ECAI 2012
Luc De Raedt et al. (Eds.)
© 2012 The Author(s).
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License.
doi:10.3233/978-1-61499-098-7-21

Computational Creativity: The Final Frontier?

Simon Colton[1] and **Geraint A. Wiggins**[2]

Abstract. Notions relating to computational systems exhibiting creative behaviours have been explored since the very early days of computer science, and the field of Computational Creativity research has formed in the last dozen years to scientifically explore the potential of such systems. We describe this field via a working definition; a brief history of seminal work; an exploration of the main issues, technologies and ideas; and a look towards future directions. As a society, we are jealous of our creativity: creative people and their contributions to cultural progression are highly valued. Moreover, creative behaviour in people draws on a full set of intelligent abilities, so simulating such behaviour represents a serious technical challenge for Artificial Intelligence research. As such, we believe it is fair to characterise Computational Creativity as a frontier for AI research beyond all others—maybe, even, the final frontier.

1 BOLDLY ONGOING

Computational Creativity is a subfield of Artificial Intelligence (AI) research – much overlapping cognitive science and other areas – where we build and work with computational systems that create artefacts and ideas. These systems are usually, but not exclusively, applied in domains historically associated with creative people, such as mathematics and science, poetry and story telling, musical composition and performance, video game, architectural, industrial and graphic design, the visual, and even the culinary, arts. Our working definition of Computational Creativity research is:

> *The philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative.*

This definition contains two carefully considered subtleties. Firstly, the word **responsibilities** highlights the difference between the systems we build and *creativity support tools* studied in the HCI community [53] and embedded in tools such as Adobe's Photoshop, to which most observers would probably not attribute creative intent or behaviour. A creative responsibility assigned to a computational system might be: development and/or employment of aesthetic measures to assess the value of artefacts it produces; invention of novel processes for generating new material; or derivation of motivations, justifications and commentaries with which to frame their output.

Our second subtlety is in the methodological requirements for evaluation. We emphasise the involvement of **unbiased observers** in *fairly* judging the behaviours exhibited by our systems, because, it seems, there is a natural predilection for people to attribute creativity to human programmers, users and audiences instead of software and hardware. It seems that people allow their beliefs that machines can't possibly be creative to bias their judgement on such issues [32, 45].

Also related to evaluation, our working definition has two conspicuous and deliberate absences. First, it makes no mention of the value of the artefacts and ideas produced. This is because – while it is implicitly assumed that we would like our research to lead to the production of novel and valuable material – the computational systems producing that material may also innovate at aesthetic levels by inventing, justifying and utilising measures of value. Therefore, we propose to talk of the *impact* [20] of creative acts and their results, rather than the value of the output they produce, and the introduction of specific value requirements might limit the scope of future Computational Creativity research. Second, while it is popular in Computational Creativity – as it is in AI in general – to apply quasi-Turing-tests, comparing generated results with those made by people, our definition does not rule out situations where systems are deemed to be creative even though they behave in wholly different ways, and to different ends, from people. Notwithstanding the fact that many Computational Creativity researchers use simulations of human creative acts to further study humanity, we maintain that one of the real potentials of computational systems is to create in new, unforeseen modalities that would be difficult or impossible for people.

For a long period in the history of AI, creativity was not seriously considered as part of the field: indeed, when Margaret Boden included a chapter on creativity in her book, *Artificial Intelligence and Natural Man* [3], some observers suggested that it was out of place [4]. This may have been for good reason! We consider throughout this paper the difficulties that beset the study of Computational Creativity; there was a lot to be said for postponing such a difficult subfield until the larger area is better understood – as it now is. But perhaps this is also symptomatic of scepticism: perhaps creativity is, for some proponents of AI, the place that one cannot go, as intelligence is for AI's opponents. After all, creativity is one of the things that makes us human; we value it greatly, and we guard it jealously.

From the beginning of the modern computing era, notable experts have questioned the possibilities of machine intelligence with reference to creative acts. For example, the celebrated early neuroscientist Sir Geoffrey Jefferson wrote:

> "Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain"
> Geoffery Jefferson [38]

This was in response to Turing, who replied that Jefferson was merely expressing "The Argument from Consciousness" against intelligent machines, before going on to demolish it as solipsism [56]. Other AI pioneers saw the possibilities for the study and simulation of creativity with computers. Claude Shannon was among them:

[1] Reader in Computational Creativity, Computational Creativity Group, Department of Computing, Imperial College, London, UK. ccg.doc.ic.ac.uk, sgc@doc.ic.ac.uk
[2] Professor of Computational Creativity, Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary, University of London, UK. geraint.wiggins@eecs.qmul.ac.uk

"Shannon wants to feed not just data to a Brain[3], but cultural things! He wants to play music to it!" Alan Turing [36, p. 251]

In the three decades following such early interest, there were a few particularly successful attempts to build creative systems, though not always reported in AI outlets. For example, artist Harold Cohen exhibited and sold paintings created by his AARON robot [43] in traditional art circles. Kemal Ebcioğlu's CHORAL system [31] could produce chorale harmonisations that are reliably distinguishable from those of J. S. Bach only by experts (and then often because they are *too* quirkily Bach-like). But during this period, these attempts were isolated, and lacked unifying theory and methodology. The first attempt of which we are aware to provide a theoretical framework for the study of creativity in AI is that of Margaret Boden [2], which continues to have philosophical impact in Computational Creativity.

There isn't space here for a survey of the field, so we draw more on our own work than on others'; we apologise to those whose research we omit. More background on the development of Computational Creativity, especially in the main academic events, is available in the AI Magazine [8]. In the next section, we describe how research is progressing in terms of our first point above – the devolution of creative responsibility from human to machine – with implementations that create and assess artefacts of their own. It is a picture of a small but buoyant field, developing methods, methodology and real tools for creative production. Thereafter, in section 3, we focus on evaluation and the development of relevant methodology, which is particularly problematic in an area where it is often hard to say a priori what one is even trying to achieve. In the final section, we speculate on possibilities for the future of Computational Creativity research.

2 WHERE NO PROGRAM HAS GONE BEFORE

This section addresses our first major issue, the most studied in Computational Creativity research: that of how to incrementally engineer software to have more and more responsibility for the creation of artefacts and ideas. An important distinction between most AI research approaches and projects within Computational Creativity is the paradigm within which we develop and test our computational systems. It is fair to characterise much of mainstream AI practice as being within a *problem solving* paradigm: an intelligent task, that we desire to automate, is formulated as a particular type of problem to be solved. The type of reasoning/processing required to find solutions determines how the intelligent task will then be treated: if deduction is required, automated theorem proving approaches are deployed; if generalisation is required, machine learning approaches are used, etc.

It seems to us inappropriate to describe the composition of a sonata, or the painting of a picture as a *problem* to be *solved*,[4] and so techniques that essentially come down to optimisation or classification are inappropriate. Therefore, in Computational Creativity research, we prefer to work within an *artefact generation* paradigm, where the automation of an intelligent task is seen as an opportunity to produce something of cultural value. The ensuing process of giving software more creative license has been characterised as *Climbing the Meta-Mountain* [18], and echoes Bruce Buchanan's idea of 'creativity at the meta-level' [6]. Whenever possible, it is preferable to hand over responsibilities by appeal to existing AI techniques, so that wheels are not reinvented. In doing so, in addition to enabling our systems to intelligently achieve certain creative aims, our projects challenge existing AI techniques, leading us (and others) to propose improvements. For instance, the HR mathematical discovery system [13] has improved constraint solving over algebraic completion problems [9]. In addition, our projects often lead to test suites and canonical problems for AI techniques; e.g., HR has contributed to the TPTP library of problems for automated theorem provers [25].

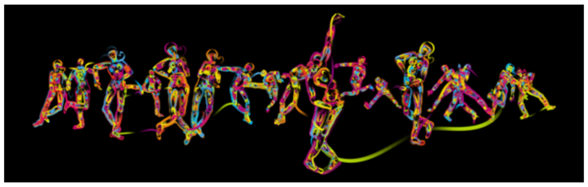
The Painting Fool project (www.thepaintingfool.com) concentrates on automating physical, but more importantly cognitive, aspects of painting, and shows how extant AI techniques can be pipelined to increase the creative responsibility of systems. The Painting Fool creates and paints scenes using: (a) constraint solving to place structural elements [17] that are themselves generated by context free grammars via the ContextFree system (www.contextfreeart.org) [24]; (b) machine learning to predict when two abstract images have too much structural similarity [19]; (c) various evolutionary methods to generate abstract art pieces [21] and image filters [26], and (d) concept formation via HR for the invention of fitness functions for scene generation [15]. An image created by the system is shown in Figure 1a [24].

Application (d) above captures the notion of handing over creative responsibility. A simple evolutionary approach was used to position and order hundreds of rectangle placeholders, producing scenes similar to that of downtown Manhattan (as seen from the Staten Island ferry) [15]. The fitness function was hand-crafted, involving positive and negative correlations between size, shape, colour and location of the rectangles. Then, in order to hand over more creative responsibility, HR was given the background concepts that constituted the hand-crafted fitness function, and asked to invent new mathematical functions involving correlations which could be themselves interpreted as fitness functions. For each of ten generated fitness functions, a highly fit cityscape scene was evolved. In one of the ten sessions, in order to maximise the fitness function, it was necessary to have more rectangles (buildings) in the centre of the scene. The scene was evolved with buildings on top of each other – which was a novel idea, as it breaks constraints from reality. While this is hardly Picasso-level imaginative thinking, it does highlight the potential for systems to "think outside the box" because of the responsibilities given to them, and to surprise their programmers/users/audiences. This kind of surprise is a *sine qua non* of Computational Creativity projects.

This example highlights an advantage of evolutionary approaches: they facilitate higher-level creative practice, because software can invent fitness functions, as above. Another advantage is the fact that they generate populations of individuals, some of which exhibit sub-optimal fitness, or achieve near-optimal fitness in unusual ways. In these evolutionary applications, there has often been more interest in individuals which are in the *second* decile of fitness rather than the top decile. This is because the less fit individuals are often more interesting in unpredictable ways than the fitter ones. Computational Creativity research involves making such observations, to determine which AI methods are most suited to creative endeavours, and why this is so. For instance, some systems such as the COLIBRI poetry generator [30] and the MuzaCazUza music generator [51], employ case-based reasoning. Such approaches have advantages, as they rely on user-given, high quality artefacts to generate new material, and the high quality of the original artefacts is often reflected in the new ones. In other systems, theories from cognitive science are implemented for generative purposes. For instance, the Divago concept generator, used in several application domains such as sound design [42], appeals to the theory of conceptual blending [33]; and the IDyOM model of musical listening has been used to generate musical melodies [47].

[3] This *Brain* is one of his theoretical machines, not its human counterpart.
[4] Although there may indeed be an artistic "problem" that the work is addressing – but that's a different, metaphorical issue.

(a)



(b)

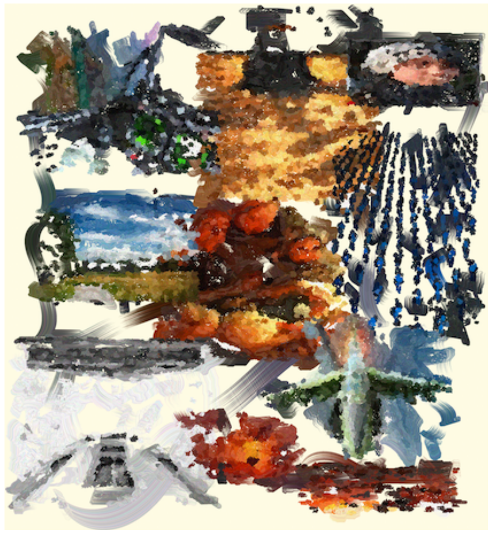


Figure 1. Two images produced by The Painting Fool software.

Application (d) also highlights the potential in Computational Creativity practice for interaction *between* generative systems to multiply their creative potential: in that application, the HR mathematical invention system and The Painting Fool computer artist were fruitfully integrated. Such collaborations are going to be more likely in future, as Computational Creativity researchers begin to make their software available through APIs and online generative systems, such as the linguistic creativity tools from the Creative Language System Group (afflatus.ucd.ie). While it is fairly rare to find two creative systems such as these combined, it is more common to see generic AI systems combined so that the whole is more than a sum of the parts [7]. For instance, in the application to discovery tasks in pure mathematics described in [14], we combined computer algebra, automated theorem proving and machine learning systems to find conjectures in number theory.

In addition to combining AI methods, it is becoming popular to produce *mashups* of fairly simple techniques, including Web 2.0 software and third party APIs, etc. For instance, as another exemplar for the idea of handing over creative responsibility, we gave The Painting Fool software the ability to create collages in response to articles from the Guardian newspaper [39]. The generative pipeline is fairly straightforward: an article is chosen; keyphrases are extracted using an implementation of the TextRank algorithm [44]; then each keyphrase is used as a search term to download images from Flickr and/or Google Images. An example collage is presented in Figure 1b. The original newspaper article was about the war in Afghanistan, and we see that the collage contains quite salient, even poignant images, including: a bomber plane, an explosion, a mother and baby, a girl in regional headgear, and – most poignant of all – a field of war graves.

A similar newspaper-based approach has been used to generate poems [22]. In contrast with the collage system, where the software did not judge its own output, the poetry generator assessed its output in terms of an aesthetic measure based on relevance to the original newspaper article, mood, lyricism and flourishes. The development of such internal measures is an important part of the building of creative systems. In most cases, the aesthetic measures developed are domain-specific – for instance, various measures for predicting the appeal of abstract art images are given in [29]. Also, there is increasing interest in learning aesthetic measures directly from user choices, for instance in an evolutionary art setting [27, 40].

Another potential of Computational Creativity is to contribute to understanding of its human equivalent. Some approaches, particularly in the musical domain, have been successful in using perceptual models, validated by comparison with human listeners, to generate new artefacts from a learned model [47]. The learning systems used are often simple, but the results produced, particularly with hybrid systems, can be surprisingly good [37]. Of particular interest in these areas is creative partnership, where a computational creative system collaborates with a person, for example, harmonising its user's melodic composition [57]. These methods can merely take inspiration from human cognition, or can attempt to simulate and thereby elucidate cognitive process. One particularly rigorous methodology is based in the requirement of machines to learn (in human-like ways or otherwise) and then generate previously unencountered artefacts by manipulation of the resulting learned models. A framework in which to place such creative models is supplied by Baars' Global Workspace Theory [1], and a long-term project has just begun to build a musical creative system [60] based on models of perception [34, 48] and information-theoretic [54] selection of partial artefacts.

3 CREATIVITY, BUT NOT AS WE KNOW IT

The second major question in Computational Creativity is how to assess progress by measuring, comparing and contrasting aspects of creative behaviour in our systems: a scientific approach is required, to identify robust indications of progress. At the highest level, there are some clear indications of success, when software is tested under terms of engagement usually reserved for people. For instance, prints of artworks by The Painting Fool have been sold [24]; theorems from the HR discovery system have appeared in the mathematical literature [11, 55]; the Ludi system [5] has invented a popular board game, for which people have paid money; and the Continuator jazz improvisation system has played alongside professional musicians [46].

Such high-level validations of Computational Creativity only happen with quite mature projects, and we need more day-to-day evaluation methods. We first have to distinguish between tests which evaluate the cultural value of the artefacts produced by systems, and tests which evaluate the sophistication of the behaviours exhibited by such systems. These, of course, can be used in conjunction to produce overall assessments of progress. Looking first at product-based assessments, comparing the artefacts created by software with those created by people is a common way of assessing progress. This is because, in many application domains, it is a significant milestone when observers cannot reliably distinguish between a computer generated artefact and one produced by a person. Often these tests are performed in a creator-blind fashion, to minimise any bias[5] against computer-generated artefacts, which has been observed [32, 45].

However, there are drawbacks with such blind comparison tests. Firstly, they ask what may be the wrong question: "If you have no idea how these artefacts were produced, which do you like the most?" Such answers may be valuable during the building of creative systems, but it is not adequate to limit evaluation to blind tests in or-

[5] But there is anecdotal evidence that the computer generated board games from Ludi [5] are admired *more* because of their computational origin.

der to avoid bias. Some researchers (ourselves included) advocate instead presenting computer generated artefacts to audiences with full knowledge of their computational origin, being loud and proud about the AI processes leading to their generation, and hoping to reduce the bias against software over time through exposure to our projects. Further, process-blind comparison tests can attract reward to software acting in naïve ways, and encourage concentrating efforts on the production of pastiches in particular styles [50]. Accusations of naïvety and pastiche are severe criticisms in creative circles.

Another reason to avoid imitation games is that they are harmful in certain cultural circles: an art dealer undertaking such comparisons would be putting herself in a no-win scenario. First, she would be implicitly endorsing the idea that visual arts is entirely skin-deep, when dealers use the personality, personal history, ideas, political interests and humanity of the artists they represent as selling points, and describe innovation at process level to promote their artists – process is of at least equal importance to product in modern art evaluation. Second, if she correctly identify the computer-generated pieces, this is no surprise, but if she fails, her credibility suffers. As a final point, comparing computational systems with people can set the machines up for a fall, and give the wrong impression that all we aim for is human-level creativity. In reality, the potential for computational systems to create via interesting, but non-human ways is an interesting driving force for many researchers [16, 18].

When assessing progress in terms of the output of creative software, there are some well-developed formalisms that we can appeal to. In particular, Boden laid some groundwork by describing artefacts as *P-creative*, i.e., new to the system/person that generated them and *H-creative*, i.e., historically new to everyone [2]. One approach extending this is to identify features of creative systems themselves that can be localised and compared, within an overarching theory. The first one such, the Creative Systems Framework [58, 59] is based on Boden's approach, and contributes to that philosophy by showing how it can be simplified. A substantial contribution to assessing creative software in terms of its output has been given by Ritchie [52]. In this influential formalism, key concepts of the novelty, typicality and quality of the artefacts are employed to provide concrete measures by which the value of a system can be estimated from the value of its output. Evaluation using Ritchie's measures has been performed for the WASP poetry generation system [35], and others.

As a bridging point for assessing the behaviour of software via process rather than product, we might initially look at the *curation coefficient* associated with particular outputs. That is, in many projects, the output is carefully scrutinised by the program's author, and only the best examples are shown to audiences, or used as exemplars in research papers, etc. For instance, the author of a poetry generation system might conclude that only one in a thousand poems are up to a sufficient standard to be presented publicly. If an anthology of such painstakingly chosen good poems was published, we (and the public) might be critical if too strong a claim of computer creativity was made, because it is easy to see that a large creative responsibility was held by the curator, i.e., the programmer choosing the best output. In this case, the software might be seen at best, as a muse, rather than a creative collaborator or poet in its own right.

A poet with no critical ability to judge its own work (hence requiring supervisory intervention) is no poet at all. Generalising from such observations, we argue in [16] that people largely criticise software for being uncreative along three axes, namely lack of skill, lack of appreciation and lack of imagination. We hypothesise that a *creativity tripod* of skilful, appreciative and imaginative behaviours are the bare minimum required to support the perception of creativity in computational systems. The generative poet mentioned above has no appreciation of what it is doing or what it is has produced. Moreover, imagine if the generative process only involved changing one word randomly in a carefully crafted template. In this case, it would be sensible to question whether to call the poems 'computer-generated' at all, because so little skill was used in their production. Finally, by not innovating at any level (e.g., changing the template, breaking or inventing constraints, etc.) it is easy to criticise the automated poet for not exhibiting any imaginative behaviours. Hence, while the software may be generative in the sense that it produces novel output, it is very difficult to project the word 'creativity' onto it.

This uncreative automated poetry generator raises an important issue in balancing the assessment of creative software with respect to product and process. It is highly likely that if the programmer stepped back from the curation process (i.e., let the software choose its best poems) then the value of the poems, as assessed by independent observers, would decrease, even though the same observers might say that the software is slightly more creative. We call this phenomenon the *latent heat effect* in Computational Creativity: as the creative responsibility given to systems increases, the value of its output does not (initially) increase, much as heat input to a substance on the boundary of state change does not increase temperature. Hopefully the value of the output may increase later as even more sophisticated behaviours are added, and in some cases, as argued below, the fact that the software itself is being more creative might be reflected in an increased perception of the value of the artefacts it produces.

Part of the problem leading to the latent heat effect is that, on the surface, the generative task doesn't change as systems are built. If we perceive the software as solving the problem of generating high-value poems, then successive versions get worse at this. However, in reality, by handing over increasing amounts of creative responsibility, the software is in effect solving sets of more difficult problems within the artefact generation paradigm. In such situations, where software is being engineered to take on more creative responsibility, any measure of progress based entirely on the output of the software would fail to correctly reward the advance in intelligence of the software. To combat this, evaluation schemes could take into account the input to a system in addition to the output, and value more highly artefacts which were produced from less extensive input, as discussed in [23] and covered in Ritchie's evaluation criteria [52].

Alternatively, evaluation schemes could take into account the level of creative responsibility given to software, and the level of sophistication of the processing required to exercise those responsibilities. We have tried to capture aspects of this way of evaluating software with the FACE descriptive model of the creative acts that software can undertake [20, 49]. In particular, we argue that software can (at least) undertake generative acts producing: (F)raming information, as described below, and (A)esthetic considerations which can be used to assess (C)oncepts and (E)xamples of those concepts. Moreover, we point out that – in principle, but rarely[6] in practice at the moment – software can also innovate at the process level, by inventing new methods for generative acts producing F, A, C or E outputs. The FACE formalism describes software in terms of the *creative acts* it undertakes, where a creative act is a tuple of generative acts.

There is another reason why assessing the processing of computational systems is important: in some application domains, software must *be seen to be AI*, rather than just producing good results by any means. In domains like the visual arts, information about *how* an artefact has been produced is often used when judging the value

[6] See [12] for an example of such process-level creative acts.

of that artefact [10, 16, 20]. It might therefore be advantageous for creative systems to explain the choices they make while generating artefacts, and give some details of their generative acts. Explaining the process of production is part of the more general notion of people and software *framing* their creative acts with information that adds value, possibly via reference to political, historical, or cultural contexts, in addition to information about process, motivations and inspirations [10]. We would argue that, even when nothing is known about a particular creator, if we know that they were human, we can refer to our own experiences in addition to default (and often Romantic) production and lifestyle scenarios to add value to the work. Unfortunately, the opposite can often be true when we know nothing about how software produced an artefact: we can project cold, heartless, simplistic (often just random) processing onto the software, and possibly use this to devalue the artefact. Hence, we advocate that the software goes further [10], and writes commentaries and stories about its production process (as in [22]), which may involve elements of fiction. Ultimately, while eschewing the imitation game aspect of Turing-style tests, for reasons given above, we would promote the dialogue aspect of such tests, i.e., the software should be available for questioning about its motivations, processes and products.

Issues of evaluation both arise from implementation projects, and drive such projects forward. We mentioned above our poetry project [22], where the software appealed to aesthetic measures to choose between the poems it produced. In fact, the aesthetic measure was itself invented by the software, and this behaviour was part of a larger implementation effort where we used the FACE evaluation model [20] to guide the building of the system. That is, starting with a straightforward template-filling program endowed through NLP resources to generate high-value poems (thus exhibiting generative acts of type E), we gave it the ability to produce its own templates (C-type generative acts), invent an aesthetic measure relative to a newspaper article and the mood of the day (A-type generative acts), and finally to generate a commentary on how and why it had turned the article into a poem (F-type generative acts). At the presentation of this work, we pointed out that the project represented a step forward for Computational Creativity research, but, due to the latent heat effect, the project represented a step backwards for automated poetry generation, as the more sophisticated version produced what would be seen as worse poetry than a simpler version which was based on template-filling. Moreover, this, and the collage generation project mentioned above, blurred the lines between intentionality expressed by different agents, including the software [28, 39].

4 LIVING LONG AND PROSPERING

We have provided a working definition of the field of Computational Creativity, and seen how it has been *boldly ongoing* since the earliest days of the modern computing era. It has blossomed in the last dozen years into an exciting subfield of AI, vying for central ground. One of our two main themes is how to hand over creative responsibility to systems, and, in doing so, how to take software *where no program has gone before*. Computational systems are not human, and so the creativity they exhibit will be *creativity, but not as we know it*: never exactly the same as in humans. We have taken this into account in the evaluation models outlined above, and argued that the time is right for the differences between human and computer creativity to be celebrated, enabling our field to *live long and prosper*.

The 2012 International Conference on Computational Creativity (computationalcreativity.net) included papers covering a healthy range of domains including the visual arts, video games, music, poetry, punning riddles, narratives, hypothesis discovery, and a new domain: cooking. There were more than 20 creative systems demonstrated and discussed, from Computational Creativity groups around the world, with various papers exploring techniques for use in generative software, such as blending, analogy making, global workspaces architectures and affective computing. There were also a healthy number of papers entirely devoted to questions of evaluating creative systems in general, covering issues such as search, creative trajectories, group influence, creative personae and meta-creation. The posters for the conference included the byline: "Scoffing at mere generation for more than a decade", penned by the local chair, Tony Veale. While intended for cheeky and humourous effect, this catchphrase highlights the progress in the field over the last dozen or so years: it is not easy to have a paper accepted at the premiere event if the research only covers how to generate a certain type of artefact. To be a contribution to the field, papers need to describe how software can evaluate aesthetic and utilitarian properties of their output, and possibly to highlight some higher level issues related to the field.

Our title in this conference suggests that we view our field as a potential final frontier for AI, and this is indeed so. Creativity requires all faculties of intelligence exhibited simultaneously, and society has a natural protection towards creativity, a most human of qualities. It is not going to be easy to engineer truly creative systems, and may be even harder to get society at large to accept them as creative individuals and collaborators. Moreover, aesthetics relates, via emotion, to consciousness, and consciousness, in our opinion, is not territory for research in simulation – for consciousness, where genuinely extant, is simply consciousness, being real and not simulated. Computational creativity, therefore, lies between other sub-fields of AI and consciousness, on the boundary between AI and beyond.

There are several directions in which Computational Creativity research might go, including (i) continued integration of systems to increase their creative potential (ii) usage of web resources as source material and conceptual inspiration for creative acts by computer (iii) using crowd sourcing and collaborative creative technologies [41] (iv) bringing together evaluation methodologies based on product, process, intentionality and the framing of creative acts by software. We propose that at least the following maxims should be at the heart of Computational Creativity projects in the future:

- When we celebrate an artefact such as a musical composition, a painting, a theorem or a poem, we are also celebrating the creative act which brought it into being.
- The artefact resulting from a creative act should be seen as an invitation to engage in a dialogue with the artefact and/or the creator and/or the culture and/or yourself.
- Software is not human, so we cannot rely on unreasoned (often Romantic) ideas about the creative process in people. So, our software needs to work hard to frame its processes and products.

Currently, having a bespoke painting, poem or piece of music created is the privilege of the few. However, one day, *the needs of the many will outweigh the needs of the few*, and we will expect the Internet to provide new ideas and new artefacts on demand, just like we expect it right now to provide old ideas and old artefacts. We will go online for: a new, relevant, joke for a speech; an exciting new recipe for a party; or a bespoke and beautiful new painting for a present. We cannot expect the world's creative people alone to supply artefacts for such a huge demand, so autonomously creative software will be necessary. The research undertaken in Computational Creativity projects – to help break the *final frontier* in AI research – will be pivotal in bringing about this technological and cultural revolution.

ACKNOWLEDGEMENTS

We would like to thank the researchers whose work is cited here, and apologise to those whose work we could not cite, due to space restrictions. Particular thanks are offered to the members of the Computational Creativity Conference Steering Committee, who have helped greatly to guide the field over the last decade. The authors are funded by EPSRC grants EP/J004049/1, *Computational creativity theory*, and EP/H01294X/2, *Information and neural dynamics in the perception of musical structure*, respectively.

REFERENCES

[1] B J Baars, *A cognitive theory of consciousness*, CUP, 1988.
[2] M Boden, *The Creative Mind: Myths & Mechanisms*, Routledge, 2003.
[3] M Boden, *Artificial Intelligence & Natural Man*, Harvester Press, 1977.
[4] M Boden, 'Preface to special issue on creativity in the arts and sciences', *AISB Quarterly*, 102, (1999).
[5] C Browne, *Evolutionary Game Design*, Springer, 2011.
[6] B Buchanan, 'Creativity at the meta-level', *AI Magazine*, 22(3), (2001).
[7] A Bundy, 'Cooperating reasoning processes: More than just the sum of their parts', in *Proc. of the 20th IJCAI*, (2007).
[8] A Cardoso, T Veale, and G A Wiggins, 'Converging on the divergent: The history (and future) of the International Joint Workshops in Computational Creativity', *AI Magazine*, 30(3), (2010).
[9] J Charnley, S Colton, and I Miguel, 'Automatic generation of implied constraints', in *Proc. of the 17th ECAI*, (2006).
[10] J Charnley, A Pease, and S Colton, 'On the notion of framing in Computational Creativity', in *Proc. of the 3rd International Conference on Computational Creativity*, (2012).
[11] S Colton, 'Refactorable numbers – a machine invention', *Journal of Integer Sequences*, 2, (1999).
[12] S Colton, 'Experiments in meta-theory formation', in *Proc. of the AISB'01 Symposium on AI and Creativity in Arts and Science*, (2001).
[13] S Colton, *Automated Theory Formation in Pure Maths*, Springer, 2002.
[14] S Colton, 'Automated conjecture making in number theory using HR, Otter and Maple', *Journal of Symbolic Computation*, 39(5), (2004).
[15] S Colton, 'Automatic invention of fitness functions, with application to scene generation', in *Proc. of the EvoMusArt Workshop*, (2008).
[16] S Colton, 'Creativity versus the perception of creativity in computational systems', in *Proc. of the AAAI Spring Symposium on Creative Systems*, (2008).
[17] S Colton, 'Experiments in constraint-based automated scene generation', in *Proc. 5th Int. Joint Workshop on Comp. Creativity*, (2008).
[18] S Colton, 'Seven catchy phrases for computational creativity research', in *Proc. of the Dagstuhl Seminar: Computational Creativity: An Interdisciplinary Approach*, (2009).
[19] S Colton, 'Evolving a library of scene descriptors', in *Proc. of the EvoMusArt conference*, (2012).
[20] S Colton, J Charnley, and A Pease, 'Computational Creativity Theory: the FACE and IDEA models', in *Proc. of the 2nd International Conference on Computational Creativity*, (2011).
[21] S Colton, M Cook, and A Raad, 'Ludic considerations of tablet-based Evo-art', in *Proc. of the EvoMusArt workshop*, (2011).
[22] S Colton, J Goodwin, and T Veale, 'Full-face poetry generation', in *Proc. of the 3rd Int. Conference on Computational Creativity*, (2012).
[23] S Colton, A Pease, and G Ritchie, 'The effect of input knowledge on creativity', in *Proc. ICCBR'01 Workshop on Creative Systems*, (2001).
[24] S Colton and B Pérez Ferrer, 'No photos harmed/growing paths from seed – an exhibition', in *Proc. of the Non-Photorealistic Animation and Rendering Symposium*, (2012).
[25] S Colton and G Sutcliffe, 'Automatic generation of benchmark problems for automated theorem proving systems', in *Proc. of the 7th AI and Maths Symposium*, (2002).
[26] S Colton and P Torres, 'Evolving approximate image filters', in *Proc. of the EvoMusArt Workshop*, (2009).
[27] S Colton, P Torres, J Gow, and P Cairns, 'Experiments in Objet Trouvé browsing', in *Proc. of the 1st Int. Conf. on Comp. Creativity*, (2010).
[28] M Cook and S Colton, 'Automated collage generation - with more intent', in *Proc. of the 2nd Int. Conf. on Computational Creativity*, (2011).
[29] E den Heijer and A Eiben, 'Comparing aesthetic measures for evolutionary art', in *Proc. of the EvoMusArt workshop*, (2010).
[30] B Díaz-Agudo, P Gervás, and P González-Calero, 'Poetry generation in COLIBRI', *Advances in Case-Based Reasoning*, 2416, (2002).
[31] K. Ebcioğlu, 'An expert system for harmonizing chorales in the style of J. S. Bach', *Journal of Logic Programming*, 8, (1990).
[32] A Eigenfeldt, A Burnett, and P Pasquier, 'Evaluating musical metacreation in a live performance context', in *Proc. of the 3rd International Conference on Computational Creativity*, (2012).
[33] G Fauconnier and M Turner, *The Way We Think: Conceptual Blending And The Mind's Hidden Complexities*, Basic Books, 2002.
[34] P Gärdenfors, *Conceptual Spaces: the geometry of thought*, MIT Press, 2000.
[35] P Gervás, 'Exploring quantitative evaluations of the creativity of automatic poets', in *Proc. of the 2nd workshop on creative systems, approaches to creativity in AI and Cognitive Science (ECAI)*, (2002).
[36] A Hodges, *Alan Turing: The Enigma*, Vintage, London, 1992.
[37] D Hörnel, 'A multi-scale neural-network model for learning and reproducing chorale variations', in *Computing in Musicology*, 11, (1998).
[38] G Jefferson, 'The mind of mechanical man', *British Medical Journal*, 1(4616), (1949).
[39] A Krzeczkowska, J El-Hage, S Colton, and S Clark, 'Automated collage generation - with intent', in *Proc. of the 1st International Conference on Computational Creativity*, (2010).
[40] Y Li and C Hu, 'Aesthetic learning in an interactive evolutionary art system', in *Proc. of the EvoMusArt workshop*, (2010).
[41] M Maher, 'Computational and collective creativity: Who's being creative?', in *Proc. 3rd Int. Conf. on Computational Creativity*, (2012).
[42] J Martins, F Pereira, E Miranda, and A Cardoso, 'Enhancing sound design with conceptual blending of sound descriptors', in *Proc. of the Workshop on Computational Creativity*, (2004).
[43] P McCorduck, *AARON'S CODE: Meta-Art, Artificial Intelligence and the Work of Harold Cohen*, Freeman, 1991.
[44] R Mihalcea and P Tarau, 'Textrank: Bringing order into texts', in *Proc. of the Conference on Empirical Methods in NLP*, (2004).
[45] D Moffat and M Kelly, 'An investigation into people's bias against computational creativity in music composition', in *Proceedings of the International Joint Workshop on Computational Creativity*, (2006).
[46] F Pachet, 'The Continuator: Musical interaction with style', *Journal of New Music Research*, 32(3), (2003).
[47] M T Pearce and G A Wiggins, 'Evaluating cognitive models of musical composition', in *Proc. of the 4th International Joint Workshop on Computational Creativity*, (2007).
[48] M T Pearce, *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*, Ph.D. dissertation, Department of Computing, City University, London, UK, 2005.
[49] A Pease and S Colton, 'Computational Creativity Theory: Inspirations behind the FACE and IDEA models', in *Proc. of the 2nd International Conference on Computational Creativity*, (2011).
[50] A Pease and S Colton, 'On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal', in *Proc. of the AISB symposium on AI and Philosophy*, (2011).
[51] P Ribeiro, F Pereira, M Ferr, and A Cardoso, 'Case-based melody generation with MuzaCazUza', in *Proc. of the AISB'01 Symposium on AI and Creativity in Arts and Science*, (2001).
[52] G Ritchie, 'Some empirical criteria for attributing creativity to a computer program', *Minds and Machines*, 17, (2007).
[53] B Schneiderman, 'Creativity support tools: Accelerating discovery and innovation', *Communications of the ACM*, (2007).
[54] C E Shannon, 'A mathematical theory of communication', *Bell System Technical Journal*, 27, (1948).
[55] V Sorge, S Colton, R McCasland, and A Meier, 'Classification results in quasigroup and loop theory via a combination of automated reasoning tools', *Comment.Math.Univ.Carolin*, 49(2), (2008).
[56] A Turing, 'Computing machinery and intelligence', *Mind*, 59, (1950).
[57] R Whorley, G A Wiggins, C Rhodes, and M T Pearce, 'Development of techniques for the computational modelling of harmony', in *Proc. of the First International Conference on Computational Creativity*, (2010).
[58] G A Wiggins, 'A preliminary framework for description, analysis and comparison of creative systems', *Journal of Knowledge Based Systems*, 19(7), (2006).
[59] G A Wiggins, 'Searching for Computational Creativity', *New Generation Computing*, 24(3), (2006).
[60] G A Wiggins, 'The mind's chorus: Creativity before consciousness', *Cognitive Computation*, Special issue on Computational Creativity, Intelligence and Autonomy, (2012).