Teaching Computational Creativity

Margareta Ackerman[1], Ashok Goel[2], Colin G. Johnson[3], Anna Jordanous[3], Carlos León[4], Rafael Pérez y Pérez[5], Hannu Toivonen[6], and Dan Ventura[7].

[1]Computer Engineering Department, Santa Clara University. ackermanmaya@gmail.com.

[2]School of Interactive Computing, Georgia Tech. ashok.goel@cc.gatech.edu

[3]School of Computing, University of Kent. {C.G.Johnson, A.K.Jordanous}@kent.ac.uk

[4]Department of Software Engineering and Artificial Intelligence, Complutense University of Madrid. cleon@ucm.es

[5] Depto. de Tecnologías de la Información, Universidad Autónoma Metropolitana. rperez@correo.cua.uam.mx

[6] Department of Computer Science, University of Helsinki. hannu.toivonen@helsinki.fi

[7] Computer Science Department, Brigham Young University. ventura@cs.byu.edu

Abstract

The increasing popularity of computational creativity (CC) in recent years gives rise to the need for educational resources. This paper presents several modules that together act as a guide for developing new CC courses as well as improving existing curricula. As well as introducing core CC concepts, we address pedagogical approaches to this interdisciplinary subject. An accessible overview of the field lets this paper double as an introductory tutorial to computational creativity.

Introduction

As computational creativity (CC) grows in popularity, it is increasingly being taught in some form as a university-level course. However, at this point, there does not exist any cannonical pedagogy nor accepted textbook for the subject; thus it may be useful to begin a discussion on pedagogy, topics, best practices, etc. We feel that a CC course is clearly an important part of the study of (at least) artificial intelligence (AI), and offer some talking points to explain why:

- It provides a method of approaching AI holistically.
- It may appeal to traditionally underrepresented groups in CS and related fields, e.g. female students.
- It is a next step in building more robust intelligent systems.
- It broadens views of constituent parts of intelligence.
- It allows students the opportunity to critically reflect about human and computer creativity (which may help improve their own creativity).
- Being interdisciplinary, it exposes our students to other fields of study, which helps expand their horizons and allows them to experience new perspectives.

To begin the discussion, then, we offer the following as a guide list of learning objectives for a CC course:

- Begin to think about computation in new ways
- Understand central questions and challenges in computational creativity (intentionality, evaluation, abstraction vs. domain specialization, sociality, etc.)
- Understand the difference between mere generation and computational creativity
- Master theoretical and conceptual tools for analysing and discussing creative systems
- Develop and understand one's own creative processes
- Become familiar with the latest advances in CC, and be able to critically discuss current state-of-the-art
- Become able to create CC systems and/or techniques
- Gain the ability to describe, employ and debate methods for evaluation of computational creativity
- Be able to identify appropriate contexts for using CC

In what follows, we offer examples of extant CC systems that can be used as archetypes for introducing key questions in the field; discuss different modeling approaches and how they provide complementary accounts of the theory; address human vs. computational creativity and how both play a role in CC education; discuss some key AI techniques and issues that may be appropriately integrated into a CC course; provide some examples of lessons learned, sample assignments and suggested best practices; and discuss interdisciplinary approaches to CC education.

This paper is aimed at:

- CC researchers who would like to teach computational creativity but may not be sure where to start;
- Those who already teach CC, who would like to expand and further develop their courses;
- Researchers new to CC, but perhaps familiar with AI or related fields, who would be able to use this paper as a starting point to learn about the field.

Introducing Computational Creativity

What is computational creativity, and how should we introduce it to our students? We present a working definition and select several systems that can be introduced in a CC course to illustrate central ideas in the field along with discussion questions targeting these ideas. We discuss several philosophically differing approaches to CC. Lastly, we mention connections between CC and human creativity, which provide another accessible entry point into this field while revealing some of its interdisciplinary facets.

Defining Computational Creativity

Recently the PROSECCO research network has defined computational creativity as "*an emerging field that studies and exploits the potential of computers to be more than feature-rich tools, and to act as autonomous creators and co-creators in their own right. In a CC system, the creative impetus comes from the machine, not the user, though in a hybrid CC system a joint impetus may come from both together.*"(Intro to CC). However, the definition of computational creativity has a complex and argued history, and indeed debates around its definition—or simply the definition of creativity *tout court*—can be an interesting way to engage students in the subject, much as arguments around the nature of intelligence can enlighten a discussion of artificial intelligence. Below we present several alternate approaches to defining CC that can be used to spark this discussion.

Complementary to giving a formal, intentional, definition of CC is to provide examples of systems that have been argued as being creative and to engage students in discussions about why these systems are, or are not, creative. Furthermore, an interesting contrast can be drawn with human creative systems; taking a CC system and a human system with similar types of outputs, and considering whether one is creative and the other not, is an interesting and informative exercise.

Some of the systems that we have found useful—and the CC topics/questions that they are useful in introducing to students— include:

- *The Painting Fool* (Colton 2012). Is it necessary for CC systems to be able to explain their creative decisions? How can a CC system draw on external world knowledge? Does engaging with artworld economics help us value the creativity of a CC system?
- *AARON* (McCorduck 1991). To what degree can a system of parameterised outputs and constraints be considered to act in a creative way? Where does the creativity lie in a system that has been developed over several decades, where the outputs from the system have influenced the development of the system?
- Emotive music generation (Monteith 2012). How important is evoking an emotional response in an audience for an artistic CC system? Does it matter that computers can't "feel" the emotion? Does a human creative artist need to be able to feel the emotion that they are aiming to engender in their audience?
- Experiments in Musical Intelligence (Cope 1996). What kind of creativity can be generated by analysis and abstraction from a corpus of existing material? Where does pastiche end and creativity begin? What is the role of inspirational examples in CC systems?
- Mexica (Pérez y Pérez 1999). What are the differences between the creativity needed to create structures (e.g. for stories) and that for creating language?
- The Joking Computer (Ritchie and Masthoff 2011). How can something that cannot laugh create jokes? Can a CC system that consists of filling in templates be regarded as creative? Would a system's outputs be regarded as creative if generated by a child?
- DARCI (Norton, Heath, and Ventura 2013). What role does intention play in creativity? How do perceptual grounding and communicating intention relate?
- IDEAL (Goel and Bhatta 2004) How do designers generate and evaluate design ideas for novel problems? What role does analogical thinking play in idea generation? What role does systems thinking play in evaluating a design idea?
- MILA (Goel and Joyner 2015) How do scientists make new discoveries? And model observed phenomenon? How can we support citizen scientists and student scientists in conducting authentic science?
- Poem Machine (Kantosalo et al. 2014) How can a computer and a human collaborate in co-creating? What does it take to transform a creative system to support human-computer co-creation? Can creative authorship be shared?
- Artificial Creativity (Saunders and Gero 2001) How do societies of creative agents behave? When is creativity an emergent property of a society?
- ER-Model (Pérez y Pérez and Sharples 2001). Can a computer model of creativity be used in multiple domains? What are the differences and similarities between those implementations?

Approaches to Computational Creativity

Several different approaches to computational creativity have been proposed. Here we discuss some notable approaches suitable for introducing CC. In particular, we contrast the operational, product-centered approach with the cognitive perspective that focuses on the creative process. Another popular model, expressing creative methods as a search problem, is also discussed.

Cognitive vs. Engineering Approach One approach to the problem of computational creativity might be characterized as operational, *as the study and simulation, by computational means, of behaviour, natural and artificial, which would, if observed in humans, be deemed creative* (Colton and Wiggins 2012). As Jordanous points out, from this perspective "the challenge is to engineer a system that would be judged to be creative by its audience, rather than engineering a system that possesses a level of creativity existing independently of an audience's perception" (Jordanous 2012b). In general terms, this kind of approach employs mathematical models and engineering methods.

A second approach might be characterized as an interdisciplinary study of the creative process employing computers as the core tool for reflection and generation of new knowledge (Pérez y Pérez 2015a). This perspective emphasizes the importance of contributing to the understanding of the creative process. It attempts to help answer questions such as, how do we conceive new ideas? How can we produce coherent sequences of actions during the creative act? How do we assess the quality of an artifact? This approach is motivated by the work of philosophers, sociologists, cognitive psychologists, and so on.

In this way, the engineering/mathematical perspective concentrates on products and processes that would be validated by an audience, while the cognitive point of view privileges the generation of insights about the phenomenon we are studying. We can imagine these ideas as a continuum, on which each of the extremes represents one of these approaches. We refer to it as the CC-continuum Most of the systems and models that have been developed in recent years can be placed along this continuum. However, we are aware that there are other possible classifications.

We believe that the CC-continuum can be a valuable tool for teaching CC. It provides a wide view of the kind of systems that have been developed and invites students to contrast their core features. In the same way, it might be useful to assist students in developing analytical skills. For instance, each time a new system or model is studied and discussed in the classroom, the students may be asked to locate it within the continuum and justify their reasons.

Creativity as Search Another abstraction for studying and describing creative methods is to view them as search in some space of potential artifacts (Wiggins 2006). According to Wiggins' framework, a creative system can be defined essentially by three components: 1) a search space of valid artifacts, 2) an evaluation method that indicates whether or not an artifact is valued, and 3) a search method to traverse the search space for valued artifacts.

This provides a relatively simple and generic approach for describing and comparing CC systems. For instance, given two systems that produce stories, how do they differ in what they consider to be valid stories? How do their methods for measuring if a story is good compare? How do they find (generate) good stories?

The conceptual division of creative operations into three distinct components (search space, value, search method) supports discussion on various ways to achieve transformational creativity. A system is *transformationally creative* if it carries out informed changes on its own operation. Transformational creativity is interesting because a system that is not transformational can be argued "to just do what it was told to do". Using the framework of Wiggins (2006), it is natural to ask if, and explain how, a system possibly modifies its search space, its way of valuing artifacts, or its method of searching for them.

Human and Computer Creativity

Research into computational creativity in the cognitive systems paradigm is related to human creativity much like research into artificial intelligence in the paradigm is related to human cognition (Goel and Davies 2011). This is in part because humans are the recipients of the products of computational creativity and in part because computational creativity is (at least) indirectly inspired by human creativity.

More generally, this is a two-way relationship. In one direction, theories of human creativity generate hypotheses and models for realizing computational creativity. In the other direction, computational creativity techniques act as hypotheses for understanding creativity in humans. On one hand, theories of human creativity provide constraints for techniques of computational creativity so that the latter are meaningful to the former. On the other, implementation on computers helps evaluate theories of creativity in humans.

A CC course may introduce both the core processes of human creativity and their use in computational creativity. The creative processes addressed may include (1) Design Thinking (thinking about ill-structured and open-ended problems with ill-defined goals and evaluation criteria) (2) Analogical Thinking (thinking about novel situations in terms of similar, familiar situations), (3) Meta-Thinking (thinking about one's own knowledge and thinking), (4) Abductive Thinking (thinking about potential explanations for a set of data), (5) Visual Thinking (thinking about images and in images), and (6) Systems Thinking (thinking about complex phenomena consisting of multiple interacting components and causal processes) (Goel et al. 2015).

The study of human creativity also offers an accessible entry point for students new to CC, especially for interdisciplinary classrooms that include students whose primary field of study is outside of computer science. The history of human creativity and myths surrounding this concept (such as the idea that children are more creative than adults)(Sawyer 2011) can act as a foundation for reasoning about computer creativity. A summary of psychology research on how humans can become more creative (Sawyer 2013) can further their understanding of creativity while directly strengthening their own creative capacities.

From Generation to Evaluation

It may be argued that the two most basic abilities for any CC system are that of *generation*—the ability to produce something—and *evaluation*—the ability to assess something. Indeed, a large proportion of the scientific contributions to the field has been historically devoted to *generative systems*.[1] As such, it is likely safe to suggest that generative techniques are necessary, if not sufficient, for computational creativity. It then follows that teaching CC should address computational generative methods. While work on evaluation has lagged somewhat behind that on generation, it is equally critical both for any argument of creativity in a particular system and for assessing the development of the field as a whole.

[1] An informal review of the papers appearing in *ICCC 2016* yields 21/32 papers ($\approx 66\%$) discussing generation.

Generative Methods

We assert that teaching CC must address the application of (generative) AI programming techniques, but the question of which ones should be considered fundamental assets is less clear. The following list describes the application of some common AI approaches to CC:

- State space search (part of what is commonly known as *GOFAI*) is fundamental in computational creativity. There are many models of creativity that address problems in terms of *spaces* and artifacts, *distance* between concepts and other objects that can be naturally modelled as refined forms of state space search (Wiggins 2006; Riedl and Young 2006).
- Markov chains have been explored in several subfields, most remarkably in music generation (Eigenfeldt and Pasquier 2010; Nierhaus 2009).
- Knowledge intensive systems (schemas, frames, grammars, rules, etc.) are also among the most common techniques, especially in domains which require some level of knowledge management, like narrative (Álvarez, Pérez y Pérez, and Aliseda 2007; Veale 2013; Bringsjord and Ferrucci 2000).
- Genetic algorithms and evolutionary approaches are often used for performing a more informed search (Baydin, De Mantaras, and Ontanon 2012; Unemi 2014).
- Learning/adaption via statistical methods has also been explored (Bickerman et al. 2010; Maher, Merrick, and Saunders 2008; Toivonen and Gross 2015).

This list is far from exhaustive, and yet it is already long enough to justify more than one course, even if the students are already familiar with some or all of the fundamentals of these techniques. While some treatment of these topics seems necessary for any CC curriculum, covering all of them is likely unnecessary for engaging and teaching the core foundations of the generative methods involved in CC. As discussed throughout this paper, different courses (current and future) tackle the problem from different perspectives.

It may be the case that no single generative approach is sufficient, and it is therefore common to employ heterogeneous methods to obtain an eclectic synergy—sometimes it is the combination of procedures itself which defines potentially creative characteristics of a system. For example, one might combine search, knowledge and evaluation (León and Gervás 2014).

Evaluation as Part of the Creative Process

The ability of a system to observe and assess its own performance is an elementary requirement for any attribution of self-awareness or intent to the system, and therefore a key criterion for creativity. Internal evaluation of its own products not only tells the system how well it is performing but also allows it to change its behavior in an attempt to perform better. A system that takes more creative responsibility could even change its standards and modify its own goals.

Jennings (2010) gives an accessible account of *creative autonomy*, the ability of a system to evaluate and to change its standard. While evaluation itself obviously is domain and application specific, the role and importance of evaluation is discussed in the literature more generally. Interestingly, the ability to evaluate artifacts relates to social aspects of creativity in several ways. First, though it may sound paradoxical, creative autonomy actually requires a social setting — interaction with other systems seems the most feasible way for a system to obtain information with which it can change its own standards. Second, a system (or an agent) can contribute to other agents and the society as a whole by evaluating artifacts produced by others.

In its simplest form, an internal evaluation function can be used to implement a generate-and-test architecture: produce an artifact, evaluate it, and only output it if it is good enough. This style of operation coincides with some models of human and social creativity, as described by Saunders and Gero (2001).

Evaluation of Creative Systems

As well as playing a role in the creative process itself, evaluation is also a key part of the scientific research process. We evaluate our creative systems so that we can understand what has been achieved as well as what needs further improvement. A number of key evaluation methodologies and approaches exist for evaluating how successful a 'creative system' actually is at being creative (listed in date order):

- 2001/2007: Ritchie's empirical criteria for assessing a system by examining the typicality/atypicality and novelty of its output against 18 formally stated criteria (Ritchie 2007).
- 2008: The Creative Tripod: a system is a potential candidate for being creative if it demonstrates *skill, imagination and appreciation* (Colton 2008).
- 2011: The FACE model: provides an abstract framework for qualifying different kinds of creative acts performed by a system, both at the product and process level. These acts include Framing information, Aesthetic measures, Concepts, and Expressions (of a concept) (Colton, Charnley, and Pease 2011).
- 2012: SPECS: the Standardised Procedure for Evaluating Creative Systems methodology requires the evaluator to evaluate the system based on standards or criteria that are drawn from a definition/characterisation of creativity that the system is intended to implement (Jordanous 2012a).
- 2004/2015: Pérez y Pérez suggests that, besides being able to generate novel, coherent and interesting (or useful) products, a creative agent must fulfill the following constraints: 1) Employ a knowledge-base to build its outcomes; 2) Interpret its own outputs in order to generate novel knowledge that is useful to pro-

duce more original pieces; 3) Evaluate its own products; such an evaluation must influence the way the generation process works (Pérez y Pérez and Sharples 2004; Pérez y Pérez 2015a).

- 2016: Ventura discusses various thresholds for differentiating mere generation from true creativity and proposes a spectrum of abstracted prototype CC systems that can be used as landmarkers by which specific CC systems can be evaluated for their relative creative ability (Ventura 2016).

It is important that students are given opportunities to practice using evaluation frameworks to analyse creative systems, alongside practical work in making the systems themselves. For example, a practical lab class can be set where students try out different evaluation techniques on existing CC systems. For an end-of-module project assignment in which students produce a CC system, they could be required to evaluate their system in a methodical way, such as one of those listed above.

Projects and Assignments

It is fitting for a CC course to include assignments that not only reinforce the material but also help students develop their own creative capacity. To this end, we present several assignments that have been successfully applied in CC courses, with the hope that such experience will simplify design of future curricula.

- **TweeterBots**: Used as an introductory assignment, the design of a simple TweeterBot allows students to engage with several aspects of CC early in the course. The objective of this assignment is to design a bot such that at least some of its tweets could have been made by a reasonably creative person. Achieving two district objectives, the creation of a bot gives students a chance to engage their creative capacities when coming up with a novel concept for a bot, while on the technical side allowing them to start delving into generative methods. It is also an excellent demonstration of *mere generation*, which students are asked to surpass later in the course.
- **Art Conversion**: It is often argued that creativity, as found in the arts, aims to express an artist's emotions. The art conversion assignment challenges this notion by asking students to extract the emotional content of artifacts in one art form, and then use this content to create art in a different medium. For example, students may perform sentiment analysis on a poem, then generate music that aims to capture the same feelings as the poem. Similarly, one may begin with a melody, and create visual art that expresses the same mood. This assignment asks students to consider what makes creative artifacts impactful on art consumers, and how computer systems may incorporate this facet of creativity. An evaluation phase may also be integrated into this assignment.
- **Computational Model of Creativity**: Students are asked to produce a high-level computational model of the creative process, focusing on the big picture and the computational aspects of the model, and accounting for as much of the "theory" as possible (i.e. they should say *what* they think the model should include and *why* they think that way and *how* they might actually do that computationally). This is typically assigned within the first week of the semester, to give the students the chance to wrestle with difficult questions before they've been exposed to the current literature.
- **Paper Debate**: Students are given a paper to read ranging in content across various CC debates and applications and some students are selected to informally present their thoughts on the paper, to start off discussion. To help give a rounded view of the paper, some students are asked to present the good points of the paper and others to critique weaknesses of the paper (though we are flexible in whether students actually stick to their assigned 'role' here!) Following the brief presentations, the students critique the paper and the issues it raises, in a group discussion. Students are assessed on the level of critical thinking in their presentations and the quality of their contributions to discussions.
- **Large, Self-proposed Projects**: Requiring students to design and build a major project over the course of an entire semester (possibly as a member of a team) teaches them about inquiry in the processes of both system design and scientific discovery. Such projects require not only eventual solution of design problems but also initial inquiry to understand the design problems. A significant part of creativity lies in this initial inquiry that entails question-answering, problem understanding, and problem formulation—before problem solving commences. By developing semester-long projects that incorporate this philosophy of design inquiry students are led not only to the answers to some questions but also to the formulation of new questions.
- **Project Assessment**: A useful way of integrating several aspects of computational creativity is requiring students to describe and assess their own projects using the concepts covered during the course. For instance, how could the project be described using the FACE model (Colton, Charnley, and Pease 2011) or Wiggins' model of creativity as search (Wiggins 2006)? Which creative acts does it perform? Is it transformationally creative? And so on. Such an assessment task links abstract conceptual issues to concrete implementations, helping students learn more about both. It is important to announce this assessment task at an early phase of the CC course, to motivate learning of the more conceptual material and to encourage students to develop their projects in more creative directions. The project assessments made by students can be leveraged when grading, in

assessment of both the project and conceptual skills.

A project could be made more concrete by asking students to select one or more specific generation methods, an application domain, and a formal evaluation metric amongst those introduced throughout the course. These three components could then be used as a starting point for designing a comprehensive course project.

One assessment method that has proven less successful is a traditional end-of-module examination. Particularly because CC is an area that often allows students to explore any domain of interest (complementing core CC topics), we have found that it is less appropriate to ask students to review an entire course worth of content for an exam scenario; and it is difficult to set exam questions that evenly cover a CC module's content. Many of the assessments listed can be successful alternatives for midterm/finals based around students investigating topics of their choice. Other possibilities for an exam include giving the students selected papers to read and critique during an exam or a take-home exam asking them to answer questions about the development and analysis of systems they have built during the course (possibly using the format of an *ICCC*[2] paper type).

Interdisciplinary Approaches

Computational creativity is inherently interdisciplinary. This sub-field of artificial intelligent intersects directly with many other fields, including psychology, cognitive science, mathematics and engineering, to name a few, and indirectly with any number of application domains, from musical composition to the culinary arts to scientific discovery. A CC course that combines perspectives from different disciplines can offer an eye-opening experience to its students, by letting them see the world from a different point of view.

Students tend to choose their field of study at a young age, often while still in their teens, and henceforth find themselves in learning environments with homogeneous perspectives and values. Interaction with faculty or students with fundamentally different mindsets not only expands their horizons but also can lead them to questions and opportunities that only arise when two or more fields are combined. Even for classes containing students from a single discipline, they often bring knowledge from personal creative hobbies, helping them relate to the course content on an individual level.

Interdisciplinary connections can be formed with application domain experts (e.g., artists, musicians, dancers, poets) by inviting speakers to discuss their field of expertise, their views on creativity and connections between their field and computing. Assignments can be focused on teaching CC concepts through applications in the relevant domain (e.g., visual art, music, dance, poetry). Projects bringing together students across diverse disciplines (such as computing and dance) has led to original research (Brockhoeft et al. 2016).

Continuing the idea further may involve co-teaching a course with faculty from other disciplines. Different disciplines use their own languages to communicate often distinct goals, and building a common base of understanding can be challenging. For example, in a class of CS and MFA students, co-taught by CS and Visual Arts faculty, the computer scientists could become frustrated that the artists do not think algorithmically (nor perhaps even understand the term) while the artists could believe that discussing art from an algorithmic perspective entirely misses the point at best and completely ruins the concept at worst. However, being placed in an environment that requires effort to understand new ways of thinking broadens the perspective of both sets of students and can result in a successful interdisciplinary experience (Norton, Heath, and Ventura 2011)].

Another variant of the interdisciplinary classroom includes students from multiple fields (e.g., architecture, engineering, computing and business) working together on an interdisciplinary project, for which their various skills complement each other. The inclusion of interdisciplinary students in teams results in more authentic/impactful projects, such as recent interdisciplianry projects on computational creativity in biologically inspired design (Goel et al. 2015).

An interdisciplinary classroom in any of its forms offers a variety of challenges, though, as discussed above, the rewards are often worth the effort. To help navigate this challenge, Pérez y Pérez (2015b) describes six features for interdisciplinary work, which apply to the interdisciplinary CC classroom. The first principle addresses the need for awareness of how our disciplinary training shapes our skills and vision of the world, allowing us to compare them with those of others. The second feature addresses the need for a common vocabulary, which makes it possible to communicate across distinct disciplines. Third is the development of academic empathy, that is, the capacity for seeing a problem from the (methodological and epistemic) perspective of others. Fourth is developing trust. Fifth is a willingness to confront ideas and to reach consensus (particularly as related to methodological and epistemic issues). The final feature is good leadership. The leader's main task is to clearly demarcate the roles and responsibilities of people from different disciplines, which can otherwise be confusing, or even intimidating, when working in an interdisciplinary setting. This framework allows students from diverse academic backgrounds (engineers, mathematicians, physicist, psychologists, philosophers, anthropologists, etc.) to analyze different CC models and discuss them from a variety of viewpoints.

Discussion and Conclusions

This paper is written from the perspectives of a group of computational creativity researchers who teach various CC modules to university students from around the world. We intend this paper to act as a guide for those who would like to start to teach CC, or those who already teach CC and would like to review their teaching.

[2] *The International Conference on Computational Creativity* (ICCC) is the primary conference in the field.

The overarching aim of this paper is to assist people in teaching CC, and to help simplify the process of putting together a CC course. We outline several areas we feel are necessary for such an endeavor; as such, this paper also functions as a guide to computational creativity for someone who is new to the area but familiar with related fields such as AI.

We begin by outlining several learning objectives that would help to shape and guide a new module in computational creativity. Then, we consider how to address the question 'what is computational creativity' with students. We consider definitions of CC, highlight some notable CC systems to cover as examples, and identify different ways to approach computational creativity. Again with the view of identifying key content to teach CC students, we also discuss the roles of generation and evaluation in computational creativity.

For educational purposes, assignments and projects play valuable roles in helping our students learn. From our experience teaching computational creativity, we outline a variety of assignment types that have worked well. We note that assignments can not only help reinforce the taught material, but also help our students develop their own creative capacity.

To conclude, the process of sharing our experience leads us to some interesting overarching observations about teaching computational creativity. CC is a highly interdisciplinary area of research, and we reflect on the implications of this for our teaching. Students react to the interdisciplinary nature of computational creativity in different ways; this depends on their ability to adapt to course content in which there are philosophical as well as practical issues to be dealt with, and perhaps no 'right answers' to be found. Many of our courses are taught mainly to computing students. Coming from a background of being taught objective material: facts, algorithms, formulae, programming, etc, the students must adapt to the subjectivity of computational creativity, just as an art student would need to adapt to objective approaches to creativity. In teaching to students from any discipline, we need to remain mindful of the open-mindedness that is necessary for students to more deeply appreciate the depth of CC content, outside of what they are used to from their own disciplines.

During a debate on teaching computational creativity at the *Seventh International Conference on Computational Creativity* in Paris in 2016, we discussed how we could share resources for CC education across the research community. For example, we may find it easy to write lectures and exercises on our specialist areas of computational creativity but not have the same depth of specialist knowledge for areas slightly removed from our own research interests. The Association for Computational Creativity website already hosts plentiful useful resources on CC research; it was, therefore, suggested during the 2016 discussion as a good option for a central repository for sharing teaching resources as well. We created a page (`http://computationalcreativity.net/home/teaching-cc/`) dedicated to this purpose, and hope that this resource will grow as computational creativity education continues to expand.

References

Álvarez, M.; Pérez y Pérez, R.; and Aliseda, A. 2007. A Generative Grammar for Pre-Hispanic Production: The Case of El Tajín Style. In Cardosa, A., and Wiggins, G. A., eds., *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 39–46. London, UK: Goldsmiths, University of London.

Baydin, A. G.; De Mantaras, R. L.; and Ontanon, S. 2012. Automated generation of Cross-Domain analogies via evvolutionary computation. In Maher, M. L.; Hammond, K.; Pease, A.; Rafael Pérez; Ventura, D.; and Wiggins, G., eds., *Proceedings of the Third International Conference on Computational Creativity*, 25–32.

Bickerman, G.; Bosley, S.; Swire, P.; and Keller, R. M. 2010. Learning to Create Jazz Melodies Using Deep Belief Nets. In Ventura, D.; Pease, A.; Pérez y Pérez, R.; Ritchie, G.; and Veale, T., eds., *Proceedings of the International Conference on Computational Creativity*, 228–237. Lisbon, Portugal: Department of Informatics Engineering, University of Coimbra.

Bringsjord, S., and Ferrucci, D. a. 2000. Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine. *Computational Linguistics* 26(4):642–647.

Brockhoeft, T.; Petuch, J.; Bach, J.; Djerekarov, E.; Ackerman, M.; and Tyson, G. 2016. Interactive augmented reality for dance. In *Proceedings of the Seventh International Conference on Computational Creativity*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European conference on artificial intelligence*, 21–26. IOS Press.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational Creativity Theory: The FACE and IDEA Descriptive Models. In *Proceedings of the 2nd International Conference on Computational Creativity*, 90–95.

Colton, S. 2008. Creativity versus the Perception of Creativity in Computational Systems. In *Proceedings of AAAI Symposium on Creative Systems*, 14–20.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack, J., and D'Inverno, M., eds., *Computers and Creativity*. Berlin, Germany: Springer-Verlag. 3–38.

Cope, D. 1996. *Experiments in Musical Intelligence*. A-R Editions.

Eigenfeldt, A., and Pasquier, P. 2010. Realtime Generation of Harmonic Progressions Using Constrained Markov Selection. In Ventura, D.; Pease, A.; Pérez y Pérez, R.; Ritchie, G.; and Veale, T., eds., *Proceedings of the International Conference on Computational Creativity*, 16–25. Lisbon, Portugal: Department of Informatics Engineering, University of Coimbra.

Goel, A., and Bhatta, S. 2004. Use of design patterns in analogy-based design. *Advanced Engineering Informatics* 18(2):85–94.

Goel, A., and Davies, J. 2011. Artificial intelligence. In Sternberg, R., and Kauffman, S., eds., *Handbook of Intelligence*. Cambridge University Press, 3 edition. chapter 23, 468–484.

Goel, A., and Joyner, D. 2015. Impact of a creativity support tool on student learning about scientific discovery processes. In *Proceedings of the Sixth International Conference on Computational Creativity*.

Goel, A.; Creeden, B.; Kumble, M.; Salunke, S.; Shetty, A.; and Wiltgen, B. 2015. Using Watson for enhancing human-computer co-creativity. In *Proceedings of the AAAI 2015 Fall Symposium on Cognitive Assistance*.

`http://prosecco-network.eu/introduction-computational-creativity`. ProSecco Network website; visited January 2017.

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

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. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Kantosalo, A.; Toivanen, J. M.; Xiao, P.; and Toivonen, H. 2014. From isolation to involvement: Adapting machine creativity software to support human-computer co-creation. In *Proceedings of the Fifth International Conference on Computational Creativity. June 10-13, 2014, Ljubljana, Slovenia*, 1–8.

León, C., and Gervás, P. 2014. Creativity in Story Generation From the Ground Up: Non-deterministic Simulation driven by Narrative. In *5th International Conference on Computational Creativity, ICCC 2014*.

Maher, M. L.; Merrick, K. E.; and Saunders, R. 2008. Achieving Creative Behavior Using Curious Learning Agents. In Ventura, D.; Maher, M. L.; and Colton, S., eds., *AAAI Spring Symposium: Creative Intelligent Systems'08*, Technical Report SS-08-03, 40–46.

McCorduck, P. 1991. *Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. W. H. Freeman & Co.

Monteith, K. P. 2012. *Automatic Generation of Music for Inducing Emotive and Physiological Responses*. Ph.D. Dissertation, Provo, UT, USA.

Nierhaus, G. 2009. *Algorithmic composition: Paradigms of automated music generation*.

Norton, D.; Heath, D.; and Ventura, D. 2011. An artistic dialogue with the artificial. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 31–40. New York, NY, USA: ACM.

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

Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

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.

Pérez y Pérez, R. 1999. *MEXICA: A Computer Model of Creativity in Writing*. DPhil dissertation, University of Sussex.

Pérez y Pérez, R. 2015a. A computer-based model for collaborative narrative generation. *Cognitive Systems Research* 36:30–48.

Pérez y Pérez, R. 2015b. Reflexiones sobre las características del trabajo interdisciplinario y sugerencias sobre cómo fomentarlo en el aula universitaria. In *En Vicente Castellanos (Ed.) Estudios Interdisciplinarios en Comunicación. México D. F.: UAM Cuajimalpa.*, 33–50.

Riedl, M., and Young, M. 2006. Story Planning as Exploratory Creativity: Techniques for Expanding the Narrative Search Space. *New Generation Computing* 24(3):303–323.

Ritchie, G., and Masthoff, J. 2011. The STANDUP 2 interactive riddle builder. In Ventura, D.; Gervas, P.; Harrell, F.; Maher, M.; Pease, A.; and Wiggins, G., eds., *Proceedings of the Second International Conference on Computational Creativity*, 159.

Ritchie, G. 2007. Some Empirical Criteria for Attributing Creativity to a Computer Program. *Minds and Machines* 17:67–99.

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.

Sawyer, R. K. 2011. *Explaining creativity: The science of human innovation*. Oxford University Press.

Sawyer, K. 2013. *Zig Zag: the surprising path to greater creativity*. John Wiley & Sons.

Toivonen, H., and Gross, O. 2015. Data mining and machine learning in computational creativity. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5(6):265–275.

Unemi, T. 2014. Automated Daily Production of Evolutionary Audio Visual Art — An Experimental Practice. In Colton, S.; Ventura, D.; Lavrač, N.; and Cook, M., eds., *Proceedings of the Fifth International Conference on Computational Creativity*, 33–37.

Veale, T. 2013. Less Rhyme, More Reason: Knowledge–based Poetry Generation with Feeling, Insight and Wit. In Maher, M. L.; Veale, T.; Saunders, R.; and Bown, O., eds., *Proceedings of the Fourth International Conference on Computational Creativity*, 152–159.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*, 17–24. Paris, France: Sony CSL.

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