Without content rules, the result would be an understandable sequence of events, but it would probably not be an interesting story.

Because of the nature of answer set programming, *Dunyazad* effectively chooses an arbitrary permutation of an event among all possibilities that satisfy its rules (consult (Gebser et al. 2011) for more background on how answer set programming works). Each rule represents a constraint on the generative space of story nodes, which makes it both easy to prune the generative space, and easy to see how an individual constraint effects the generative space. The following variables determine the space of possible choice structures:

- The number of options (minimum 2 for a "choice" node; maximum 4 for performance reasons).
- The action for each option (there are 13 actions in the current domain model).
- The possible argument bindings for each action (most actions have 2-4 arguments, and each argument generally has 5-7 type-appropriate bindings at a given story state).
- The values for each outcome variable of each action (most have 1-2 outcome variables with 2 values each).

Unsurprisingly, there are a staggering number of possibilities under the constructive rules, but this space is reduced drastically by the sense and content rules.

## Choice Generation

*Dunyazad*'s design is in part based on concrete human advice for writing choice-based narratives offered by Choice of Games (an interactive narrative publisher) in several online articles (Choice of Games LLC 2010). In particular, *Dunyazad* focuses primarily on expectations and outcomes, which factor prominently in an article about the fundamentals of choice design titled "5 Rules for Writing Interesting Choices in Multiple-Choice Games," (Fabulich 2010).

*Dunyazad*'s choice structure subsystem is devoted to estimating and managing the poetics of the choices it generates. Abstractly, this subsystem reasons about choices in terms of expectations and outcomes, using estimates of player perception. This same structure for representing and reasoning about choices could be used by other systems that wanted to generate choices intentionally.

The most basic structure of *Dunyazad*'s choice representation has already been described: a choice consists of *context*, *options*, and *outcomes*. *Context* in this case is a world state, *options* correspond to discrete, fully-specified actions that the player-character can take, and *outcomes* are the changes in world state that result from a particular action. To actually reason about the poetics of a choice, however, the system needs to make some assumptions about the player's experience, which gives rise to three more entities: *player goals*, *player expectations* and *perceived outcomes*.

*Player goals* are the basis for reasoning about how players might perceive choices. Some basic player goals can be predicted by the author, and to the extent that players actually pursue these goals, an author can design choice poetics. For example, an author might presume that players will want to keep their character alive and healthy, and that players will also want to maintain the health of their allies. A choice where the player is forced to sacrifice either their character's health or the health of an allied character could then be constructed with the goal of adding to the player's sense of tension. If players do in fact value their character's health and that of their allies, the choice should be a tense one (other details of its construction notwithstanding). For players who don't value one of these goals, the choice will lack the tension that the author intended, but that doesn't mean that the author's strategy for creating a tense moment was invalid. The author also has methods for encouraging the pursuit of various goals, such as using standard narrative techniques to try to promote empathy with the characters.

*Dunyazad* relies on the same strategy as this hypothetical author to create choice poetics: through its fixed introduction segment and according to genre conventions, it encourages players to pursue certain goals. It then estimates the poetic effects of the choices it creates assuming that the player will be invested in those goals. *Dunyazad* assumes are that the player will pursue the following goals:

- Avoid injury to themselves and their allies (high priority).
- Avoid threats to themselves and other non-aggressive characters (high priority).
- Have all actions they take be successful (low priority).
- Acquire and retain tools for their skills (low priority).

Given assumed *player goals*, a choice can be considered in terms of its *player expectations* and *perceived outcomes*. Both of these will vary from player to player, but just like with *player goals*, human authors can often estimate them. Like the *player goal* estimation, *player expectation* and *perceived outcome* estimation depends on the system author. This works via the skill link system as mentioned in the Representation section above: the author of an action specifies which skills are linked to which outcomes and how, and this information is used by the system to estimate player expectations. For example, if the player has a goal to maintain their health, but they're missing the `fighting` skill, an option allowing the player to `attack` an enemy will be marked as dangerous, because the `fighting` skill is linked to the `injured` value of the `aggressor_state` outcome variable, and that outcome would cause the player to be injured, threatening their goal.

As an example, consider the choice in fig. 1. This choice has two options, which correspond to the "flee" and "pacify" actions. Unbeknownst to the player (before they've made a decision at least), the outcome of the "flee" action in this case will be a successful escape, while the outcome of the "pacify" action (not shown) will be a failure that does not change the world state (i.e., the monster continues to threaten the player). While generating this choice, the system creates a *player expectation* for each option for each player goal, indicating how the player would expect that option to impact that goal. Taking "escape from threats" as a player goal, both options at this choice are expected to threaten that goal, because the system knows that both options could fail to achieve it. At the same time, both options are expected to enable that goal, because depending on their outcomes, either option could achieve that goal.

But is either option *likely* to succeed or fail? Assuming that the player has the "wilderness lore" skill (linked by a "contest" link to the "flee" action) but the monster does as well, the first option is indeterminate. However, based on a "required" skill link, if the player does not have the "music" skill, the second option is likely to fail.

There are thus five possible non-exclusive *player expectations* per *player goal*:

- Irrelevant–this option is irrelevant to this goal.
- Threatens–this option risks failing this goal.
- Enables–this option might achieve this goal.
- Fails–this option is expected to fail this goal.
- Achieves–this option is expected to achieve this goal.

Threatens and enables expectations are assigned based on all possible outcomes of an action, while fails and achieves are based on outcomes that the player has reason to believe are likely. Combinations of these expectations can describe a variety of situations. For example, a choice which threatens, enables, and fails a goal could be seen as a desperate gamble: it has a possibility of success, but it is expected to fail.

Similarly, there are five *perceived consequences* for each *player goal*:

- Irrelevant–this outcome does not affect this goal.
- Hinders–this outcome hinders progress towards achieving this goal, but does not actually cause it to fail.
- Advances–this outcome contributes to achieving this goal, but does not actually achieve it.
- Fails–this outcome directly fails this goal.
- Achieves–this outcome directly achieves this goal.

Unlike *player expectations*, *perceived consequences* are mutually exclusive, and while expectations only reason about the potential outcomes of an action, *perceived consequences* are assigned based on actual outcomes.

Returning to our example, the *player expectations* for the first option with regards to the goal "escape from threats" will include "threatens" and "enables," but since both the player and the monster have the associated contest skill, no stronger expectation is formed. For the second option, because the player lacks the relevant "music" skill[6], the *player expectations* will be "threatens," "enables," and "fails." The *perceived outcome* of the first option will be that it achieves the "escape from threats" goal, while the *perceived outcome* of the second option will be that it fails this goal.

This representation of player goals, expectations, and perceived outcomes enables rich reasoning about the poetics of a choice. To start with, it's easy to encode simple choice idioms (see (Mawhorter et al. 2014)). An example would be a dilemma–traditionally a choice with exactly two options which lead to two different negative consequences. A choice with exactly two options, each of which is expected to fail one of two goals and enable the other fits these criterion. The perceived outcomes here determine what kind of dilemma it is, for example a false dilemma might have identical outcomes for both options.

*You come to a tavern and decide to rest for a while. A merchant is selling a music book and she is selling an oboe and a noble is bored and a peasant is bored and an innkeeper seems knowledgeable.*

- *You play a song for the peasant. (+music)*
- *You gossip with him. (+elocution)*
- *You offer to trade the merchant some perfume for the music book. (no skill)*
- *You tell the noble a story. (+storytelling)*

Figure 3: A "relaxed" choice.

To give a more concrete demonstration of choice generation, consider figs. 3 and 4. These examples were generated

[6]Relevant to an actual player's expectations is whether they are *aware* of this lack. For now, *Dunyazad* ensures this by mentioning any relevant skills (or lack thereof) in parentheses after each option. These are omitted in fig. 1 to avoid confusion.

*As you travel onwards, a dragon slowly approaches you. It is threatening you.*

- *You attack it. (-combat)*
- *You attempt to pacify it with music. (-music)*

Figure 4: A "grim" choice.

using a single player goal based on the idea of power-gaming: "Succeed at every action." Evaluating expectations relative to that goal, fig. 3 is the result of asking for a "relaxed" choice: a choice where there are no options which the player expects to fail. Figure 4[7] is the opposite: a "grim" choice where every option is required to be expected to fail.

When generating individual choices, the system uses everything available to it (the choice of setups, the background including the player's starting skills, and the configuration of options and outcomes) to create choices that satisfy the given constrains, which can be expressed directly at the level of player expectations. This ability to reason about player expectations is critical in a system that wants to use choice structures to achieve poetic effects. Of course, the system isn't reasoning directly about the player's actual expectations, but merely about the system author's guess as to what those expectations will be. For human authors, this is often enough to achieve their goals, and for systems without dynamic player modelling, it will have to be enough as well.

## Abstract Architecture

The underlying principles behind *Dunyazad*'s choice structure rules suggest requirements for systems that want to build choices intentionally. At a basic level, the ability to reason about all parts of a choice: the context, options, and outcomes, is required. It should be noted here that reasoning about the options individually is not sufficient: a system that wants to dynamically construct choices that create poetic effects needs to be able to reason about the range of options available at a choice and assert things like "There are no options available which the player expects will lead to positive outcomes." That brings up the next requirement: such a system needs to be able to reason about player goals, expectations, and perceptions. These things do not have to be modelled exactly as *Dunyazad* models them, but they should be represented in such a way that the system can reason about them.

In order to creatively construct a choice that gives the player a feeling of "agency" or "regret" or "power" a system needs to be able to define those things in terms of the player's view of the game. It is of course possible to give a player these feelings in an interactive narrative without representing them in any sort of system, but that just means that a human author has done the reasoning required, not that it never happened. And if a human author did that reasoning, then the system will not be able to freely generate such choices: it will be limited to generating them in situations that the human author was able to foresee.

[7] These examples were slightly edited from their original form for brevity.

So what is the next step once you have a system that reasons about all parts of a choice and the player's perspective besides? The next step is to identify the choice poetics that you want your system to create, and define them in terms of the player's perspective. For example, if you want the player to experience "agency," you might define your objective as "The system should create choices with multiple options that the player expects will lead to significantly different world states," as in (Cardona-Rivera et al. 2014). Given this concrete definition of your goal in terms of player expectations, the system should be able to construct such choices. If your goal is hard to pin down in terms of player expectations, playing some interactive narratives that create the feeling you want to create and analysing their choice structures would be the place to start.

While a computer-generated interactive narrative that successfully evoked a particular feeling using choice structures would be an achievement in its own right, even better would be a system that used multiple choice structures in service of a more complex goal. For example, if there are narrative generation mechanisms trying to achieve a desired tension level, making sure that choice structures are also contributing to this goal would be a benefit. Or if the player-character is supposed to be stumped by a mystery at some point in the plot, perhaps generating choice structures where there are no clear good or bad options could reinforce that point. The ability to craft choices towards poetic ends unlocks many new options for an interactive narrative system.

## Future Work

*Dunyazad* is still under development, and getting it to generate full interactive narratives as opposed to individual choices is our current focus. Once it does so, it will be critical to evaluate the narratives it generates, both from a creative standpoint and to determine whether players actually perceive the effects that it is trying to create. If *Dunyazad* is able to create full interactive narratives with choices that support their stories, it will represent an important step towards narrative generators that take full advantage of interactive as well as traditional poetics. However, even if it only generates individual choices, *Dunyazad* still enables experiments that can contribute to knowledge of choice poetics.

Generating full stories will require significant authoring effort. *Dunyazad*'s current domain model has 13 actions, 6 potentials, 6 setups, and 4 player goals. In order to generate experiences even few minutes long, it would need to generate stories with dozens of nodes across perhaps 8-12 vignettes. The primary authoring effort to get to that point with a satisfying level of variety lies in creating more distinct setups, as well as adding a few more actions. Although actions, potentials, setups, and player goals are all modular from a technical standpoint and can be authored individually, practically they need to take each other into account in order to create interesting output. This is mostly a consequence of the content rules. For example, adding an action which results in a new state probably won't change the system's output by itself, because without any player goals or potentials that involve that state, the system will never consider the new action to be relevant. Actions, potentials, setups, and

goals thus form interconnected subsystems that are linked by certain key states. Although this makes authoring a bit tricky, these subsystems are at least somewhat independent of each other: the actions that involve trading items can be authored without worrying about injury and death.

Besides further work on *Dunyazad*, there are several promising research directions suggested by this work. First, the fact that an interactive narrative system is making assumptions about its players begs for a mechanism by which the system could actually measure its players. (Thue et al. 2011) is an example of exactly that, but by increasing both the complexity of choice manipulation and the detail of the player model things would become even more interesting. There is a link here to the world of intelligent tutoring system: systems like Graesser, Chipman, Haynes, and Olney's *AutoTutor* already have components that attempt to measure a student's knowledge and even emotions (Graesser et al. 2005). Adapting these to work in an interactive narrative context as opposed to a tutoring context would give the system a much better means of estimating a reader's expectations and goals than authorial guesswork.

Expanding a choice-poetics based system to deal with broader interactive poetics would be interesting too. Many games offer interactions much more complicated than discrete choices, and reasoning about these would be more difficult. Many of the same principles apply, however, and creating a system that analyzed complex interactive situations in terms of player expectations, available actions, and their consequences would allow the deliberate construction of more complicated and open-ended narratives.

Finally, a system capable of deliberate choice creation might enable new and more dynamic forms of narrative. This is an eventual aim of *Dunyazad*: to create a branching story with myriad paths where the freedom to explore a huge possibility space is enabled by collaboration between a human author and a computer system. If generative tools could enable human authors to design entire narrative possibility spaces, the resulting fictions would be the products of both human and machine creativity.

## Acknowledgements

The authors would like to acknowledge the support of NSF grant IIS-1409992.

## References

Bae, B.-C., and Young, R. M. 2008. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In *Interactive Storytelling*. Springer. 156–167.

Barber, H., and Kudenko, D. 2007. Dynamic generation of dilemma-based interactive narratives. In *3rd Artificial Intelligence and Interactive Digital Entertainment Conference*.

Cardona-Rivera, R. E.; Robertson, J.; Ware, S. G.; Harrison, B.; Roberts, D. L.; and Young, R. M. 2014. Foreseeing meaningful choices. In *10th Artificial Intelligence and Interactive Digital Entertainment Conference*.

Cheong, Y.-G., and Young, R. M. 2006. A computational model of narrative generation for suspense. In *AAAI*, 1906–1907.

Choice of Games LLC. 2010. "Game Design" posts. `http://www.choiceofgames.com/category/blog/game-design/`. Accessed 2015-2-27.

El-Nasr, M. S. 2007. Interaction, narrative, and drama: Creating an adaptive interactive narrative using performance arts theories. *Interaction Studies* 8(2):209–240.

Fabulich, D. 2010. 5 rules for writing interesting choices in multiple choice games. `http://www.choiceofgames.com/2010/03/5-rules-for-writing-interesting-choices-in-multiple-choice-games/`. Accessed 2015-2-27.

Fendt, M. W.; Harrison, B.; Ware, S. G.; Cardona-Rivera, R. E.; and Roberts, D. L. 2012. Achieving the illusion of agency. In *Interactive Storytelling*. Springer. 114–125.

Gebser, M.; Kaufmann, B.; Kaminski, R.; Ostrowski, M.; Schaub, T.; and Schneider, M. 2011. Potassco: The potsdam answer set solving collection. *AI Comm.* 24(2):107–124.

Graesser, A. C.; Chipman, P.; Haynes, B. C.; and Olney, A. 2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on* 48(4):612–618.

Klein, S.; Oakley, J. D.; Suurballe, D. J.; and Ziesemer, R. A. 1971. A program for generating reports on the status and history of stochastically modifiable semantic models of arbitrary universes. Technical Report TR142, Computer Sciences Department, The University of Wisconsin–Madison.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *5th International Conference on Computational Creativity*, 285–292.

Mawhorter, P.; Mateas, M.; Wardrip-Fruin, N.; and Jhala, A. 2014. Towards a theory of choice poetics. In *International Conference on the Foundations of Digital Games*, FDG '14.

McKinney. 2011. Spent. Web, `http://playspent.org/html/`. Accessed 2015-2-9.

Orland, K. 2011. 'Old Republic' writer discusses '60 man-years' of work. `http://kyleorland.com/blog/2011/12/21/old-republic-writer-discusses-60-man-years-of-work/`. Accessed 2015-2-28; full article archived at `http://web.archive.org/web/20120206234922/http://ingame.msnbc.msn.com/_news/2011/12/20/9569126-old-republic-writer-discusses-60-man-years-of-work`.

Szilas, N. 2003. Idtension: A narrative engine for interactive drama. In *Technologies for Interactive Digital Storytelling and Entertainment Conference*, volume 3, 187–203.

Thue, D.; Bulitko, V.; Spetch, M.; and Romanuik, T. 2011. A computational model of perceived agency in video games. In *7th Artificial Intelligence and Interactive Digital Entertainment Conference*.

Yu, H., and Riedl, M. O. 2013. Data-driven personalized drama management. In *9th Artificial Intelligence and Interactive Digital Entertainment Conference*.

# "In reality there are as many religions as there are papers" – First Steps Towards the Generation of Internet Memes

**Diogo Costa, Hugo Gonçalo Oliveira, Alexandre Miguel Pinto**
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
dcosta@student.dei.uc.pt, hroliv@dei.uc.pt, ampinto@dei.uc.pt

**Abstract**

We report on the first steps towards the automatic generation of Internet memes starring public figures. Their images are retrieved from the Web and combined with famous quotes, altered according to recent information on the figures. Current implementation, in Portuguese, exploits several computational resources and aims to produce artifacts with coherent text, image, and some humor value. A preliminary evaluation survey confirmed a strong relation between generated memes and present events. Results on humor were also positive.

## Introduction

The term meme originally denotes *an idea, behavior, or style that spreads from person to person within a culture* (Dawkins, 1976; Blackmore, 2000). On the Internet domain, memes became a popular and effective way of transmitting an idea. They are a product of human creativity that typically take the form of an image, often combined with a short phrase. They tend to be funny, make people laugh, and aim to be spread throughout the World Wide Web by sharing and re-sharing in social media.

We present the first steps towards the development of MemeGera, a system for the automatic generation of Internet memes – or better, protomemes[1] – starring public figures (hereafter, characters). MemeGera uses famous quotes, altered as follows: one word is replaced by another that is semantically related to the character and its current information[2]. These sentences, presented together with a character's image, should convey a simple and effective idea, make sense for the character, even if only for a short period of time after generation, and exhibit some novelty. To deal with the latter, the system exploits fresh information on the character, such as that in recent news or tweets. The produced text+image combinations have thus a transient flavor which, together with their humor potential, may qualify them as "jokes *du jour*". Long-term knowledge on the character, from its Wikipedia page, is also explored, but so far only used to favor fresh information.

[1]The definition of meme implies social sharing, which will only occur if people actually spread the protomeme.

[2]As the title of the paper, a twist of Mahatma Gandhi's quote: "In reality there are as many religions as there are individuals".

We see the generation of meme sentences as a kind of linguistic creativity, a topic that covers tasks such as the generation of: poetry (Toivanen, Gross, and Toivonen, 2014; Gonçalo Oliveira and Cardoso, 2015); metaphors (Veale and Hao, 2008); neologisms (Smith, Hintze, and Ventura, 2014); or verbally-expressed humor (Binsted and Ritchie, 1994; Valitutti et al., 2013). Given the funny aspect inherent to memes, our work is probably closer to the latter. Yet, although not essential, the character and its image also play an important role in the success of our memes.

In the remaining of this paper, we provide some background knowledge on the study of humor, together with computational approaches to this topic. We then present the automatic method for generating memes and list each of the steps involved. The current implementation targeted Portuguese, our native language, and is described right after. Although the method may seem quite straightforward, our effort involves the combination of several knowledge sources. Before concluding, we describe an illustrative example and report on the results of an online survey, which suggests that we are heading in the right direction. All the memes used in the survey are shown in the end of the paper, together with information about their generation and evaluation.

## Background and Related Work

This section addresses the topic of humor from a theoretical point of view, followed by an enumeration of computational approaches for humor generation and recognition.

### Theoretical Study of Humor

Humor has been studied from a variety of perspectives ranging from psychology and philosophy (Morreall (2013)), to its sociological aspects in literature (Kuipers (2010)) and, more recently, via the computational approach (e.g. Suslov (1992); Ritchie (2014)). Theoretical accounts of humor encompass the superiority theory, endorsed by Descartes, where "*our laughter expresses feelings of superiority over other people or over a former state of ourselves*"; the relief theory, a hydraulic model proposed by Shaftesbury, and later refined by Sigmund Freud, according to which laughter acts as a mechanism for releasing accumulated nervous energy built up from many possible emotionally-charged situations; and the incongruity theory, proposed by Beattie and sponsored by Kant, Schopenhauer, and Kierkegaard, among oth-

ers, which claims "*laughter is the perception of something incongruous – something that violates our mental patterns and expectations*", which is now the dominant theory.

Socioliterary studies (e.g. Kuipers (2010)) explore the mechanisms through which humor is related to social boundaries, and how it differs between groups; whereas computational approaches address the building of formal theories of humor (Ritchie (2014)), the synthesis of a sense of humour via specific algorithms (Suslov (1992)), and the generation of humorous text and jokes (Ritchie (2009)).

Humor expressed in Portuguese has also been studied from a theoretical point of view. While presenting linguistic mechanisms for achieving humor in this language, Tagnin (2005) states that, since humor breaks conventionality in language, understanding it is a sign of fluency.

### Humor generation

The automatic generation of humor has been a research topic for more than two decades. In early work by Binsted and Ritchie (1994), a model, implemented under the name of JAPE, was proposed for generating punning riddles. The generated puns (e.g. *What do you call a murderer that has fiber? A cereal killer*) took advantage of spelling or word sense ambiguities. STANDUP (Manurung et al., 2008) follows the lines of JAPE, but is more robust, user friendly, and was developed with the purpose of allowing young children, especially those with linguistic disabilities, to explore language and improve their skills.

Given a concept and an attribute, HAHAcronym (Stock and Strapparava, 2005) rewrites existing acronyms and generates new ones with a humor intent. It relies on an incongruity detector and generator that selects opposing domains and opposing adjectives, while considering also rhythm and rhymes. For instance, the acronym FBI may become *Fantastic Bureau of Intimidation*. Or given the concept of 'processor' and the attribute 'fast', it generates the acronym OPEN – *Online Processor for Effervescent Net*.

Valitutti et al. (2013) explored the generation of adult humor based on the replacement of a word in a short message. The word should introduce incongruity and lead to a humorous interpretation, achieved by applying three constraints. It must: (i) be of the same form as the original word, i.e. 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 something related to sex; (iii) take place at the end of the message and keep the coherence of the original sentence. An example of an output is: *I've sent you my fart.. I mean 'part' not 'fart'....*

Besides English, there were attempts for generating puns in Japanese (e.g. Sjöbergh and Araki (2007a)). We are not aware of any work of this kind for Portuguese.

### Humor recognition

In the scope of natural understanding, there has been work on the automatic recognition of verbally-expressed humor. Researchers typically focus on a specific kind of jokes, such as *knock-knock* (Taylor and Mazlack, 2004) and *That's what she said* (Kiddon and Brun, 2011), or on a less specific kind of humor but transmitted in bounded kinds of text, such as single sentences (Mihalcea and Strapparava, 2006; Sjöbergh and Araki, 2007b), or tweets (Barbieri and Saggion, 2014). Humor recognition is generally seen as a text classification problem and relies on a set of humor relevant features to train a classifier, given their presence in humorous and non-humorous text. For instance, Barbieri and Saggion (2014) exploit hashtags, such as *#humuor* or *#irony*, to collect positive examples. Selected features generally include the occurrence of antonymous or ambiguous words, alliteration, and other words or expressions typically used in jokes, such as slang or idiomatic expressions.

For Portuguese, the closest works to humor recognition we are aware of include the automatic detection of irony (Carvalho et al., 2009) or proverbs (Rassi, Baptista, and Vale, 2014) in text.

### Internet Memes

Internet memes are a current trend in social media. They are typically a reusable combination of text and graphics. Popular memes include Boromir from the Lord of the Rings with the template "*One does not simply X*", Morpheus from the Matrix with "*What if I told you Y*", or Batman slapping Robin, with a personalized text in their speech balloons. There is however a subtype of Internet memes related to current events, where new images, text, or both, can be used – if successful enough, they might be reused. Events that triggered several memes include the football player Luis Suárez biting his opponent in a World Cup 2014 match (e.g. "*If you can't beat them, eat them*"), or when the pop singer Madonna fell on stage, while wearing a cape, during a performance in the BritAwards 2015 ceremony (e.g. "*56 years old, still does her own stunts*", "*Has a cape, can't fly*"). While most memes show a break of conventionality (e.g. unexpected situation, confusing interpretation, taboo meaning), we address the previous subtype, which, as suggested by the superiority theory, makes fun of the portrayed character. In fact, the image is sometimes enough to make people laugh (e.g. when it displays a funny person or situation).

We are not aware of any published work on the automatic generation of Internet memes. Existing web services for meme generation rely on the user input of both images and text. There is work however on the automatic combination of images and text, such as Grafik Dynamo (2005) and Why Some Dolls Are Bad (2008), by Kate Armstrong[3]. In those projects, a narrative is dynamically generated by combining sequences of images, retrieved from social networks, with speech balloons. The result is often non-sense.

## Method

This section provides a high-level description of our proposed method for meme generation. Specific details of its current implementation are given in the next section.

Among other parameters, our algorithm for the generation of memes (see figure 1) uses the name of a public figure, our character, currently provided by the user. Informally, it starts by retrieving $n$ recent messages (e.g. tweets) mentioning the character, from where the top-$k$ frequent nouns

[3] http://katearmstrong.com/

are collected. Then, it selects a random quote from a pool of famous quotes, pairs it with one of the top-$k$ nouns, and generates a sentence, more precisely, an altered quote where the last noun of the original quote is replaced by one of the top-$k$ nouns – similarly to Valitutti et al. (2013), replacing the last noun will increase surprise and humor potential. After repeating this process for a predefined number of times, generated sentences are ranked by a dedicated scoring function. The highest-ranked sentence is pasted on an image of the character, automatically retrieved from the Web, and the combination is finally returned as the generated meme.

The scoring function considers the humor value of the sentence, the frequency of the replacement noun, and its presence in a more stable long-term information source on the character. Words without previous associations to the character are considered novel and are thus favored in the ranking. We may find some parallelism between this and the work of Toivanen, Gross, and Toivonen (2014), where novel associations in documents are identified by their overlap with known associations from a background corpus.

**Require:**
$charName$:name of character
$n$ : # of messages to retrieve
$k$ : # of top frequent common nouns to consider
$m$ : # of $< quote$,frequent noun$>$ pairs to generate

**procedure** MEMEGERA
  $messages \leftarrow \{msg : msg$ mentions $charName\}$
    $: \#messages = n$
  $freqNouns \leftarrow$ top-$k$ most frequent nouns in $messages$
  $quotes \leftarrow \{quote : quote$ is a famous quote$\}$
  $pairs \leftarrow \{< quote, freqNoun >$ randomly generated$\}$
    $: \#pairs = m$
  $maxEval \leftarrow 0$
  $bestQuote \leftarrow \emptyset$
  **for each** $< quote, freqNoun > \in pairs$ **do**
    $nq \leftarrow$ replace last noun in $quote$ with $freqNoun$
    $ne \leftarrow score(nq, freqNoun, charName)$
    **if** $ne > maxEval$ **then**
      $maxEval \leftarrow ne$
      $bestQuote \leftarrow nq$
  $image \leftarrow$ get image of $charName$ from the Web
  $resultingMeme \leftarrow$ paste $bestQuote$ in $image$
  return $resultingMeme$

Figure 1: Meme generation algorithm

## Implementation

Although our method is language-independent, its current implementation targets Portuguese. MemeGera was implemented in Java and exploits several available resources, for different purposes, including a classifier for Portuguese humor, currently in development. We also describe the function that currently ranks the generated sentences.

### Tools and Resources

Famous quotes used in this work were acquired from the Portuguese edition of Wikiquote[4], a collaborative repository of quotes, run by the Wikimedia Foundation. For the current version of the system, we selected quotes from three well-known thinkers – Mahatma Gandhi, Aristotle and Confucius – who were the authors of many quotes, most of them timeless and generic enough for our purpose. We soon realized that long quotes would not produce the desired effect, so we only used quotes with up to 15 words, totaling 90.

We use the social network Twitter[5] and Twitter4J[6], a Java API, to retrieve tweets mentioning the names of the selected characters. While we could have used a news site or aggregator, the choice of Twitter relied on the fact that its messages are shorter, up-to-date, and mix different and less controlled opinions. In recent years, Twitter has been widely exploited by computer programs, not only for text mining, but also in computational creativity research (e.g. Veale (2014); Cook, Colton, and Gow (2014) or the recent PROSECCO Code Camp[7], focused on the development of creative Twitterbots).

Natural language processing is made by the OpenNLP toolkit[8] and its models trained for Portuguese tokenization and part-of-speech tagging. Since the models were not trained with tweets, a few annotation errors are expected. But this is not severe because we end up using only words in a morphological lexicon, LABEL-Lex[9], in which we rely to perform inflection, so that words agree with the sentence they are put in. Also, we count the lemmas frequency in the tweets, and not the words frequency. Lemmatization is performed by LemPort (Rodrigues, Gonçalo Oliveira, and Gomes, 2014), a Portuguese lemmatizer.

Nouns long-associated to famous people were collected from the abstracts of their articles in the Portuguese Wikipedia, retrieved directly from the DBPedia[10] entries under the category of *Person*.

Images of the meme characters are retrieved automatically from Google Images[11], at runtime. The first hit for each character is always used.

The Mallet[12] toolkit was used in the development of a humor classifier for Portuguese, presented in the next section. Given a positive and a negative dataset, Mallet automatically converts input text to features, and learns a classifier, using one of the algorithms available out-of-the-box.

### Humor Classifier

We have recently started to work on a classifier for recognizing humorous pieces of text, in Portuguese, currently trained with the Mallet toolkit. The first step for its development was the collection of examples of humorous and non-humorous Portuguese documents, labeled respectively as positive or negative. The selected datasets were then imported to Mallet, which was used to train a classifier with the

[4] http://pt.wikiquote.org/

[5] https://twitter.com/

[6] http://twitter4j.org/

[7] http://codecampcc.dei.uc.pt/

[8] https://opennlp.apache.org/

[9] http://label.ist.utl.pt/pt/labellex_pt.php

[10] http://dbpedia.org/

[11] https://images.google.com

[12] http://mallet.cs.umass.edu/

best available learning algorithm. Instead of labeling the examples manually, we collected them from selected sources which we now present.

**Positive Dataset:** While it is rather easy to collect negative examples, the same does not apply for humorous examples in Portuguese. After searching in the Web, we were able to find the following compilations of Portuguese jokes:

- *Bíblia de Anedotas*[13] (in English *Bible of Jokes*);
- *O Sagrado Caderno das Piadas Secas*[14] (in English, *The Sacred Book of Dry Jokes*).

To focus on shorter jokes, we discarded all with more than 25 words, and were left with 790 positive examples.

**Negative Dataset:** The non-humorous dataset should contain text with a similar structure to the positive examples but without a potential humor effect. We thus collected sentences of similar length ($\leq$25 words) from non-humorous sources. Since many of the collected jokes have a question-answer structure, we included this kind of text as well. The following resources were used:

- 304,211 sentences from the Portuguese Wikipedia, each collected randomly from a different article;
- Text from Portuguese corpora available through the AC/DC project (Santos and Bick, 2000)[15]:
  - 81,478 sentences from CETEMPublico, a corpus with editions of the Portuguese newspaper Público (1991-1998).
  - 25,000 sentences from CONDIVport, a corpus of sports newspapers, fashion and health magazines;
  - 6,767 question-answer pairs from *Museu Da Pessoa*, a corpus of interviews.

In the end, we had a total of 417,456 negative examples.

**Validation:** After importing the positive and negative datasets, a classifier was trained with the Maximum Entropy algorithm, selected after a 10-fold cross-validation, where it yielded 99.8% accuracy. These numbers look promising, but they were computed in a dataset with mostly negative examples. Although the $F_1$ for the negative class was 99.9%, it was just 63.7% for the positive, with a recall of 49.4%.

We should stress that the classifier is still in an early stage of development. In the future, instead of relying only in the *black-box* text classification of Mallet, additional features should be integrated, including a subset of those used by others (Sjöbergh and Araki, 2007b; Mihalcea and Strapparava, 2006). Moreover, we are aware that we cannot expect much of the current classifier, at least for the kind of sentences we are generating. While it was trained with classic and timeless jokes, understanding the generated sentences requires not only general world knowledge, but further information that may be valid only on a specific moment in time.

[13] http://rbep.cm-porto.pt/rbep/upload/dnloads/BibliadeAnedotas.doc

[14] https://www.facebook.com/CadernoDasPiadas

[15] http://linguateca.pt/ACDC/


Figure 2: Meme generated for the pop singer Madonna. The text translates to *Keep your thoughts positive, because your thoughts become your falls.*

### Ranking function

As referred earlier, MemeGera generates a set of $m$ sentences that combine a known quote with a noun $f$ retrieved from Twitter. Towards the selection of the most promising generated sentences, these are currently ranked by the following linear combination:

$$\begin{aligned} Score = & \, humorProb * \alpha \\ & + wordFrequency * \beta \\ & + notInWikipedia * \gamma \end{aligned}$$

There, *humorProb* is the probability returned by the humor classifier; *wordFrequency* is the number of tweets where $f$ occurs, divided by the total number of retrieved tweets, $n$; and *notInWikipedia* is a binary function that is 1 if the word is not in the Wikipedia abstract of the character, or 0 otherwise.

## Results

We generated several memes with different configurations. Although not enough experiments were performed to select the best configuration, at a certain point, we started to use fixed parameters, to have a base for comparison. In all reported experiments, generation was based on 200 tweets ($n = 200$), written in Portuguese (according to Twitter), and using the top-5 frequent nouns ($k = 5$). The best sentence was selected from a set of 20 ($m = 20$). The ranking function used the weights: $\alpha = 0.7$, $\beta = 0.25$, $\gamma = 0.05$.

### Example

Figure 2 illustrates the output of MemeGera with a meme generated on the $26^{th}$ February 2015, the day after Madonna fell on stage. The original quote, attributed to Mahatma Gandhi, was *Keep your thoughts positive, because your thoughts become your words*.

Since people were talking about the fall, the most frequent nouns in tweets were: *tombo* (tumble), *queda* (downfall), *palco* (stage), *vídeo* (video) and *madonno*. The last one results from an incorrect part-of-speech tag given to the proper noun Madonna. There is no risk of using it though, because

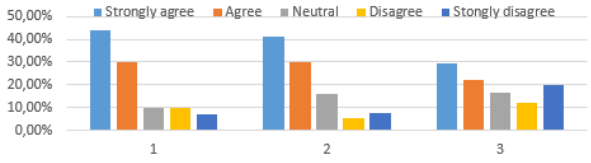



Figure 3: Overall results of the performed evaluation.

it is not in the morphological lexicon. None of the top-5 nouns were in Madonna's Wikipedia abstract.

In the end of this paper, we present more memes, together with information that will help understanding why they were generated, and their scores according to the online survey where they were used.

### Evaluation survey

In order to have a first appreciation of the results produced by MemeGera, we made an online survey, answered by 41 human subjects, all Portuguese native speakers. The survey had the title "Imagens com texto" (Images with text) and it never mentioned the word meme, nor automatic generation.

The survey had 5 memes, for which the name of the character was presented together with three questions, to be answered according to a Likert scale: *strongly agree* (5), *partially agree* (4), *neutral* (3), *partially disagree* (2) and *strongly disagree* (1). The questions were:

1. The text is syntactically and semantically coherent (Does it follow the grammar rules and makes sense?).
2. There is coherence in the combination of text, image and the present time (We suggest to search for breaking news about the character).
3. The combination of the text and the image produce a humorous effect (Did it make you smile?).

The used memes were generated between the $25^{th}$ February 2015 night and $26^{th}$ morning. Their characters were manually selected for being mentioned in fresh news in online media. All of these memes used the highest-ranked sentence from the 20 generated. They are presented in the end of the paper, together with information on their generation, as well as their individual scores in the survey questions.

The survey opened just a few hours after generating the last meme, and was opened for about 24 hours. This means that some memes would only be interpreted appropriately by someone following the daily news. Figure 3 presents overall results, which combine the answers to the five memes.

The survey confirmed that it is often safe to replace one word in a sentence by another of the same part-of-speech. If inflection is handled properly, syntax remains coherent, which makes it easier for semantics, especially when using generic quotes. Answers on the coherence between text, image and the present time are also positive. The meme in figure 6 was the one with more negative answers in the first two questions. First, possibly because it is not very easy to find semantic connections between *tumble* and *awake*. Second, because this meme was related to a very recent event and, although we suggested the subjects to search for the character in the news, most of them probably did not do it, and were not aware of Madonna's fall.

As for the humor aspect, while we cannot say that the generated memes are very funny and have the ability to make everybody laugh, the overall results are encouraging, as the majority of the answers are positive. It is always subjective to assess the presence of humor, especially in this case, where world knowledge and following recent news was a requirement. A curious fact is that the memes with clearly positive answers in this aspect are those with Portuguese politicians. Given that all our subjects were Portuguese, they are probably better informed about Portuguese characters, who probably play a more relevant role on the subjects lives, and make them more responsive to laugh at. This is related to another issue: the image itself or, sometimes, just the character, might play an important role in the humor value, since there are people for which we are more prone to laugh at than others.

## Concluding remarks

We have presented the first steps towards the development of MemeGera, a system that generates combinations of text and image that may be seen as Internet memes. Famous quotes are altered according to a public figure and complemented with their image, automatically retrieved from the Web. Fresh information on the public figure, in the form of frequent words, is currently obtained from Twitter. Several altered quotes are generated and the best is selected after a ranking that considers the humor value and the novelty of retrieved words, in an attempt to positively discriminate the most promising sentences. The humor value is given by an automatic classifier, trained with positive and negative examples of humor expressed in Portuguese. However, this tool is still far from what we expect from it and gave very low scores to the generated sentences (rarely more than 1%).

On the other hand, we should stress that MemeGera has the ability of generating a different and novel sentence each time, based on fresh news. In fact, the results of an online survey showed that it is not only capable of generating coherent sentences, with some relation to the character, but that the generated combinations have some humor potential.

The work described in this paper lead to the development of the *@memegera* Twitterbot that, from time to time: (i) reads the list of current trends in the Portuguese Twitter; (ii) checks if any of them is the name of a known person; if so, (iii) generates a meme on that person and posts it. The bot is still in a test phase, but we may soon start relying on users feedback (e.g. retweets, favored) for evaluation and adaptation of the weights in the ranking function.

Additional plans include both improvements to the system and to its evaluation. In the scope of this and other projects, the humor classifier shall be improved by: (i) enriching the datasets with humorous text from the Twitter accounts of famous Portuguese humorists; (ii) considering additional features (e.g. ambiguity of words and adult slang, for which there are available Portuguese resources we could use). To increase variation, we will devise adding more quotes to our pool, as long as they are not too specific. Regarding evaluation, in a further survey, we aim at recording the reaction of

the subjects in the moment when the meme is first presented to them, and draw conclusions from their facial expressions.

## Acknowledgements

The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the 7th Framework Programme for Research of the European Commission, under FET grant number 611733.

## References

Barbieri, F., and Saggion, H. 2014. Automatic detection of irony and humour in twitter. In *Procs. of 5th Intl. Conf. on Computational Creativity (ICCC)*.

Binsted, K., and Ritchie, G. 1994. An implemented model of punning riddles. In *Procs. of 12th National Conf. on Artificial Intelligence (Vol. 1)*, AAAI '94, 633–638. Menlo Park, CA, USA: AAAI Press.

Blackmore, S. 2000. *The Meme Machine*. Oxford University Press, USA.

Carvalho, P.; Sarmento, L.; Silva, M. J.; and de Oliveira, E. 2009. Clues for detecting irony in user-generated contents: Oh...!! it's "so easy" ;-). In *Procs. of 1st Intl. International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, TSA '09, 53–56. ACM.

Cook, M.; Colton, S.; and Gow, J. 2014. Automating game design in three dimensions. In *Procs. of AISB Symposium on AI and Games*.

Dawkins, R. 1976. *The Selfish Gene*. Oxford University Press, Oxford, UK.

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.

Kiddon, C., and Brun, Y. 2011. That's what she said: Double entendre identification. In *Procs. of 49th Annual Meeting of the Association for Computational Linguistics*, 89–94. Portland, OR, USA: ACL Press.

Kuipers, G. 2010. Humor styles and symbolic boundaries. *Journal of Literary Theory* 3:219–239.

Manurung, R.; Ritchie, G.; Pain, H.; Waller, A.; O'Mara, D.; and Black, R. 2008. The construction of a pun generator for language skills development. *Applied Artificial Intelligence* 22(9):841–869.

Mihalcea, R., and Strapparava, C. 2006. Learning to laugh (automatically): Computational models for humor recognition. *Computational Intelligence* 22(2):126–142.

Morreall, J. 2013. Philosophy of humor. In Zalta, E. N., ed., *The Stanford Encyclopedia of Philosophy*. Spring 2013 edition.

Rassi, A. P.; Baptista, J.; and Vale, O. 2014. Automatic detection of proverbs and their variants. In *Procs. of 3rd Symposium on Languages, Applications and Technologies (SLATE 2014), Bragança, Portugal*, OASICS, 235–249. Schloss Dagstuhl.

Ritchie, G. 2009. Can computers create humor? *AI Magazine* 30(3):71–81.

Ritchie, G. 2014. Logic and reasoning in jokes. *European Journal of Humour Research* 2(1):50–60.

Rodrigues, R.; Gonçalo Oliveira, H.; and Gomes, P. 2014. LemPORT: a high-accuracy cross-platform lemmatizer for portuguese. In *Procs. of 3rd Symposium 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. of 2nd Intl. Conf. on Language Resources and Evaluation*, LREC 2000, 205–210.

Sjöbergh, J., and Araki, K. 2007a. Automatically creating word-play jokes in japanese. In *Procs. of NL-178*, 91–95.

Sjöbergh, J., and Araki, K. 2007b. Recognizing humor without recognizing meaning. In *Procs. of Applications of Fuzzy Sets Theory: 7th Intl. Workshop on Fuzzy Logic and Applications (WILF 2007) Camogli, Italy*, volume 4578 of *LNCS*, 469–476. Springer.

Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A neologism creator nomen ipsum. In *Procs. of 5th Intl. Conf. on Computational Creativity (ICCC)*.

Stock, O., and Strapparava, C. 2005. The act of creating humorous acronyms. *Applied Artificial Intelligence* 19(2):137–151.

Suslov, I. M. 1992. Computer model of a "sense of humour". i. general algorithm. *Biophysics* 37(2):242–248.

Tagnin, S. E. O. 2005. O humor como quebra da convencionalidade. *Revista Brasileira de Lingüística Aplicada* 5(1):247–257.

Taylor, J. M., and Mazlack, L. J. 2004. Computationally recognizing wordplay in jokes. In *Procs. of Cognitive Science Conference (CogSci)*, 2166–2171.

Toivanen, J.; Gross, O.; and Toivonen, H. 2014. The officer is taller than you, who race yourself! In *Procs. of 5th Intl. Conf. on Computational Creativity (ICCC)*.

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. of 51st Annual Meeting of the Association 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. of 22nd Intl. Conf. on Computational Linguistics*, volume 1 of *COLING '08*, 945–952. ACL Press.

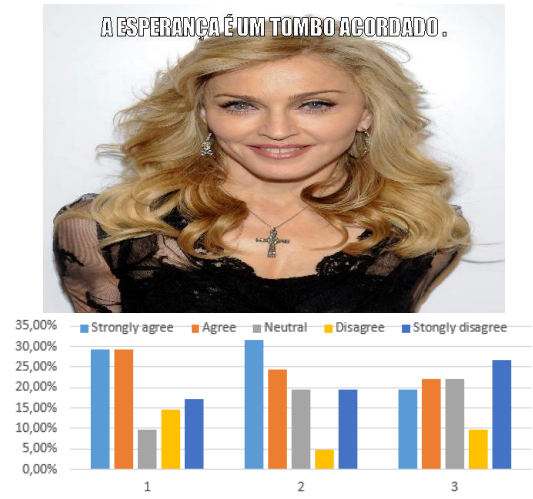
Veale, T. 2014. Coming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Procs. of 5th Intl. Conf. on Computational Creativity (ICCC)*.


| Character | Paulo Portas (Portuguese Deputy Prime-Minister) |
|---|---|
| **Wikipedia words** | *jurista, jornalista, conservador, vice-primeiro-ministro, arquitecto, economista, irmão, dirigente, empresário, ...* (lawyer, journalist, conservative, deputy prime minister, architect, economist, brother, leader, businessman, ...) |
| **Top-5 nouns** | *março* (29), *trajetória* (23), *alteração* (22), *dívida* (16), ~~*do*~~ (14) (march, trajectory, change, debt, ~~of~~) |
| **Context** | He had been talking to the media about the downward trend of the Portuguese public debt, which should start on March. |
| **Original quote** | *Se nós não entendemos a vida, como poderemos entender a morte?* (If we cannot understand life, how can we understand death?) |
| **Generated quote** | *Se nós não entendemos a vida, como poderemos entender a dívida?* (If we cannot understand life, how can we understand the debt?) |
| **Humor probability** | 0.00253 |
| **Frequency** | 0.08 |
| **In Wikipedia** | No |
| **Score** | 0.07177 |

Figure 4: Meme of the Portuguese Deputy Prime-Minister, Paulo Portas.


| Character | Yanis Varoufakis (Greek Finance Minister) |
|---|---|
| **Wikipedia words** | Not in the used version of Wikipedia |
| **Top-5 nouns** | *salário* (37), *governo* (29), *funcionário* (22), *manutenção* (18), *ministro* (17) (salary, government, federal employee, maintenance, minister) |
| **Context** | Since he became a member of the Greek government, he has been all over the news regarding the negotiations between Greece and the Eurogroup. This time, he was negotiating the maintenance of the current number of federal employees and their salaries. |
| **Original quote** | *Na busca da virtude, não temas superar teu professor.* (In the pursuit of virtue, do not fear overcoming your teacher.) |
| **Generated quote** | *Na busca da virtude, não temas superar teu governo.* (In the pursuit of virtue, do not fear overcoming your government.) |
| **Humor probability** | 0.00102 |
| **Frequency** | 0.145 |
| **In Wikipedia** | No |
| **Score** | 0.08696 |

Figure 5: Meme of the Greek Finance Minister, Yanis Varoufakis.




| Character | Madonna (pop singer) |
|---|---|
| **Wikipedia words** | *cantor, compositor, atriz, dançarino, empresário, produtor, álbum ...* (singer, songwriter, actress, dancer, businessman, producer, album ...) |
| **Top-5 nouns** | *tombo* (46), *queda* (21), ~~*madonno*~~ (29), *palco* (21), *vídeo* (13) (tumble, downfall, ~~madonno~~, stage, video) |
| **Context** | A few hours before this generation, she had fell on stage, during a live performance. |
| **Original quote** | *A esperança é um sonho acordado.* (Hope is a waking dream.) |
| **Generated quote** | *A esperança é um tombo acordado.* (Hope is a waking tumble.) |
| **Humor probability** | 0.00231 |
| **Frequency** | 0.23 |
| **In Wikipedia** | No |
| **Score** | 0.10911 |

Figure 6: Meme of the pop singer Madonna.


| Character | Vladimir Putin (Russian President) |
|---|---|
| **Wikipedia words** | *presidente, ex-agente, chefe, serviço, primeiro-ministro, governo, país, ...* (president, former agent, chief, service, prime-minister, government, country, ...) |
| **Top-5 nouns** | *passo* (88), *risco* (36), *país* (36), *intervenção* (34), *vitória* (33) (step, risk, country, intervention, victory) |
| **Context** | He had been on the news due to his role in the Ukrainian crisis. While there is not an international intervention, people try to figure out his next step. |
| **Original quote** | *Em todas as coisas, o sucesso depende de preparação prévia.* (In all things, success depends on a previous preparation.) |
| **Generated quote** | *Em todas as coisas, o sucesso depende de intervenção prévia.* (In all things, success depends on a previous intervention.) |
| **Humor probability** | $7.8 \times 10^{-4}$ |
| **Frequency** | 0.17 |
| **In Wikipedia** | No |
| **Score** | 0.09305 |

Figure 7: Meme of the Russian President, Vladimir Putin.


| Character | José Sócrates (former Portuguese Prime Minister) |
|---|---|
| **Wikipedia words** | *político, secretário-geral, ministério, ordenamento, mestrado ..* (politician, secretary general, ministry, planning, masters, ...) |
| **Top-5 nouns** | *prisão* (93), *mês* (54), *ex-premiê* (41), *retour* (34), *novembro* (33) (jail, month, ex-Prime Minister, comeback, november) |
| **Context** | He has been detained into custody for being a suspect in a corruption case. This week he had got out for a few hours to give his testimony in a court. |
| **Original quote** | *Minha vida é minha mensagem.* (My life is my message.) |
| **Generated quote** | *Minha vida é minha prisão.* (My life is my jail.) |
| **Humor probability** | 0.00975 |
| **Frequency** | 0.465 |
| **In Wikipedia** | No |
| **Score** | 0.17308 |

Figure 8: Meme of the former Portuguese Prime-Minister, José Sócrates.

# A chart generation system for topical metrical poetry

**Berty Chrismartin Lumban Tobing and Ruli Manurung**
Faculty of Computer Science
Universitas Indonesia
Depok 16424, West Java, Indonesia
berty.chrismartin@ui.ac.id,maruli@cs.ui.ac.id

**Abstract**

Several poetry generation systems that are in some way inspired or motivated by existing articles such as newspaper stories have recently appeared. However, most if not all of them employ template-based generation, which limits both the expressiveness of the system and the ability to faithfully convey the message of the source article. In this paper we present our work on a poetry generation system that uses a dependency parser to extract the predicate argument structure of the input article, and tries to maintain this structure through deep syntactic text generation whilst complying with a given target form. The combinatorial nature of this task presents huge challenges, and we describe several improvements that have been applied in an attempt to produce poetry in a tractable fashion.

## Introduction

Poetry generators are systems that are capable of automatically generating poetry given certain restrictions and contexts. Gervás (2002) presents an overall evaluation of various poetry generators. Various generation approaches are employed, e.g. evolutionary algorithms (Manurung, Ritchie, and Thompson 2012), case-based reasoning (Diaz-Agudo, Gervás, and González-Calero 2002), template-based generation (Colton, Goodwin, and Veale 2012), (Rashel and Manurung 2014), and constraint programming (Toivanen, Järvisalo, and Toivonen 2013).

In this paper, the task we are aiming to solve can be referred to as *meaningful poetry generation*, where the goal is to generate a text that exhibits poetic aspects such as rhyme, metre, alliteration, and other phonetic or orthographic patterns, but also broadly tries to convey a given meaning representation. This last requirement is what distinguishes this task from other forms of poetry generation, which primarily focus on generating texts that take the form of a poem.

The way in which an input meaning representation is provided, and the manner in which a poetry generation system attempts to preserve the fidelity of the input meaning representation, varies. Most systems can be said to be "loosely inspired" by their input meaning representations, as they use words and phrases from the input as fillers for textual templates. Although the resulting poems may include words derived from the input, they do not necessarily take into account aspects such as predicate-argument structure and head-modifier relationships that are crucial towards semantic interpretation. For example, it is possible that a poetry generator that extracts keywords from an article about the Gulf War writes a poem about Iraq invading the USA, and not the other way around. Some notable exceptions are PoeTryMe (Gonçalo Oliveira 2012) and McGonagall (Manurung, Ritchie, and Thompson 2012). PoeTryMe selects content in the form of a set of words and relations between them that are obtained from a semantic graph, and conveys this content using templates that are known to express such specific relations. McGonagall employs a fitness function that measures the semantic similarity between a candidate poem and a given target semantics in such a way that structural similarity is significantly preferred. One other interesting related work is Gervás (2015), which explores various modifications and extensions to an existing poetry generation system, WASP, to consider much tighter constraints on the content of generated poems.

This paper describes a system that uses a meaning representation that explicitly captures predicate-argument structure and tries to maintain this structure through deep syntactic text generation whilst complying with a given target form.

We first discuss chart generation, the basic mechanism the system employs to produce text, before discussing how we extract meaning representations from input news articles. Some results from experiments using this initial system are shown. We then present a complexity analysis of the algorithm and suggest four different improvements to make the system generate meaningful metrical poems in a much more tractable manner. Finally, some results are shown from experiments using the revised system.

## Chart Generation

Moreso than an author of prose, an author of poetry may have to perform a lot of rewording, paraphrasing, and various other alterations to the text, in order that the end result can satisfy the various poetic constraints such as rhyme and metre. Moreover, in literary texts, creative language use often results in more exibility of lexical choice, word-order, and grammaticality, hence an even larger search space for the paraphrasing.

One efficient method for constructing all valid para-

phrases of a natural language utterance is chart generation (Kay 1996). Given an input meaning representation, a set of grammar rules, and a lexicon, it systematically generates all syntactically well-formed texts that convey the input meaning. It employs a dynamic programming technique to overcome the inefficiency caused by backtracking due to the pervasive non-determinism in natural language grammar rules.

A data structure known as the chart stores all complete constituents once they are generated, so regardless of the number of paraphrases they may appear in, they will only be constructed once. The chart also stores incomplete constituents, which are predictions of larger constituents yet to be generated. A chart contains entries that are labelled with 'dotted rules' which describe both complete constituents, called inactive edges, and incomplete constituents, called active edges. An active and an inactive edge can combine to yield a new edge that represents a larger constituent.

An example of an inactive edge is $np \rightarrow det\ noun\ \bullet$, which represents a noun phrase ($np$) constituent that consists of a determiner followed by a noun, whereas an example of an active edge is $np \rightarrow det \bullet noun$ which represents a partially constructed noun phrase which is still lacking a noun. Note the position of the dot ($\bullet$) that delineates the portion of the constituent that has been constructed from that which is still lacking.

For chart generation, it is not enough for the dotted rules to simply state syntactic constituency. They must also state the semantics of each constituent, and how their arguments must unify when being combined. When two edges combine, their semantics must also be unioned to obtain the semantics of the new edge. Moreover, some semantic subsumption checking must be performed to prevent false sentences from being generated. For example, given the input semantics $loves(john, mary)$, an edge with the semantics $loves(john, X)$, where the variable $X$ indicates an unbound argument, can be added to the chart, because its semantics still subsume the input. However, according to the input grammar, a chart generator may also construct an edge with the semantics $loves(X, john)$, whose semantics does not subsume the input. Therefore, it must be rejected.

The algorithm can be informally described as follows:

1. Add entries for all words whose semantics subsume the target semantics to the chart.
2. *Bottom-up prediction*: for each inactive edge in the chart, add new active edges to the chart for each grammar rule that have it as the first constituent on the right hand side.
3. *Scanning*: for each active edge in the chart, look for inactive edges whose category matches that of the first constituent needed, and add a new edge that combines the two.
4. *Completion*: for each inactive edge in the chart, look for an active edge that is looking for a constituent with a matching category.
5. The above processes are repeatedly applied to all new entries to the chart until no more new entries can be added.

Let us consider a simple example. Suppose the following target semantics are to be generated: {*dog(d), definite(d), see(s), cat(c), definite(c), arg1(s,d), arg2(s,c)*}.

Assume the grammar consists of the following three rules:
$s(x) \rightarrow np(y)vp(x, y)$
$np(x) \rightarrow det(x)noun(x)$
$vp(x, y) \rightarrow verb(x, y, z)np(z)$

and the lexicon consists of the following four entries:

| Word | Category | Semantics |
|---|---|---|
| cat | noun(x) | x:{cat(x)} |
| saw | verb(x,y,z) | x:{see(x),arg1(x,y),arg2(x,z)} |
| dog | noun(x) | x:{dog(x)} |
| the | det(x) | x:{definite(x)} |

Following the algorithm described above, edges are entered to form the chart seen in Table 1. The process is as follows:

- Initially, edges 1,2,4,6, and 7 enter the chart. They represent the lexical items that convey a portion of the target semantics.
- Edges 3,5, and 8 enter the chart as a result of the *prediction* operation. Based on the grammar, the algorithm hypothesises the existence of larger constituents.
- Edges 9 and 11 enter the chart as a result of combining the inactive and active edges 1+3 and 6+8 respectively.
- Edges 10 and 12 enter the chart as a result of the *prediction* operation on edges 9 and 11. Note that edge 12, although cannot form any part of a sentence that conveys the input semantics, still enters the chart, but will not combine with any other edge due to the semantic subsumption checking.
- Edge 13 and subsequently edge 14 enter the chart as a result of combining edges 5+11 and 10+13 respectively.

## Metre compatibility

Manurung (1999) first introduced an extension to chart generation to take into account rhythmic constraints of poetry.

In most forms of poetry, metre is the arrangement of words such that rhythmic patterns emerge from their lexical stress, which is the relative prominence of stress received by syllables in a word. To simplify matters, we will assume that syllables may receive one of either two types of lexical stress: weak stress or strong stress. Thus, the rhythm of natural language strings can be represented as lists, which we call stress patterns, denoting the type of stress received by each syllable in an utterance, which can be either weak (denoted as '`w`') or strong (denoted as '`s`'). For example, the list `[w,s,w,s,w,s,w,s,w,s]` would be a stress pattern that represents a line of iambic pentameter.

These stress patterns can be used as the representation for specifying the metrical constraints that are provided as input for the chart generator. The starting point for constructing stress patterns is lexical stress, which can be obtained from pronunciation dictionaries such as the CMU Pronouncing Dictionary[1].

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

| No. | Phrase | Category | Semantics | Operator |
|---|---|---|---|---|
| 1 | dog | $noun(d)$ | d:dog(d) | Lexical |
| 2 | the | $det(d)$ | d:definite(d) | Lexical |
| 3 | the | $np(d) \rightarrow det(d) \bullet noun(d)$ | d:definite(d) | Prediction (2) |
| 4 | saw | $verb(s, d, c)$ | s:see(s), arg1(s,d), arg2(s,c) | Lexical |
| 5 | saw | $vp(s, d) \rightarrow verb(s, d, c) \bullet np(c)$ | s:see(s), arg1(s,d), arg2(s,c) | Prediction (4) |
| 6 | cat | $noun(c)$ | c:cat(c) | Lexical |
| 7 | the | $det(c)$ | c:definite(c) | Lexical |
| 8 | the | $np(c) \rightarrow det(c) \bullet noun(c))$ | c:definite(c) | Prediction (7) |
| 9 | the dog | $np(d) \rightarrow det(d)\ noun(d)\bullet$ | d:definite(d),dog(d) | (1)+(3) |
| 10 | the dog | $s(\_) \rightarrow np(d) \bullet vp(\_, d)$ | d:definite(d),dog(d) | Prediction (9) |
| 11 | the cat | $np(c) \rightarrow det(c)\ noun(c)\bullet$ | c:definite(c),cat(c) | (6)+(8) |
| 12 | the cat | $s(\_) \rightarrow np(c) \bullet vp(\_, c)$ | c:definite(c),cat(c) | Prediction (11) |
| 13 | saw the cat | $vp(s, d) \rightarrow verb(s, d, c)\ np(c)\bullet$ | s:see(s), arg1(s,d), arg2(s,c), definite(c), cat(c) | (5)+(11) |
| 14 | the dog saw the cat | $s(s) \rightarrow np(d)\ vp(s, d)\bullet$ | s:see(s), arg1(s,d), arg2(s,c), definite(c), cat(c), definite(d), dog(d) | (10)+(13) |

Table 1: Sample entries during chart generation for "the dog saw the cat"

Stress patterns are not only used to represent input target forms, but also the metre of texts that are incrementally constructed through chart generation. When two edges are combined, their stress patterns are appended to obtain the stress pattern of the new edge that arises. Therefore, when attempting to add a new edge to the chart, the system can first check whether or not its stress pattern can appear as a contiguous subsequence of the target stress pattern. For example, the verb phrase "saw the cat" has a stress pattern `[s,w,s]`, and can thus be said to be compatible with an iambic pentameter metre because it can appear as a subsequence of `[w,s,w,s,w,s,w,s,w,s]`. However, the prepositional phrase "with the cat" has a stress pattern of `[w,w,s]`, and is thus not compatible, and hence should not be added to the chart.

By applying this metre check everytime an entry is attempted to be added to the chart, the search space can be significantly reduced, as it ensures that only texts that satisfy the metre constraints will be added to the chart.

## Implementing topicality

As mentioned above, we aim to generate poems that explicitly convey a given meaning representation, preserving the fidelity of the message by taking into account predicate-argument structure, head-modifier relationships, and lexical semantics.

To that end, we implemented a preprocessing module that obtains meaning representations from a given text, which in our case is a newspaper article from an online website. An input URL is provided, and the main article content is extracted using a popular context extraction tool[2].

The article is split up into sentences, and each sentence is parsed using the Stanford Dependency parser[3] (Klein and Manning 2003). The set of dependency relations produced is taken to be the input meaning representation for the poem to be generated. Strictly speaking, a dependency parse cannot be said to be a genuine semantic representation, as it is still closely related to the constituent structure of the original sentence. Although the dependency relations do include semantic relations of an entity being the *agent*, *subject*, or *object* of another entity, a genuine semantic representation should abstract away from any syntactic decisions, whereas the dependency parse still contains relations such as *advmod* (adverb modifier) and *xcomp* (open clausal complement).

Nevertheless, such a representation is still a useful abstraction from the original text, and arguably does convey the semantics of the original text. In fact, the Stanford CoreNLP tool[4] actually refers to the dependency parse as a "semantic graph". In particular, it represents predicate-argument and head-modifier relations very well. It is precisely such relations that the keyword and phrase extraction-based approaches of previous topical poetry generation systems fail to capture.

For example, given an input sentence "The fox jumps over the dog.", the dependency parse output is as follows:

```
{det(fox-2, The-1),
 nsubj(jumps-3, fox-2),
 root(ROOT-0, jumps-3),
 det(dog-6, the-5),
 prep_over(jumps-3, dog-6)}
```

Since the chart generator will populate the initial chart with lexical entries based on the input meaning representation, whereas the relations above actually define relations between words, we must first explicitly add clauses for each word by introducing a `lex` relation and introduce a new variable for each word. Subsequently, we replace the arguments in the dependency relations to unify with these new variables, yielding the following representation:

```
{lex(a, [the, det]),
 lex(b, [fox, noun]),
 lex(c, [jumps, verb]),
 lex(d, [over, prep]),
 lex(e, [the, det]),
```

[2]https://code.google.com/p/boilerpipe/

[3]http://nlp.stanford.edu/software/lex-parser.shtml

[4]http://nlp.stanford.edu/software/corenlp.shtml

| Synset ID | Gloss |
|---|---|
| #102118333 | alert carnivorous mammal with pointed muzzle and ears and a bushy tail; most are predators that do not hunt in packs |
| #110022759 | a shifty deceptive person |
| #114764910 | the grey or reddish-brown fur of a fox |

Table 2: WordNet entries for the noun "fox"

```
lex(f, [dog, n]),
det(b,a),
nsubj(c,b),
det(f,e),
prep_over(c,f)}
```

### Mapping words to concepts

Although the meaning representation from the dependency parse explicitly states predicate-argument and head-modifier relations, it does so over strings of text such as "fox" and "jumps". To properly treat this as a semantic input, and to maximize the paraphrasing power of the generation component, these strings must first be transformed into semantic concepts. To achieve this, these strings are mapped onto appropriate WordNet synsets (Fellbaum 1998).

This increases the paraphrasing power of the generator, as it enables the generator to select synonyms to convey the concept, which may be necessary to satisfy rhythmic constraints.

Unfortunately, words are ambiguous symbols that may have many meanings, and such a mapping process raises the issue of word sense disambiguation (Agirre and Edmonds 2007). For example, given the word "fox" as a noun, WordNet has three different senses, which can be seen in Table 2.

In the initial version of the system that we develop, we simply take all possible senses for the word given the appropriate part of speech tag as returned by the dependency parser. Thus, in the example of "fox" above, all three senses of the noun are considered, but the three senses of "fox" as a verb are not.

### Lexical resources

To accommodate the input meaning representations obtained from the dependency parser, the grammar, lexicon, and semantic representations must first be suitably modified. The lexicon is constructed by consulting WordNet and the CMU pronouncing dictionary. Before mapping to WordNet synsets, lemmatization is first applied to the words found in the dependency parses. Since WordNet only contains open class words, entries for closed class words such as determiners, prepositions, etc. are added manually.

## Initial Experiments

To summarize the previous two sections, given an input URL and a target form, the poetry generation system proceeds as follows:

*Ask in french surface*
*Call her years, check her*
*Think were toy tennis*
*Skid in chase, land her*
(a)

*Game were this tuesday*
*Is her but basket*
*James were drill friday*
*He were this target*
(b)

*This baby tell me*
*That I can miss you?*
*That you should hold me*
*Will know I miss you*
(c)

Table 3: Sample output of initial experiments

1. Given an input URL, download the page and extract the main news content.
2. Parse each sentence of the text using the dependency parser.
3. For each sentence, apply chart generation to produce a text that conveys target semantics in the form of the target stress pattern.
4. Assemble all possible poems from the successfully generated sentences.

To test this system, three input articles were provided: two news articles from the sports section of the New York Times: "Maria Sharapova Is Finding Her Stride On Clay at Roland Garros"[5] and "James and the Heat Coolly Even the N.B.A. Finals"[6], and the lyrics to a contemporary R&B song, "Officially Missing You"[7]. From each of these input articles poems were generated using target stress patterns of 4 lines long, each consisting of 5 syllables, with a rhyming pattern of AB-AB. For the two news articles a stress pattern of `[s,w,s,s,w]` was specified, but for the song lyrics the generator was only constrained by the number of syllables. Table 3 shows some randomly selected sample output for each input article. Note that they are all perfect in terms of rhyme and metre, and they all roughly convey some aspects of semantics of their respective input articles.

## Improving runtime complexity

Despite the fact that chart generation utilizes dynamic programming to make the process efficient, and that metre compatibility checking can substantially reduce the search space, the system as described is still very inefficient, and takes sev-

[5] http://www.nytimes.com/2014/06/07/sports/tennis/maria-sharapova-is-finding-her-stride-on-clay-at-roland-garros.html

[6] http://www.nytimes.com/2014/06/09/sports/basketball/lebron-james-and-miami-heat-coolly-even-the-series.html

[7] http://en.wikipedia.org/wiki/Officially_Missing_You

eral hours on a modern desktop PC to compute the sample output presented in the previous section.

A brief algorithm analysis will now be presented, together with some insights on how to speed up the process.

Assume that we are trying to generate a poem consisting of $A$ lines based on an input article containing $Z$ sentences. Assume also an input target semantics of $N$ clauses, where for each word appearing in the semantics there are $L$ possible WordNet synsets, with each synset having $K$ synonyms. The lexicon that needs to be considered contains a total of $N \times L \times K$ entries. Finally, assume a grammar that contains $M$ rules.

The chart generation process starts by considering all words from the lexicon that can possibly convey a section of the input semantics, and the bottom-up operator checks the $M$ rules whether they predict the appearance of a word with the appropriate syntactic category. By taking into consideration repeated application of the scanning and completion operators discussed previously, until no more entries can be added to the chart, the theoretical worst case complexity is estimated to be:

$$O((((L \times K)^A \times P(N, A) \times M \times A) \times Z)^P) \quad (1)$$

where $P(N, A)$ is the permutation function, $\frac{N!}{(N-A)!}$.

## Idea 1: Summarizing input text

In our initial experiment described above, the entire input news article is parsed and processed. As an example, the New York Times article about Maria Sharapova consisted of 1198 words. One idea to reduce complexity would be to try to summarize the article beforehand, and extract the semantic representation of the summary as input for the poetry generator instead. Aside from issues of complexity, attempting to convey the meaning of an entire news article in a short poem without really considering issues of discourse processing and coherence is slightly naive. Document summarization systems are precisely designed to analyse a text at the discourse level and to determine the most salient portions. Thus, aside from reducing complexity, this approach may also leverage the ability of such summarization systems to select a subset of content from the input news article that is more relevant to be conveyed.

Assuming that the $Z$ sentences of the news article is summarized into $P$ sentences, where $P$ is the number of lines in the target form to be generated and is $< Z$, the complexity becomes:

$$O((((L \times K)^A \times P(N, A) \times M \times A) \times P)^P) \quad (2)$$

In our experiments, we use the popular document summarization tool MEAD[8] (Radev et al. 2003).

## Idea 2: Sense disambiguation

In our initial version, the system simply considers all possible senses of a word when mapping to WordNet concepts. Given that this is done for all words in the input text, this creates a combinatorial explosion, many of which are likely to be incoherent combinations of senses.

To select the most appropriate word sense, the context of the target word, in this case the sentence in the news article to which it belongs, is compared against the context of the various available senses, i.e. the gloss and/or example sentences from WordNet. The modified Lesk algorithm is a well-known instance of this approach (Banerjee and Pedersen 2002). We employ a vector space model approach, where the two contexts are represented as vectors in a high-dimensional space and the sense that yields the highest cosine similarity is selected as the appropriate sense. In recent years, so-called word embeddings that have been trained using neural networks on very large corpora have yielded very good results. We use pre-trained vectors that have been made available as part of the GloVe[9] (Global Vectors for Word Representation) toolkit.

By applying word sense disambiguation, $L = 1$, thus the complexity becomes:

$$O(((K^A \times P(N, A) \times M \times A) \times P)^P) \quad (3)$$

## Idea 3: Positional indexing

Chart generation is actually a dynamic programming approach to text generation that is motivated by chart parsing, which analyses a sentence and produces all parse trees based on a given grammar. In chart parsing, bottom-up processing starts with adding entries for each word appearing in the text to be parsed. However, since the order of the words is already known, entries in the chart are indexed based on the position they appear in the sentence. This index speeds up the process, since only edges that are incident to each other can possibly combine to yield new edges that represent larger constituent structures.

However, in chart generation such positional indexing is typically not used, as one does not know beforehand where words will appear in the sentence, and the overriding aspect that governs which edges can combine is that of semantic subsumption.

When considering metre compatibility during an attempt to add an edge to the chart, the system currently checks whether it can appear as a contiguous substring in the target form, but does not specify where precisely this substring is located. As a result, this substring matching process must be repeated every time, for every edge. When considering the interaction of this aspect with that of rhyme, it is possible that the chart generator spends a lot of time building partial structures that appear to be valid constructions early on, but eventually cannot fit the metre.

To overcome this, we augment the chart data structure by also recording the start and end position of each edge in terms of the syllable count within the poem. When adding lexical items to the chart at the beginning of the generation process, multiple instances are recorded for each word at each possible position within the poem. However, the metre compatibility check need only be computed once during the beginning, and when the generation subsequently proceeds,

[8] http://www.summarization.com/mead/

[9] http://nlp.stanford.edu/projects/glove/

the system need only ensure that pairs of incident edges are being combined, without having to perform any additional metre substring matching.

The complexity is thus further reduced to become:

$$O(((K^A \times P(N,A) \times M) \times P)^P) \quad (4)$$

Note, however, that due to the additional bookkeeping overhead and redundancy of having multiple entries for words based on the position they appear in, the memory complexity increases.

### Idea 4: Greedy collation

In our initial version above, for each input sentence, chart generation is applied to produce a text that conveys the target semantics in the form of the target stress pattern for one line. Following this process, all possible combinations of these lines are assemble to yield all possible poems. This is a major source of inefficiency. The final modification that is carried out in an attempt to improve the efficiency of the generation algorithm is to replace this exhaustive combinatorial process with a greedy algorithm that selects subsequent lines so as to maximize an objective function that considers the aspects of rhyme, metre, and semantics.

Firstly, all possible candidates for the first line are tried in turn. For each subsequent line $l$, a candidate is selected that maximizes the following objective function:

$$f(l) = \alpha_1 \times rhyme(l) + \alpha_2 \times syll(l) + \alpha_3 \times sem(l)$$

where:

- $\alpha_1, \alpha_2, and \alpha_3$ are weight factors in the interval [0,1] and $\alpha_1 + \alpha_2 + \alpha_3 = 1$.
- $rhyme(l)$ is a function that returns a value of 1 if $l$ ends with a correct rhyme, 0 otherwise.
- $syll(l)$ is a function that returns a normalized syllable count, e.g. the ratio of the number of syllables found in $l$ to the number of syllables in the target form for that line.
- $sem(l)$ is a function that returns a normalized semantic content count, e.g. the ratio of the number of semantic clauses conveyed by $l$ to the maximum number semantic clauses obtained for that sentence during generation.

The complexity is thus further reduced to become:

$$O((K^A \times P(N,A) \times M) \times P) \quad (5)$$

## Subsequent experiment

To test the various modifications that were designed and implemented, the system was run with the exact same input as during the initial experiment, and results can be seen in Table 4. As can be seen, the overall quality of the results suffers as a result of some of the modifications, and possibly most notably the use of a greedy algorithm to assemble the resulting poem. For instance, from the point of view of the rhyme and metre the solutions are sub-optimal.

On the other hand, whereas previously the generator would run for many hours to complete, the empirical running time measurements from the modified system show that the modified system typically takes approximately 20-30 seconds to generate poems given the same size of input.

*Is she*
*Court were full even*
*She were take couple*
*She were a woman*
(a)

*James were a system*
*Are an air system*
*Was a james*
*Aver get way*
(b)

*Tell me are you*
*Wish you now call me*
*Fix over with you*
*Guess was it was I*
(c)

Table 4: Sample output of subsequent experiments

## Discussion & summary

In this paper we have presented work in progress on the development of a poetry generation system that uses a dependency parser to extract the predicate argument structure of the input article, and tries to maintain this structure through deep syntactic text generation whilst complying with a given target form. The combinatorial nature of this task presents huge challenges, and several improvements have been suggested and applied in an attempt to produce poetry in a tractable fashion. Whilst this does drastically improve the complexity of the algorithm, changing the running time from several hours to a matter of seconds, the quality of the output seems to visibly suffer.

Deep natural language generation that is constrained by a target semantics at one end and a target form on the other end is a very difficult task. Whereas other poetry generation systems try to achieve this through the means of evolutionary computation and template-based generation, our work can be seen to be related to the work reported in (Toivanen, Järvisalo, and Toivonen 2013), as the task can be cast as a constraint satisfaction problem. Unfortunately, imposing syntactic constraints on a constraint satisfaction problem, where the syntactic constraints are defined as context-free grammar rules is a very computationally expensive problem. Our approach is to utilize chart generation, a well-known dynamic programming technique where the grammar rules are a fundamental component of the algorithm. Another strategy worth considering for future work is *context-free grammar filtering* (Kadioglu and Sellmann 2008), a time and space efficient arc-consistency algorithm that allows the formal specification of constraints as a context-free grammar within a constraint satisfaction problem framework.

## References

Agirre, E., and Edmonds, P., eds. 2007. *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Banerjee, S., and Pedersen, T. 2002. An adapted lesk al-

gorithm for word sense disambiguation using wordnet. In Gelbukh, A., ed., *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, 136–145. Springer Berlin Heidelberg.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Diaz-Agudo, B.; Gervás, P.; and González-Calero, P. 2002. Poetry generation in COLIBRI. In *Proceedings of the 6th European Conference on Case Based Reasoning (ECCBR 2002)*.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Gervás, P. 2002. Exploring quantitative evaluations of the creativity of automatic poets. In *Proceedings of the 2nd. Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence (ECAI 2002)*.

Gervás, P. 2015. Tightening the constraints on form and content for an existing computer poet. In *AISB 2015 Symposium on Computational Creativity*. University of Kent, Canterbury, United Kingdom: Society for the Study of Artificial Intelligence and Simulation of Behaviour.

Gonçalo Oliveira, H. 2012. PoeTryMe: a versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*, C3GI 2012.

Kadioglu, S., and Sellmann, M. 2008. Efficient context-free grammar constraints. In *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, 310–316.

Kay, M. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, 200–204. Santa Cruz, USA: ACL.

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 423–430. Association for Computational Linguistics.

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.

Manurung, H. M. 1999. A chart generator for rhythm patterned text. In *Proceedings of the First International Workshop on Literature in Cognition and Computer*.

Radev, D.; Otterbacher, J.; Qi, H.; and Tam, D. 2003. MEAD ReDUCs: Michigan at DUC 2003. In *DUC03*. Edmonton, Alberta, Canada: Association for Computational Linguistics.

Rashel, F., and Manurung, R. 2014. Pemuisi: A constraint satisfaction-based generator of topical indonesian poetry. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the Fourth International Conference on Computational Creativity*, 160–167.

# TheRiddlerBot
# A next step on the ladder towards creative Twitter bots

**Iván Guerrero**[1], **Ben Verhoeven**[2], **Francesco Barbieri**[3], **Pedro Martins**[4], **Rafael Pérez y Pérez**[5]
[1]Universidad Nacional Autónoma de México, D.F., México
[2]CLiPS Research Center, University of Antwerp, Belgium
[3]Universitat Pompeu Fabra, Barcelona, Spain
[4]CISUC, University of Coimbra, Portugal
[5]Universidad Autónoma Metropolitana, Cuajimalpa, D.F., México

**Abstract**

We present a computational model for the generation of a Twitter bot that aspires to be considered creative by generating riddles about celebrities and well-known characters. The riddles are created by combining information from both well-structured and poorly-structured information sources. This model has been implemented as an interactive Twitter bot (@TheRiddlerBot) that presents its outputs as contests to its followers, checks the posted answers and replies accordingly. Lastly, we present a discussion about the main attributes of a creative Twitter bot, and the remaining work for our bot to qualify as such.

## Introduction

On several social networks, but especially Twitter, a new variety of users, the bots, are increasingly interacting not only with human users, but even among themselves. The first Twitter bots that appeared on the web were considered in the best case graceful, and sometimes even useful, or helpful, but they were far from being considered creative. To be creative usually relates to the generation of something novel and interesting, not only to oneself, but also to partners sharing a common background (Mayer 1999). According to this, a creative activity can be considered a social activity as well, since the environment evaluates any generational process to determine if it can be considered truly creative or not. In this sense, the environment establishes diverse constraints to any creational process, and the main challenge for an inventor resides in freeing himself from all these conventions to create something novel, interesting and yet valuable.

Novel ways of interacting inside social networks have added new and barely studied constraints to the creative process. Contests for the generation of micro-stories (Hamid 2014) - 100 words long stories -, similar to Tweet messages, or the generation of writing maps - writing prompts to inspire writers - (Maps 2015) have emerged from these new ways of interaction.

The problem that we tackle in this paper is the design and implementation of a Twitter bot that can be considered creative, focusing on missing features in the prevailing bots. The use of more realistic and diverse knowledge sources (Twitter, Facebook, Wikipedia, online news sites), evaluative mechanisms for its own outputs, and the definition of a purpose which surpasses the generation of pseudo-random messages, are examples of such omissions. The goal of our bot is to generate riddles about celebrities, formed as questions to encourage readers to assert the name of a famous character.

The rest of the paper is organized as follows. We first give an overview of the state of the art of Twitter bots, after which we give a general description of a model to automatically generate riddles and its implementation in a Twitter bot (@TheRiddlerBot). We present the results of a questionnaire where we asked people to evaluate a set of riddles. We then close with a general discussion of our proposal and our conclusions.

## Related research

We now present relevant research from two different fields: riddle generation, and automatic Tweet generation. The existing theories related to the generation of riddles are not yet complete mainly because their descriptions only contemplate a subset of riddles (typically those in the question-answer format). Nevertheless, we present several approaches that provide relevant features that should be present in any riddle to be considered as such. Besides, we describe several first generation Twitter bots, Tweet-generating systems that autonomously perform useful and well-defined services (Veale 2014), that are using Twitter in diverse ways. We distinguish feeder bots, which create tons of Tweets for their followers; watcher bots, which are constantly looking for specific texts to extract information; and interactive bots, which ask followers for specific ways of communication and information sharing (Cook 2015). We describe different Twitter bots as examples of the state of the art. We will focus on both the creative aspects and unique features that are already present, as well as missing features.

Pepicello (Pepicello and Green 1984), among others, has researched riddles extensively, and described them as text fragments that employ ordinary language restricted by semiotic, aesthetic and grammatical artistic constraints. They argue that ambiguity in these descriptions is a key

aspect of a riddle, and they define three types: phonological (use of words with the same phonetic code), morphological (use of words with the same writing) and syntactic (phrases with different possible interpretations). According to this work, the goal of a riddle is to confuse the guesser by utilizing one or more of these ambiguities.

Additionally, Weiner (Weiner and De Palma 1993) defines a riddle as a language game, initiated by a question, with the goal to mislead the guesser. They describe two pragmatic mechanisms for the generation and comprehension of riddles: accessibility hierarchy and parallelism. The former relates to categorization, the capability to relate different concepts to accomplish specific goals. They describe parallelism as the tendency to remain in the same cognitive space unless a force makes us change to an alternative representation. They state that we, as humans, employ these two mechanisms to generate and comprehend riddles.

According to this work, there exist two types of concepts present in every category: context-invariant (what first comes to our minds) and context-variant (present when a relevant context appears). They state that a riddle must bring to our minds the context-invariant information to mislead the answer of the riddlee. Parallelism, in turn, helps on to generate false expectations on the part of the guesser.

*JAPE* - Joke Analysis and Production Engine - (Binsted 1996) is a question-answer riddle generation system. Herein, several strategies to generate riddles are described: syllable substitution, word substitution and metathesis. The first mechanism consists in confusing the syllable in a word with a similar sounding word; the second, confuses an entire word with a similar sounding word; the third, reverses the sounds of two words to suggest a similarity in meaning between two phrases.

To generate riddles, JAPE uses templates consisting of 'canned text' with slots where words or phrases are inserted. To determine which words are to be incorporated to the final riddle, the system makes use of predefined schemas, which establish relationships between words which must hold to build a joke. These schemas are manually built from previously known jokes.

In an effort to delineate novel uses of Twitter, Angelina-5 (Cook and Colton 2014) is a software for the generation of 3D games that uses a module to evaluate its textures, i.e. images utilized for decorating walls and ceilings inside the scenario, in a Twitter account (@angelinasgames). Each game has a theme, initiated by a word or phrase. Angelina-5 obtains a set of words associated to it from an English corpus, and uses them to retrieve sound effects, textures, 3D models and fonts to create a game. The bot periodically Tweets images and asks its followers to associate terms to it. These terms are collected into a repository to be further used as tags for the image.

This bot can be classified as a watcher with the goal of obtaining tags for a tweeted image from the user. The bot does not have any capabilities for analyzing the information received, given its very limited function within Angelina-5. Nevertheless, it is a functional example of how bots can receive information from humans to enhance the capabilities of a system, a desirable function to contemplate in our bot.

Flux Capacitor (Veale 2014) is a generator of well-formed and interesting character arcs (conceptual starting and ending points for a character inside a narrative). These character descriptions are defined in terms of properties, and a well-formed arc contemplates representative changes by looking for templates (such as $XbecomesY$) in Google n-grams (Brants and Franz 2006). Apart from that, relationships among properties to describe such states are retrieved from WordNet (Fellbaum 1998). The output of the bot serves the MetaphorIsMyBusiness (@MetaphorMagnet) Twitter bot to generate metaphors related to character twists in a story.

This bot has several aspects that differentiate it from first generation bots, such as its capability to deal with massive, poorly-structured knowledge databases (those lacking a well-defined format), and its purpose to create outputs surpassing the generation of pseudo-random messages. Another aspect of the bot is its high curation coefficient, the ratio of good outputs to all outputs, since the system contemplates mechanisms to evaluate its own outputs and filter those considered with low quality.

## General description

We present a model for a Twitter agent with creative behaviors such as its abilities to utilize real-world, poorly-structured data sources, to evaluate its own outputs, and to interact with Twitter users. We describe as well the implementation of our model in a Twitter bot (@TheRiddlerBot) that generates creative riddles about fictional or real characters (e.g. celebrities) using cross-references from different knowledge bases.

### Model description

The model consists of five main modules each subdivided in three layers (see Figure 1). Each module has a specific task ranging from the selection of a relevant celebrity, to the publication of the riddle in Twitter and tracing the answers of the followers. Besides, a layered structure of the system provides every module of tools for retrieving additional information from diverse sources, for processing the information available, and for evaluating its outputs. Now we describe the main characteristics of each module and how its tasks are distributed among the diverse layers of the model.

**Character selection module** This module initiates by retrieving a list of celebrity names from diverse knowledge bases. Some sources may have well-structured information, such as the Non-Official Characterization (NOC) list (Veale 2015), whereas others may lack this structure, such as

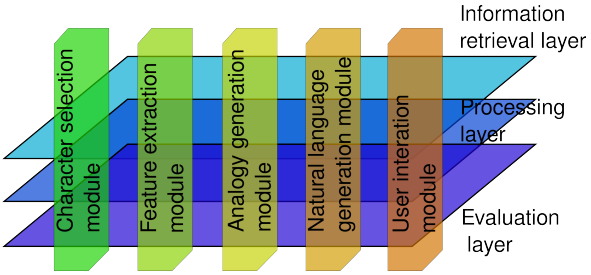



Figure 1: Model architecture

Google News, trending topics from Twitter, or public information from Facebook. This task resides inside the first layer of the module, the information retrieval layer. The data obtained is then passed to the processing layer, where one of the celebrities is selected according to diverse criteria such as his public relevance. These criteria give clues about the current importance of the celebrity due to the events he or she has recently been involved in. Finally, the evaluation layer determines if the selected character has been lately used to generate riddles, in which case it is not suitable for a new riddle. Once the character selection process finishes, the name of the celebrity is passed to the next module to look for as many facts as possible about him.

**Feature Extraction Module** This second module gathers attributes about the previously selected character from both well-structured sources, such as the NOC list, and poorly-structured sources, such as Wikipedia. Furthermore, common sense knowledge bases (see the Perception dataset of the Nodebox project[1]) serve as repositories for hypernyms (super categories) of the character's attributes. These tasks are performed inside the first layer of the module. All the information obtained is then passed to the processing layer, where a subset of features is extracted according to their uniqueness and interestingness. A subset of features is considered unique if they describe only one celebrity. This evaluation is important because a riddle with unique traits is not always desirable, since it becomes easy to solve. A riddle is considered interesting when it describes a character with attributes that altogether represent relevant traits, but do not provide excessive information so that the riddle cannot be easily guessed. A set of attributes is considered relevant when the sum of the n-gram percentage of its elements, according to the Google N-gram viewer. An isolated attribute is considered to provide excessive information when its n-gram percentage is too low, and it can be considered unique. These values still need to be determined and further studies must be done to evaluate its accuracy.

Lastly, the evaluation layer determines if the subset selected has not been previously used for the same character, and that the evaluation of the attributes in previous riddles is acceptable to keep using them. These features are finally sent to the next module to extract additional information from them.

**Analogy Generation Module** The third module initiates by gathering information about similar characters according to the features of the character selected for the riddle. For this purpose, it uses information available at the NOC list as well. Then, it retrieves descriptions of analogies for the generation of relations between characters. We consider two different types of relations between a character and his attributes. Direct relations exist between a character and his features ('Diego Rivera' lived in 'Mexico', 'Tequila' is produced in 'Mexico'); higher-order relations exist between a character and a concept related to one of his features ('Diego Rivera' lived in the country where 'Tequila' is produced). For this last example, we substituted an attribute by its hypernym to create the relation ('country' is a hypernym of 'Mexico').

The information for the generation of analogies is passed to the processing layer where such analogies are created according to the attributes selected for the character. Finally, the evaluation layer determines if the mixture of attributes and analogies has been previously used to create riddles of the same character, in which case the analogies are discarded and new attributes are analyzed. With the set of features and analogies complete, the information is now passed to the next module to convert them into utterances.

**Natural Language Generation Module** This module initiates with the retrieval of different types of phrasal templates for each part of the riddle (initial phrase, clues, final question). These templates are stored and retrieved from a repository specially developed for this project. A phrasal template is a previously-known sentence with slots to be further filled by specific words (Becker 1975). Each slot is commonly associated with a part-of-speech tag which allows to preserve the syntax of the sentence. Inside the processing layer, the module performs a process to select one template for each type of sentence. The selection process begins with the random selection of an initial-phrase template. Then, several clue templates are selected preventing than recently utilized templates are now repeatedly chosen. Lastly, a final-question template is picked in accordance to the first selected template.

These templates have the purpose of providing the system with a wider variety of possible generations. Once a template is selected, the slots are filled with either character's attributes, or analogy information.

**User Interaction Module** The last module extracts a list of aliases for the character to be guessed. The processing layer prepares and tweets the riddle, and starts looking for responses in Twitter, which are compared against the previously obtained aliases. If there is a match, the riddle is considered to be finished. Users get points for each correct answer, which makes this system into a kind of game. When a wrong answer is detected, the user is notified and encouraged to try again. The number of incorrect answers of a

[1] http://www.nodebox.net/perception

riddle can be further employed in the evaluation layer of the module. They cast light on the difficulty level of the riddle, and on the interestingness and uniqueness of the attributes employed to generate it.

## System description

We have been incrementally implementing the previous model in a Twitter bot called 'TheRiddlerBot'[2]. We started out with random character selection, direct relations with tratis, and Twitter publishing capabilities. By now, we have incorporated several new features into the system, which are explained below.

The character selection process retrieves a list of celebrities from the NOC list, and randomly selects one of them. If the character has not been used to generate one of the last riddles, he is passed to the following module. The NOC list is a matrix where every row contains information about a famous character and every column is a trait, so every cell contains the value of a trait for a specific character. Several additional matrices exist where further information about specific traits, such as clothing, fictional worlds, vehicles, weapons..., can be found as part of the project. Due to its simplicity, we utilize the Pattern package in Python (De Smedt and Daelemans 2012) for a wide variety of tasks, from the feature extraction from comma-separated-values files, to the reception and sending of Tweets.

Direct relations between traits and the given character are obtained from the NOC list as well, whereas common sense knowledge is obtained from the Perception demo[3] to incorporate additional traits to the available character's knowledge.

From the list of available traits, three of them are randomly selected and evaluated to determine its interestingness and uniqueness (in the current version this evaluation is not fully implemented yet). The selected traits are considered unique when we cannot find additional celebrities inside the NOC list with the same values. To determine the interestingness of the attributes, we look for them on the character's Wikipedia page, and if they are present, we consider them relevant. If the number of characters sharing the selected values surpasses a threshold, they are not considered relevant, in such case a new subset of features is randomly selected and the generation process re-initiates. Finally, the evaluation layer looks for similar subsets for the same character in previous riddles. If this subset-character pair has been previously used, the feature extraction process re-initiates until a suitable subset is found. These features are finally sent to the next module to determine if additional relations can be obtained.

We retrieve from the additional matrices of the NOC list project information to generate higher-order relations about fictional worlds and group affiliations. For this purpose, we look for characters who share their profession with the previously depicted character. Then, we filter out those characters who don't have any information about their fictional worlds or group affiliations. From the remaining characters, one is randomly selected to create an analogy by means of a template. We randomly depict a template from the repository available inside the system (see table 1).

| Analogy type | Template |
|---|---|
| Fictional world | <$char_1$>: like <$char_2$> in <$world_1$> |
| | <$char_1$>: like someone in <$world_2$> |
| Group affiliation | <$char_1$>: like <$char_2$> in <$group_1$> |
| | <$char_1$>: like someone in <$group_2$> |

Table 1: Sample analogy templates

An analogy template consist of two parts: concept, and reinterpretation of the concept, in the form <$concept$> : <$reinterpretation$>. In general, an analogy template contains redescriptions of the selected character for the riddle (<$char_1$>), in terms of a second character (<$char_2$>), and the fictional world or the group affiliation of one of the characters (<$world_i$> or <$group_i$>). For instance, we can reinterpret the character 'The Joker', whose fictional world is 'The Dark Knight Rises', in terms of another character with the same profession, 'criminal'. In this case, we employ 'Morpheus', who can be considered a criminal in 'The Matrix', and the first template for fictional worlds, to state that 'The Joker' is like 'Morpheus' but in 'The Dark Knight Rises'. Besides, using the second template we obtain that 'The Joker' is similar to 'someone' in 'The Matrix'.

Finally, the evaluation layer determines if the mixture of attributes and analogies has been previously used to create riddles of the same character, in which case the analogies would be discarded and new attributes are selected.

The main task of the language generation module is to represent the attributes and analogies obtained from the previous step as utterances. Inside the feature extraction layer, the date of death of the character is retrieved from his Wikipedia page to determine if he is still alive or not. This information is further employed to conjugate the verbs of the generated phrases (a riddle about a deceased person is written in past tense). Some phrases require additional information of the character to present a more elaborated text, for this reason we obtain positive and negative adjectives describing the character from the NOC list. Inside the generation layer, we convert features and analogies to text. For this task we employ three different types of phrasal templates: introductory templates (see table 2), clue templates (see table 3), and final question templates (see table 4).

For the clue templates, several attributes are available for the system to select one of them. The list includes clothing, opponent, opponent activity, married partner, typical activity, vehicle and country.

Every template consists of three different types of

[2]Source code available on github: https://github.com/ivangro/theriddlerbot

[3]http://www.nodebox.net/perception

| Type | Template |
|---|---|
| First person | I |
| Third person | Tell me the name of a person that |

Table 2: Sample introductory templates

| Feature type | Template |
|---|---|
| Group affiliation analogy | -be/VB the <$char_2$> of <$group_1$><br>-could have belonged to <$group_2$>, but do/VB not<br>-be/VB like <$char_2$> but be/VB not part of <$group_2$> |
| Fictional world analogy | -be/VB similar to someone in <$world_2$><br>-be/VB the <$char_2$> of <$world_1$> |
| Profession attribute | -be/VB <$value$><br>-be/VB <$value$>, <$pos\_adj$> yet <$neg\_adj$> |
| Opponent attribute | -do/VB not like <$value$><br>-be/VB definitely not a close friend of <$value$> |
| Hyperonym attribute | -be/VB known as <$value$><br>-be/VB <$value$>, <$pos\_adj$> yet <$neg\_adj$> |
| Clothes | -have/VB been seen wearing <$value$> |

Table 3: Sample clue templates

elements: words, verbs (marked with the tag VB), and slots (<$slotfiller$>). The verbs are conjugated in accordance with the type of template depicted for the introductory phrase. Afterwards, the attributes and analogies selected in the previous module are converted to text by replacing the slot fillers of the selected template with the values associated to the attributes and analogies. To conclude, the final question template is selected in accordance to the introductory-phrase template. Once the three phrases are generated in natural language, they are chunked as a riddle.

The last module tweets the generated riddle to open a new contest. To determine who wins a contest, we obtain a list of aliases for the character from his Wikipedia page. Every time a follower replies a riddle, his answer is obtained to be compared against each of the available aliases for the character, and if one of them matches, the contest is declared finished and a Tweet is published to point out the winner; if none of the aliases match, a reply to the owner is sent stating that the answer was not accurate, and the contest continues. If, after several hours, the riddle had no correct answer, a Tweet exposing the celebrity is sent, and a new contest begins.

## Example of a riddle

Now, we show how to generate a riddle about 'The Joker'. Once the character selection module finishes, several attributes are obtained for the character from a variety of sources (see Table 5).

| Type | Template |
|---|---|
| First person | Who might I be? |
| Third person | Who is this? |

Table 4: Sample final question templates

| Type | Value |
|---|---|
| Hypernym | 'maniac', 'madman', 'criminal' |
| Group affiliation | 'The Dark Knight Rises' |
| Clothes | 'a purple topcoat',<br>'a green wig' |
| Pos. adjectives | 'playful', 'witty', 'flamboyant',<br>'cunning', 'brilliant', 'creative' |
| Neg. adjectives | 'maniacal', 'cruel', 'sadistic',<br>'inhuman' |

Table 5: Sample attribute and analogy values for 'The Joker'

From the available attributes, a subset of three traits is selected (for this example, group, profession, and clothes), and their corresponding values are sent to the analogy module. If one of the attributes is suitable to generate analogies, the process initiates. In this case, the group attribute is used to create an analogy. We look inside the knowledge base for characters who share a hypernym (see Table 6).

| Character | Group affiliation |
|---|---|
| 'Fagin' | 'Oliver Twist' |
| 'John Dillinger' | 'Public Enemies' |
| 'Fredo Corleone' | 'The Godfather' |
| 'Snake Plissken' | 'Escape From New York' |
| 'Morpheus' | 'The Matrix' |

Table 6: Characters sharing a hypernym with 'The Joker'

With the information obtained from the previous steps, we randomly select an analogy template (<$char_1$>: like <$char_2$> in <$group_1$>), and it is encapsulated, with the rest of the values for the natural language generation. Here, an introductory template is selected ('Tell me the name of a person that'), three clue templates are selected, one for each of the attributes or analogies employed, and a final question template is retrieved as well ('Who is this?').

The clue template for group is (be/VB the <$char_2$> of <$group_1$>), for hypernym is (be/VB <$value$>, <$pos\_adj$> yet <$neg\_adj$>), and for clothes is (have/VB been seen wearing <$value$>). If several values are available for an attribute, one of the is randomly picked to replace the empty slot.

Finally, we create the riddle by chunking the three templates where its slots are replaced with the corresponding values:

> Tell me the name of a person that is the Morpheus of The Dark Knight Rises, is criminal, playful yet cruel, has been seen wearing a purple topcoat. Who is this?

## Model evaluation

As described above, we save all the posted riddles and their metadata (number of retweets, favorites, answers, etc.) in a database. The metadata could be used for the evaluation of the model if we assume that a riddle with more wrong answers is harder or that a riddle with a lot of favorites is better. Unfortunately, our bot is not popular enough yet, so there is very little interaction. Here are some numbers to give you an idea. At the time of writing (April 29th 2015), our bot has 57 followers. Since February 2nd 2015 (date of implementation of the database) 285 riddles were posted. Ten different users gave correct answers to 34 riddles in total.

So we decided to perform a different evaluation. We asked 86 people to each evaluate five riddles. We first asked the participants to guess the answer to the riddle. Then, we presented the correct answer and asked if they knew the person in question. The participant indicated whether he considered the quality of the riddle satisfactory and, if negative, gave us the reason why it wasn't good.

Figure 2 shows the percentage of correct answers (15.58%), and the number of known celebrities (54.19%) once the correct answer was presented.

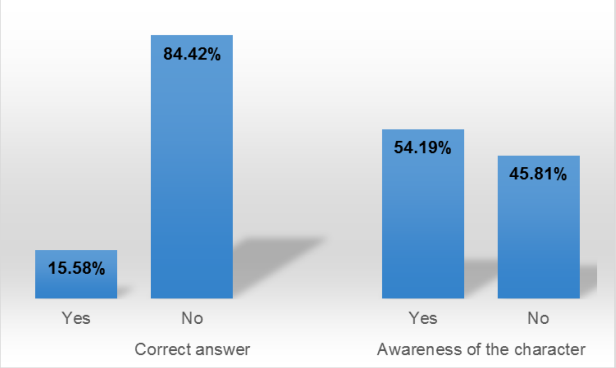



Figure 2: Results for correct answers and good descriptions

Figure 3 shows the number of riddles considered to have accurate descriptions of the characters (41.86%). When that was not the case, the main reason chosen was that the description was too vague (36.51%). Among the additional reasons given, the most recurrent was that the character was already dead and the riddle was written in present (< 1%).

Finally, we present here the top 5 answered riddles, according to the number of times they appeared and the number of correct answers given to them.

Tell me the name of a person that can be found in UK, enjoys robbing from the rich, likes wearing a feathered cap. (Answer: Robin Hood).

Who is a creator, can be found in Italy, wears a paint-stained smock? (Answer: Michelangelo).

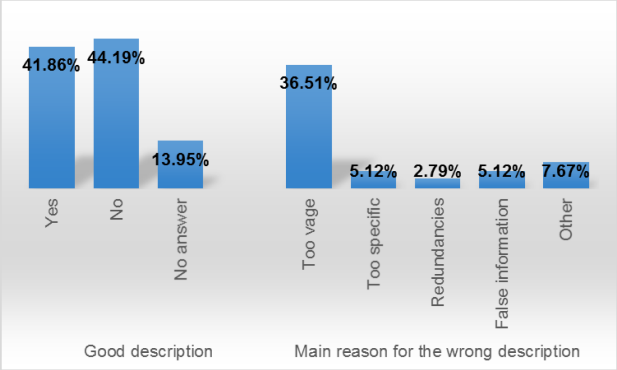



Figure 3: Results about the accuracy of the description

Who is a creative professional, pretty yet superficial, can be found in USA, enjoys monetizing celebrity status? (Answer: Paris Hilton).

Who is a religious leader, loves spreading Christianity, likes wearing sandals? (Answer: Jesus Christ).

Who is the Hermione Granger of The Simpsons, wears an orange dress, is the Timothy McGee of The Simpsons Family? (Answer: Lisa Simpson).

## Discussion and future development

Relevant results were obtained from applying the questionnaire. The percentage of known celebrities once the answer was presented (54.19%) indicates that the process for the selection of celebrities should be improved. From this result we realised that almost half of the riddles could not be correctly answered because people did not have enough information about the character. One reason for this result is that owners of the NOC list (the main source for celebrities) and the riddlees were from different countries, and they did not have enough information in common.

The percentage of good descriptions of the celebrities (41.86%) represents our curation coefficient (the ratio of good outputs to all outputs), and the major cause for our descriptions to be considered wrong was its vagueness. This indicates that further work must be done to improve the interestingness of our riddles (the description of a character with relevant attributes, but without excessive information to be easily guessed). Thereby, additional mechanisms to determine the number of traits to incorporate to a riddle based on its relevance, might prevent descriptions from being too vague.

The low number of correct answers (15.58%) suggests that the complexity of the generated riddles is high. Nevertheless, by improving the character and trait selection processes will mitigate this problem.

In general terms, the current version of the system still lacks selection mechanisms relying on informed decisions. For instance, the character selection process randomly picks a celebrity from a list; the feature selection randomly chooses three character traits, despite of the final evaluation which determines whether they are good enough or not to continue with the process; most of the templates are randomly picked as well, and the values replacing the empty slots in such templates follow the same track. We consider that transforming as many random selections as possible into informed decisions will contribute in an overall increment of the final quality of the outputs generated by our bot, and will provide our model of additional traits for it to be considered creative. A key aspect to distinguish simple generation from creative generation is the curation coefficient in the outputs. To increase the number of high-quality riddles generated by our system several improvements will take place in the next release of the system.

The work presented here is a first step in building up a robust, Twitter bot that can be considered creative. For the next release we still need to improve several aspects related to intermediate output validation, and mechanisms for the automatic expansion of the current knowledge bases utilized by the system.

Despite the fact that several knowledge bases such as Conceptnet (Liu and Singh 2004) or Facebook, are not part of the project yet, the current version of the system already contains fully working mechanisms for information retrieval, and is still pending to exploit this information to generate more interesting, high quality riddles.

## Conclusions

We have described a computational model to generate riddles about celebrities. It consists of modules to select a celebrity, to retrieve relevant traits to describe him, to generate analogies between his attributes and convert such descriptions into utterances, and to tweet the generated riddle and interact with Twitter users by evaluating their answers. The model presents a subdivision of each module in layers. The first layer is responsible for all the data extraction processes; the second, for processing of the information retrieved; the last, for the evaluation and validation of the generated outputs. We consider this layered approach relevant because it provides tools to enrich the intermediate outputs of every module. It contemplates the retrieval of additional information, when required, and the validation of intermediate results to achieve a higher quality in the outputs. We present an implementation of our model in a Twitter bot named 'TheRiddlerBot'. Herein, we introduce several difficulties emerging from reifying our model, such as gathering character traits, generating analogies, and generating natural language utterances.

We consider 'TheRiddlerBot' as a creative agent according to the following considerations. If we describe a creative bot in terms of its capability to deal with poorly-structured knowledge to generate something interesting and novel, we have provided our system with such capabilities. Some authors on the field consider as essential properties for an artifact to be considered creative, novelty, quality and typicality of its outputs (Ritchie 2007). Although similar riddles can be found widespread over the literature, we consider that our system generates novel outputs since the traits employed by our implementation, considering the incorporation of analogies, make them rare to replicate. We still need to implement direct and indirect evaluations for the overall quality of the riddles, but we have sketched in this document several validation mechanisms to ensure the overall quality of our outputs. According to our definition of a riddle, questions to encourage readers to assert the name of a famous character, we argue that our outputs are typical examples of this type of queries.

According to Pérez y Pérez (2013, 2014), any output must be presented in a correct manner (coherence), generate new knowledge to the reader (interesting), and be considered new (novelty) to be creative. We verify the coherence of our riddles particularly in two stages: the analogy and natural language generation models. The analogy and phrasal templates provide the system with well-formed structures to generate complex attributes of a riddle (analogies), and to generate readable phrases written in natural language. During the evaluation layer at every module, we validate the novelty of our riddles, since at every stage of the process we ensure that the intermediate outputs have not been previously utilized. Our system considers a riddle to be interesting looking for the traits to describe a character at his Wikipedia page, and also detecting that we have not utilized the same subset on previous riddles. The first validation gives us clues about the relevance of the traits. If the reader does not know all the presented information, he will be capable of learning new qualitites of a celebrity. The second validation lets the system be certain of the uniqueness of the employed traits.

## Acknowledgements

This research was sponsored by PROSECCO[4]. We would like to thank them as well for organizing the code camp on computational creativity[5] in Coimbra, Portugal, where this research project began and started to grow.

This research was also sponsored by the National Council of Science and Technology in Mexico (CONACyT), project number: 181561.

The second author is supported by the FWO Research Foundation - Flanders.

## References

Becker, J. D. 1975. The phrasal lexicon. In *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, TINLAP '75, 60–63. Stroudsburg, PA, USA: Association for Computational Linguistics.

[4] http://prosecco-network.eu

[5] http://codecampcc.dei.uc.pt

Binsted, K. 1996. *Machine humour: An implemented model of puns (PhD thesis)*. University of Edinburgh.

Brants, T., and Franz, A. 2006. *Web 1T 5-gram database, Version 1*. Linguistic Data Consortium.

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*.

Cook, M. 2015. A brief history of the future of twitterbots. Presented at the PROSECCO Code Camp on Computational Creativity.

De Smedt, T., and Daelemans, W. 2012. Pattern for python. *Journal of Machine Learning Research* 13:2031–2035.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Hamid, A. 2014. Just 100 words.

Liu, H., and Singh, P. 2004. Conceptnet: A practical commonsense reasoning tool-kit. *BT Technology Journal* 22(4):211–226.

Maps, W. 2015. Writing maps.

Mayer, R. E. 1999. *Fifty years of creativity research*. Cambridge, UK: In R.J. Sternberg, Handbook of creativity.

Pepicello, W. J., and Green, T. A. 1984. *The language of riddles: new perspectives*. Columbus, USA: Ohio State University Press.

Pérez y Pérez, R., and Otoniel, O. 2013. A model for evaluating interestingness in a computer-generated plot. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Pérez y Pérez, R. 2014. The three layers evaluation model for computer-generated plots. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Veale, T. 2014. Comming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Proceedings of the Fifth International Conference on Computational Creativity*.

Veale, T. 2015. A game of tropes: Exploring the placebo effect in computational creativity. In *Submitted to the International Conference on Computational Creativity*.

Weiner, J. E., and De Palma, P. 1993. Some pragmatic features of lexical ambiguity and simple riddles. *Language & Communication*.

## Author Index


The Sixth International Conference on Computational Creativity

ICCC 2015

Park City, UT, USA

29 June - 2 July 2015

supported by:


ISBN: 978-0-8425-2970-9