On impact and evaluation in Computational Creativity: A discussion of the Turing Test and an alternative proposal

Alison Pease[1] and **Simon Colton**[2]

Abstract. Computational Creativity is the AI subfield in which we study how to build computational models of creative thought in science and the arts. From an engineering perspective, it is desirable to have concrete measures for assessing the progress made from one version of a program to another, or for comparing and contrasting different software systems for the same creative task. We describe the Turing Test and versions of it which have been used in order to measure progress in Computational Creativity. We show that the versions proposed thus far lack the important aspect of interaction, without which much of the power of the Turing Test is lost. We argue that the Turing Test is largely inappropriate for the purposes of evaluation in Computational Creativity, since it attempts to homogenise creativity into a single (human) style, does not take into account the importance of background and contextual information for a creative act, encourages superficial, uninteresting advances in front-ends, and rewards creativity which adheres to a certain style over that which creates something which is genuinely novel. We further argue that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable to apply any defensible version of the Turing Test.

As an alternative to Turing-style tests, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process. While these models require further study and elaboration, we believe that they can be usefully applied to current systems as well as guiding further development of creative systems.

1 The Turing Test and Computational Creativity

The Turing Test (TT), in which a computer and human are interrogated, with the computer considered intelligent if the human interrogator is unable to distinguish between them, is principally a philosophical construct proposed by Alan Turing as a way of determining whether AI has achieved its goal of simulating intelligence [1]. The TT has provoked much discussion, both historical and contemporary, however this has principally been within the philosophy of AI: most AI researchers see it as a distraction from their goals, encouraging a mere trickery of intelligence and ever more sophisticated natural language front ends, as opposed to focussing on real problems. Despite the appeal of the (as yet unawarded) Loebner Prize, most subfields of AI have developed and follow their own evaluation criteria and methodologies, which have little to do with the TT.

[1] School of Informatics, University of Edinburgh, UK
[2] Department of Computing, Imperial College, London, UK

Computational Creativity (CC) is a subfield of AI, in which researchers aim to model creative thought by building programs which can produce ideas and artefacts which are novel, surprising and valuable, either autonomously or in conjunction with humans. There are three main motivations for the study of Computational Creativity:

- to provide a computational perspective on human creativity, in order to help us to understand it (cognitive science);
- to enable machines to be creative, in order to enhance our lives in some way (engineering); and
- to produce tools which enhance human creativity (aids for creative individuals).

Creativity can be subdivided into everyday problem-solving, and the sort of creativity reserved for the truly great, in which a problem is solved or an object created that has a major impact on other people. These are respectively known as "little-c" (mundane) and "big-C" (eminent) creativity [2]. Boden [3] draws a similar distinction in her view of creativity as search within a conceptual space, where "exploratory creativity" searches within the space, and "transformational creativity" involves expanding the space by breaking one or more of the defining characteristics and creating a new conceptual space. Boden sees transformational creativity as more surprising, since, according to the defining rules of the conceptual space, ideas within this space could not have been found before.

There are two notions of evaluation in CC: (*i*) judgements which determine whether an idea or artefact is valuable or not (an essential criterion for creativity) – these judgements may be made internally by whoever produced the idea, or externally, by someone else and (*ii*) judgements to determine whether a system is acting creatively or not. In the following discussion, by evaluation, we mean the latter judgement. Finding measures of evaluation of CC is an active area of research, both influenced by, and influencing, practical and theoretical aspects of CC. It is a particularly important area, since such measures suggest ways of defining progress in the field,[3] as well as strongly guiding program design. While tests of creativity in humans are important for our understanding of creativity, they do not usually *cause* humans to be creative (creativity training programs, which train people to do well at such tests, notwithstanding). Ways in which CC is evaluated, on the other hand, will have a deep influence on future development of potentially creative programs. Clearly, different modes of evaluation will be appropriate for the different motivations listed above.

[3] The necessity for good measures of evaluation in CC is somewhat paralleled in the psychology of creativity: "Creativity is becoming a popular topic in educational, economic and political circles throughout the world – whether this popularity is just a passing fad or a lasting change in interest in creativity and innovation will probably depend, in large part, on whether creativity assessment keeps pace with the rest of the field." [4, p. 64]

The Turing Test is of particular interest to CC for two reasons. Firstly, unlike the general situation in AI, the TT, or variations of it, *are* currently being used to evaluate candidate programs in CC. Thus, the TT is having a major influence on the development of CC. This influence is usually neither noted nor questioned. Secondly, there are huge philosophical problems with using a test based on imitation to evaluate competence in an area of thought which is based on originality. While there are varying definitions of creativity, the majority consider some interpretation of novelty and utility to be essential criteria. For instance, one of the commonalities found by Rothenberg in a collection of international perspectives on creativity is that "creativity involves thinking that is aimed at producing ideas or products that are relatively novel" [5, p.2], and in CC the combination of novelty and usefulness is accepted as key (for instance, see [6] or [3]). In [4], Plucker and Makel list "similar, overlapping and possibly synonymous terms for creativity: imagination, ingenuity, innovation, inspiration, inventiveness, muse, novelty, originality, serendipity, talent and unique". The term 'imitation' is simply antipodal to many of these terms.

In the following sections, we firstly describe and discuss some attempts to evaluate Computational Creativity using the Turing Test or versions of it (§2), concluding that these attempts all omit the important aspect of interaction, and suggest the sort of direction that a TT for a creative computer art system might follow. We then present a series of arguments that the TT is inappropriate for measuring creativity in computers (or humans) in §3, and suggest that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable and impractical. As an alternative to Turing-style tests, in §4, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process. We conclude our discussion in §5.

2 Attempts to evaluate Computational Creativity using the Turing Test or versions of it

There have been several attempts to evaluate Computational Creativity using the Turing Test or versions of it. While these are useful in terms of advancing our understanding of CC, they do not go far enough. In this section we discuss two such advances (§2.1 and §2.2), and two further suggestions on using human creative behaviour as a guide for evaluating Computational Creativity (§2.3). We highlight the importance of interaction in §2.4.

2.1 Discrimination tests

Pearce and Wiggins [7] assert for the need for objective, falsifiable measures of evaluation in cognitive musicology. They propose the 'discrimination test', which is analogous to the TT, in which subjects are played segments of both machine and human-generated music and asked to distinguish between them. This might be in a particular style, such as Bach's music, or might be more general. They also present one of the most considered analyses of whether Turing-style tests such as the framework they propose might be appropriate for evaluating Computational Creativity [7, §7]. While they do not directly refer to Boden's exploratory creativity [3], instead referring to Boden's distinction between psychological (P-creativity, concerning ideas which are novel with resepct to a particular mind) and historical creativity (H-creativity, concerning ideas which are novel with respect to the whole of human history[4]), they do argue that much creative work is carried out within a particular style. They cite Garnham's response [8] to Boden's ideas, in which he emphasizes the importance of exploratory as compared to transformational creativity: "the origins of the symphony are lost in history and its major triumphs are the work of composers who did not invent the basic symphonic form." (Bundy argues along similar lines in [9]). Thus, Pearce and Wiggins suggest that their test rewards an appropriate level of novelty, since they found in their experiments that subjects could identify machine-generated compositions which were either too strange (too far away from well-explored areas) or too predictable (conforming too much to the well-explored areas). In anticipation of the objection that the process by which something has been created is important to judgements of creativity and thus a behaviour-based test is insufficient, Pearce and Wiggins refer to Hofstadter's argument that interaction with a system at an arbitrarily deep level can shed great insight into the processes it uses to generate its output [10]. While seeing the evaluation of the creativity of machine composers as an extension of their framework rather than a fully developed aspect, Pearce and Wiggins suggest that this type of evaluation is relevant for musical creativity within a specific style (that is, exploratory creativity). They also suggest that it may generalise to other creative domains such as art or story generation.

2.2 A Turing Test for artistic creativity

In [11], Boden discusses the Turing Test and artistic creativity. She provides an interpretation of the Turing Test which is specifically designed for computer art systems:

> "I will take it that for an 'artistic' program to pass the TT would be for it to produce artwork which was:
>
> 1. indistinguishable from one produced by a human being; and/or
> 2. was seen as having as much aesthetic value as one produced by a human being." [11, p. 409]

Boden describes several systems which produce art or music, which she considers to be either non-interactive or unpredictably interactive (such as a piece of art which responds to audience members or participants in ways they do not understand). She discusses comparisons with both mediocre human art, in this case pastiches of given styles (perhaps comparable to work by an art student exploring a given style), as well as examples which match world class human art, of interest as an artwork in itself (comparable to work done by a practising artist). She argues that the following systems all pass (her version of) the TT:

- Richard Brown's Starfish[5] – a computer generated starfish which appeared to be trapped inside a glass table, which interacted with audience members by responding to their movements and sounds. This featured in the Millennium Dome;
- AARON, a software program written by the artist Harold Cohen that creates original artistic images which are exhibited in art galleries around the world (described by McCorduck in [12]);

[4] Note that these two types of creativity are *not* analogous to the little-c/big-C distinction, since Boden talks of P-creativity being a subset of H-creativity [3, pp. 32-33].

[5] For further details, see http://www.mimetics.com/vur/mindzone.html.

- Computer art by Boden and Edmunds [13] which was exhibited in honour of world famous artists. This was composed of vertical stripes of colour which were continually changing, where the colours were partially determined by audience participation in an unpredictable manner, with constraints on certain colour combinations;
- Cope's system Emmy (Experiments in Musical Intelligence) [14, 15] which generated music in particular styles, such as that of Mozart, which was indistinguishable from human-composed Mozart pastiches, and was performed in concert halls.

Boden argues that these systems satisfy the second criterion: their aesthetic value has been proven by the degree of interest in their work (presumably, from members of the public, artists and musicians, rather than solely AI researchers). These all model exploratory creativity, where a style is explored. For examples of transformational creativity, Boden refers to systems by Todd and Latham [16] and Sims [17]. However, since these are much more interactive, she does not (yet) consider them to be candidates for the TT. Regarding the first criterion, Boden mentions anecdotally some occasions on which critics have admired a piece of art and then retracted the view when the art was discovered to be machine-generated. This suggests that, in some cases at least, systems have satisfied her first criterion.

We have a number of objections to Boden's usage of the term 'Turing Test' for the above evaluation criteria. Firstly, Boden reinterprets the TT and presents her own version, which differs substantially from Turing's proposal in at least two ways: (*i*) there is no interaction with the system, and (*ii*) by using a disjunctive rather than conjunctive relationship between the two criteria, she allows that all systems which produce output with "as much aesthetic value as produced by a human being" passes the TT. Systems which produce output of sufficient interest to be exhibited are therefore evaluated to have passed the TT. In particular, Boden argues that "If being exhibited alongside Rothko, in a 'diamond jubilee' celebration of these famous artists, does not count as passing the Turing Test, then I do not know what would." [11, p. 410]. This lack of emphasis either on interaction, or on discrimination between human and computer-produced artefacts seems to be rather missing the point of the TT. In particular, Boden seems to have expanded the term 'Turing Test' from being just one way of testing that intelligence might have been exhibited, to being a way of testing whether software has done something (or produced something) culturally significant. Our second objection is that the evidence for the second criterion, which is closest to the TT, is never explicitly addressed, and only implicitly in an anecdotal fashion. In fact, we see Boden's argument as supporting the idea that computer-created art may very well be distinguishable from human-created art, yet still have great aesthetic and cultural value, (see §3.1 for further argument on this point); that is, that the TT is inappropriate in this context. Clearly, art generation software could fail the originally conceived Turing Test, yet pass Boden's version of it.

Despite our objections to using a misleading naming based on the Turing Test, Boden's criteria can certainly be valuable for evaluating creative systems. However, we would caution that software which exhibits very little behaviour that would normally be considered (in computing or human circles) as creative can be evaluated positively using Boden's criteria. In particular, Brown's Starfish project, while a beautiful demonstration of neural net technology, and an exciting piece of human-computer interaction, certainly cannot be described as an example of software acting creatively. It is an example of kinetic art which was conceived, designed, produced, programmed and evaluated by humans (Richard Brown, Jonathan Mackenzie and Gavin Baily). While the software is generative, and to some extent unpredictable, it exhibits no higher level cognitive functioning such as the generation and/or application of aesthetic considerations or any behaviour which might be deemed remotely imaginative.

While Boden's criteria for the assessment of art-generating software are valid, we argue that calling it a Turing Test confuses the assessment of intelligence and creativity with the assessment of cultural impact, and that software which wouldn't ordinarily be considered creative can pass the test, hence the criteria have limited value for the assessment of software developed in a Computational Creativity context.

2.3 Using human creative behaviour as a guide for evaluating Computational Creativity

Wiggins proposes the following working definition of Computational Creativity:

> "The performance of tasks [by a computer] which, if performed by a human, would be deemed creative." [18, p. 451]

This type of behavioural test, in which output from a computer is compared to that from humans, has much in common with the Turing Test. In addition, Colton [19] has argued that creativity in software is often marked negatively, i.e., while there may be no obvious set of behaviours that software must exhibit in order to be regarded as creative, there are some common ways in which software can be immediately disregarded as being uncreative. In particular, Colton proposes that the criticisms levelled at software can largely be grouped into three categories: the software doesn't exhibit enough (or the right kind of) *skill*; the software has no *appreciation* of what it is doing, what it produces or what other people/machines do; the software exhibits no *imagination* in its processing. Hence, he suggests that Computational Creativity researchers should aim to build software which exhibits behaviour that might be deemed as skilful, appreciative and imaginative.

2.4 The importance of interaction

All of the versions of the TT which we have discussed here have one obvious similarity; there is no interaction with the program. This leaves out what is, arguably, the main strength of the TT. We have already introduced Hofstadter's argument that interaction with a system at an arbitrarily deep level can shed great insight into the processes it uses to generate its output (see §2.1). Hofstadter goes on to say:

> "In the spirit of much of the best science of our century, the Turing Test blurs the supposedly sharp line between probing of behavior and probing of mechanisms, as well as the supposedly sharp line between "direct" and "indirect" observation, and thus reminds us of the artificiality of such distinctions. Any computer model of mind that passes a truly deep Turing Test - one that probes for the fundamental mechanisms of thought will agree with "brain structures" all the way down to the level where the essence of thinking really takes place." [10, pp. 490-491]

The key word here is 'probe': interaction must form a necessary part of any test based on the TT, for it to hold any relevance to CC. For example, a Turing Test for artistic creativity which consisted of requests to draw something specific might be informative. A human

interrogator might attempt to distinguish between a computer art system and a human artist by making requests, such as:

- Draw something in the style of Picasso.
- Can you break/change/enhance the rules of the Impressionist style and draw something within the new style you've just created?
- Draw something which reflects your feelings towards the war in Afghanistan.
- Draw something warm.
- Show me your best painting and explain to me why you think it's good.
- Who or what has influenced your work?
- How does your work fit into the wider artistic community?

In order to avoid pitfalls of the current TT and focus on the important issues, the test could be conducted without the need for natural language,[6] timing issues, and so on.

3 Arguments that the Turing Test is inappropriate for measuring creativity in computers (or humans)

In this section, we argue that the Turing Test is largely inappropriate in the context of CC. Attempts to pass the Turing Test may result in losing differing, and valuable, styles of creativity (§3.1); might fail to take into account the importance of background and contextual information for a creative act (§3.2); encourage superficial, uninteresting advances in front-ends (§3.3); and result in rewarding creativity which adheres to a certain style over that which creates something which is genuinely novel (§3.4). We suggest that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently impractical (§3.5).

3.1 The Turing Test penalises different styles of creativity

Creativity is a cultural notion, and people around the world understand, study and assess human creativity in many different ways, as detailed in [20]. There are also many different categories of creative humans: for instance, people with cognitive disorders such as autism, people with mental health problems, different nationalities and tribes, different genders, and what mathematician Alexander Borovik calls "that forgotten tribe of humanity, children".[7] We can often distinguish creative work performed by one of these groups; developmental psychologists can determine approximate age of a creator during childhood, people can often determine gender or nationality of an author, and so on. We do not discriminate against any of these categories purely because they are identifiable, rather we relish their differences. A writer with autism might tend to write more literally than one without, who might employ devices such as metaphor and imagery in their work. An artist with synaesthesia who can taste colour may well use colour differently to an asynaesthete. A poet under the influence of drugs might have different sorts of insights than when they were sober. A Chinese percussionist will compose music which is different to that of an African drummer. We can extend this to include animal creativity: the (plain looking) male Vogelkop Bowerbird will decorate the lawn in front of its bower in order to attract female Bowerbirds – we doubtless could distinguish a lawn which has been decorated by a human to one decorated by a Bowerbird [21] (who, for instance, has been known to consider litter such as Snickers wrappers to be highly decorative). In all of these, and countless more examples, it would be absurd to suggest that a member of one group is less creative than a member of another *simply on the grounds that we can distinguish which category they fall into*.[8] From here it is a natural step to argue that we should not discriminate against computers, even if their brand of creativity turns out to be distinguishable from human creativity (clearly this argument depends on one's motivation for studying CC).

Negrotti [23] suggests that instead of continuing to judge the computer's capabilities directly against those of the human mind, the potentials of the computer as an 'alternative intelligence' can be explored. Re-conceiving the nature of our interaction with the computer leads to a less impoverished appreciation of the human-computer as a creative assemblage. Just as it may be productive to think of the A in AI as standing for a respectable "alternative", rather than the rather derogatory "artificial", it may be productive in CC to aim to build systems which are creative in ways which are unique to machines. Humans and machines have different strengths, and rather than attempting to shoe-horn machines into a way of thinking which can be passed off as human, we should aim to develop computational systems which make the most of their strengths. It is simply carbon fascism to argue that only biological creativity is worth studying. Bedworth and Norwood [24] argue along such lines: instead of perceiving AI as recreating humans, they suggest that we should develop intelligent devices whose complexity could be used to complement human ability. Such devices would differ from the human mind in terms of nature and power, but be compatible with it. The TT forces us into the undesirable position, to paraphrase Hofstadter, of trying to make a machine act like it is not a machine.[9]

3.2 The Turing Test cannot take framing information into account

The context in which an idea or artefact has been created can affect how creative we judge the originator to be, and the value we ascribe to the idea/artefact. For example, an idea may be considered interesting if produced by a child or novice, yet dull if produced by an adult or expert, and similarly, the child/novice may be seen as more creative that the adult/expert. That is, the very thing that we are supposed to determine in a TT (who is responsible for a certain piece of work) is necessary information in the judgement of creativity. For that reason *interaction* is key, so the versions of the TT above which omit this, make the evaluation impossible. For instance, in the poetry magazine *Anon*, in which reviewers use the double blind review process to decide whether to accept or reject a poem, Askew [26] considers the difficulties of reviewing poetry without knowledge of the author. As an example, she cites a poem on childbirth, arguing that if it was written by a mother she would consider it rather mediocre, but if written by a man then she would consider it to be insightful and thoughtful. There is much work on the advantages and disadvantages

[6] These requests could be translated into a language which the program understands, without cheating, thus bypassing the need for verbal interaction.

[7] Personal communication.

[8] In psychology, inter-group comparisons have focussed on whether one group is more creative than another. For instance, work in developmental psychology such as [22] suggests that familiarity with a domain can be necessary for the flexibility required for creativity (Boden also subscribes to this view in her metaphor of exploration and transformation of conceptual spaces). Possible links between madness and creativity has been much explored, with proponents on either side (see [5]).

[9] The original quote is "... sometimes I think that all of AI has something of this playful, spoofing character. It is, after all, a delightful game to try and make a machine act like not a machine," ...[25, p. 475]

of blind peer review (for example [27]): while there are sometimes good arguments for double blind review, it is widely acknowledged to be difficult to fully evaluate a paper without the framing information of authorship and context.

3.3 The Turing Test rewards 'window dressing' and trickery

Many of the objections for using the TT to evaluate progress in AI carry over to CC. We shall not discuss most of them here: the most apt to creativity is a remark made by Lady Lovelace in her memoir on Babbage's Analytical Engine: "The Analytical Engine has no pretensions to originate anything. It can do whatever we know how to order it to perform." Turing considers this objection in [1]; both his response and Lady Lovelace's objection are explored by Boden [3] and Bringsjord, Bello and Ferrucci [28] and we do not expand them.

Hofstadter [10] addresses the issue we raised in §1 about encouraging developers of programs to focus on the wrong thing. He argues that in order to avoid the "race for flashier and flashier natural-language 'front ends' with little substance behind them", the person in the interrogator role must ask questions at the right sort of level, which will be difficult to achieve, and comments that "What is needed is a prize for advances in basic research, not a prize for window-dressing." [25, p. 491]. Techniques such as using random numbers to create what Hofstadter calls an "Artificial Wiggliness", in order to more closely resemble a hand-drawn figure could be seen in some situations as the equivalent in art programs of "flashy natural-language front ends". This is a technique used in the letterform-processing program MetaFont [29], as well as in AARON, and is hypothesised by Hofstadter to be key in our willingness to attribute AARON with artistic insight, despite being a simple, surface technique, of no real interest to CC researchers. Bringsjord *et al.* [28] argue that those in AI who do use the TT as a motivating goal know that they are competing in trickery; they are building programs which can *fool* a judge into believing that they are intelligent, rather than actually being intelligent. Thus, their goal is to create an agent which has a Chinese Room Argument-style rulebook comprehensive enough to be able to convince a judge: "In such scenarios it's really the human creators against the human judges; the intervening computation is in many ways simply along for the ride" [28, p. 2].

3.4 The Turing Test encourages pastiche

In §1 we argued that the motivation of the CC researcher will affect which evaluation criteria are appropriate. The problems with the TT and Computational Creativity are present, to different degrees, in different types of creativity, such as Boden's exploratory and transformational creativity, and other distinctions between everyday creativity and truly great creativity. In some circumstances, it may be appropriate for exploratory search to drive creative acts, but in others, this leads only to pastiche. As a particular example, while Photoshop image filters can produce images which look remarkably Impressionistic, it is very difficult to ascribe creativity to such processes as they do not innovate in either process or aesthetic evaluation. Given the value of such processes for graphic designers, etc., there is a danger that CC researchers will aim to write such pastiche generation software, missing the point of innovation and imagination in the creative process, and holding the study of creativity in software back, whatever the motivation of the CC researcher.

3.5 The Turing Test is simply too hard

We have seen that Boden argues that some systems have already passed her version of the TT. Similarly, Hofstadter argues that AARON's creations could "almost certainly be passed off as human art", and that they "look surprisingly like products of a sophisticated human artist" [10, p. 468]. Thus if we base a version of the TT on an inability to distinguish between human and computer-produced ideas, it appears that some systems may pass this test. However, in §2.4 we argue that tests based on the TT should include some form of interaction, and we suggested the sort of lines a TT for artistic creativity might follow. None of the systems so far discussed (nor any other in existence today) is anywhere close to passing this sort of test. Thus, even if the TT may at some point be a useful test of CC, it is not currently viable. While it may be useful to have a difficult (possibly unattainable) goal as an overall motivation, in practice CC needs pragmatic ways of measuring intermediate progress, which will enable us to objectively and falsifiably claim that program P_1 is more creative in ways X, Y and Z than program P_2 (where P_1 and P_2 may be different versions of the same program). Boden [3] suggests that it is more helpful to ask 'where does x lie in creativity space?' (assuming a continuous n-dimensional space for n criteria where we can measure each dimension), than 'is x creative?' (assuming a Boolean judgement), or even 'how creative is x?' (assuming a linear judgement). Turing-style tests do not allow for such subtleties. The recommendation of focusing on achievable goals in CC is echoed by Cardoso *et al*:

> To achieve human levels of Computational Creativity, we do not necessarily need to start big, at the level of whole poems, songs, stories or paintings; we are more likely to succeed if we are allowed to start small, at the level of simple but creative phrases, fragments and images [30, p. 17].

We take this to suggest that a measure of progress which covers the whole spectrum of possible achievement will be of greater practical use than one which only can only measure achievement of a grand vision.

4 Alternative suggestions: Two descriptive models

We have outlined problems with measures of CC that fail to value a type of creativity which may be specific to computers (§3.1), do not account for contextual information for a creative act (§3.2), or fail to reward genuine advances in CC (§3.3) or the genuinely novel over pastiche (§3.4). In particular, we argued for the need for workable measures which allow us to measure intermediate progree and make falsifiable claims about our programs (§3.5). These issues with Turing-style tests for CC help to motivate alternative measures of progress. In this section we describe our efforts to develop alternative measures which, we hope, avoid some of the pitfalls of the TT.

In [31, 32] we introduce and motivate two descriptive models, the FACE model and the IDEA model, which form a framework to aid us in the development and evaluation of creative software. These models are not intended to capture human creativity, nor even all of Computational Creativity. Our far more modest goal is to add another plank to the framework, begun by [33] and continued by [34], [35] and [19] to *provide a means of formalising some aspects of Computational Creativity*. At present, our discussion is limited to notions which could be used to describe creative software. While these notions are inspired by human creativity, we do not aim for a model of human creativity. Even within Computational Creativity, we merely

suggest that the FACE and IDEA models provide one possible way – by no means the only way – of describing software designed for creative purposes. The twin processes of generation and evaluation are considered fundamental within creativity studies (for instance, see [36, 33, 37, 38]). We maintain this distinction in our two complementary models; FACE, which proposes acts of creativity as the fundamental units to be assessed in creative systems, and IDEA, which describes ways of evaluating the acts.

4.1 The FACE model

The FACE model assumes eight kinds of generative acts, which produce the following kinds of results:

F^p:	a method for generating framing information
F^g:	an item of framing information
A^p:	a method for generating aesthetic measures
A^g:	an aesthetic measure
C^p:	a method for generating concepts
C^g:	a concept
E^p:	a method for generating expressions of a concept
E^g:	an expression of a concept

In order to cover as many creative acts as possible, we assume only that there must be something new created for the question of creativity to arise. This could be very small, a brush stroke of an artist, an inference step by a mathematician, a single note written in a piece of music. Our model, then, covers "merely generative" acts as well as "fundamentally generative" acts. Thus, by drawing our base line at the lowest level, our model can be used to describe the most basic "creative act" possible, and we avoid the thorny issue of where an act of creation starts. Important questions about where on the scale from basic to sophisticated an act must be to be judged creative, can be postponed.

In [30], Cardoso, Veale and Wiggins describe *The Upsidedowns of Gustav Verbeek*. These are panels which tell a story up to a half way point, the continuation of which then appears almost magically when one turns the panels upside down. Cardoso *et al.* celebrate the "artfulness" of Verbeek, while lamenting the "almost painful" gap between human and machine creativity: however they also show a simpler example of the same principle, which, they argue, *is* within reach of Computational Creativity. We show another example of this type in Figure 1. While the FACE model is designed for describing creative acts undertaken by computer, it is illustrative to describe (theoretically) how creative acts in human artistic endeavours might produce artwork such as the Verbeek piece described above. In particular, we could describe Verbeek as having undertaken a creative act of the form $\langle C^g, E^g \rangle$, which comprises an expression E^g of the concept C^g that the picture must make sense when upside down (and fit into the story). We could further describe this creative act as building on the results of multiple previous creative acts, for instance where the aesthetic A^g was invented as the notion of art having different meanings when viewed from multiple perspectives; and the generation of framing information F^g including contextual history of this genre of art, the artist's motivation, justification, etc.

Still using the Verbeek example as inspiration, at the process invention level, creative acts involving generative acts of the form F^g produce new methods for expressing the concept of art which have a different meaning when viewed upside down (for example, birds flying in the sky can double as waves in the sea, or a hat on one's head can double as a mouth on one's face). Moreover, creative acts involving generative acts of the form C^p produce methods for generating new perspectives from which the art might make sense (other examples would be rotating 90° rather than 180° - see Figure 2, or three-dimensional or moving images). Finally, methods for generating the aesthetic of art having multiple meanings when viewed from multiple perspectives would be denoted within creative acts involving generative acts A^p (another example would be the aesthetic of art having multiple meanings when viewed from a single perspective), and generative acts of the form F^p might include methods for generating new motivations, justifications etc.

Figure 1. A man coming out of the water – rotate 180° to see the same man drowning

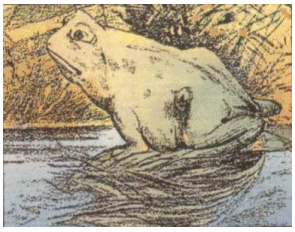

Figure 2. A frog – rotate 90° to see a horse

Clearly, not all of these generative aspects may be present in a single creative act, and they may be performed by different parties. While the model is not broad enough to cover all potentially creative software systems, we believe that it covers more than enough to guide and describe the first wave of creative systems. For example, a system which was able to perform creative acts involving generative acts of the form F^p would be more sophisticated than anything we have now: this is producing new ways to generate justifications and explanations of a creative act.

In [31], we use the FACE model to suggest ways in which different pieces of software for the same type of tasks – or indeed different versions of the same creative software – could be assessed. In particular, we suggest that a simple *quantitative* approach whereby a count of the number of creative acts produced in a given time period might be used. An alternative, or supplementary, approach might be *cumulative*, whereby software is assessed as more creative if it performs creative acts involving more types of generative acts, or a particular ordering of types of creative act could be put forward for individual domains of discourse. For instance, it could be argued that software is more creative if it invents and utilises an aesthetic measure rather than just employing a given one. We also suggest a various *qualitative* approaches where the value of the results of the creative acts of the form $\langle C^G, E^G \rangle$ are assessed against given (or invented) aesthetic measures. For instance, the average quality of the results of creative acts might be used, or an analysis of the worst ever, or best ever might be more appropriate. Finally, we suggest that the types of methods

employed within the individual generative acts might be used to differentiate creative software. For instance, a random method might be seen as less creative than one which uses induction, etc.

4.2 The IDEA model

Within the IDEA model, we begin to formalise notions of how creative acts can be measured, in terms of notions related to impact. We simplify matters by assuming an (I)terative (D)evelopment (E)xecution (A)ppreciation cycle within which software is engineered and its behaviour is exposed to an audience. We generalise past usual AI notions of correctness, soundness and value, because we are in a situation where software is meant to invent its own aesthetic or utilitarian criteria, rather than simply optimise solutions with respect to given value measures. To do this, we assume an *ideal audience* of individuals i, which is able to provide two indicators of the effect that an individual creative act, A, has had on them: (a) an indication of their change in well-being, $wb_i(A)$, between -1 and 1, with -1 indicating that they felt worse, +1 indicating that they felt better, and 0 indicating ambivalence, and (b) an indication between 0 and 1 of the cognitive effort they spent in trying to appreciate a creative act and the artefact(s) it produced, $ce_i(A)$. Denoting the mean value of the well-being rating over the n people as $m(A)$, we propose the following measures for use in impact assessment exercises:

$dis(A) = disgust(A) = \frac{1}{2n}\sum_{i=1}^{n}(1 - wb_i(A))$
$div(A) = divisiveness(A) = \frac{1}{n}\sum_{i=1}^{n}|wb_i(A) - m(A)|$
$ind(A) = indifference(A) = 1 - \frac{1}{n}\sum_{i=1}^{n}|wb_i(A)|$
$pop(A) = popularity(A) = \frac{1}{2n}\sum_{i=1}^{n}(1 + wb_i(A))$
$prov(A) = provocation(A) = \frac{1}{n}\sum_{i=1}^{n}(ce_i(A))$

By compounding the provocation measure with the others, we can attempt to capture some kinds of impact that creative acts might have:

$acquired\_taste(A) = (pop(A) + prov(A))/2$
$instant\_appeal(A) = (1 + pop(A) - prov(A))/2$
$opinion\_splitting(A) = (1 + div(A) - prov(A))/2$
$opinion\_forming(A) = (div(A) + prov(A))/2$
$shock(A) = (1 + dis(A) - prov(A))/2$
$subversion(A) = (dis(A) + prov(A))/2$

These all return a value between 0 and 1, and we argue that if A reaches towards 1 for any of these measures, it has had some impact, such as being shocking, or divisive.

In [31], we flesh out the models, by including notions of ideal background information and an ideal programming environment, and using these to suggest further ways to compare the creative acts performed by software and their impact. In particular, we suggest six stages for the development of software for creative purposes: (*i*) a developmental stage: where all the creative acts undertaken by the software are based on inspiring examples (using terminology from [35] (*ii*) a fine-tuning stage: where the creative acts performed are abstracted away from inspiring examples, but are still too close to have an impact as novel inventions (*iii*) a re-invention stage: where the software performs creative acts similar to ones which are known, but which were not explicitly provided by the programmer (*iv*) a discovery stage: where the software performs creative acts sufficiently dissimilar to known ones to have an impact due to novelty, but sufficiently similar to be assessed within current contexts (*v*) a disruption stage: where the software performs some creative acts which are too dissimilar to those known to the world to be assessed in current contexts, hence new contexts have to be invented, and (*vi*) a disorientation stage: where all the creative acts performed are too dissimilar to known ones for there to be any context within which to judge any of the activities of the software. We suggest that an analysis of the software with respect to which stage of development it is in, can be used to compare and contrast creative programs.

5 Conclusions and Further Work

We have described Computational Creativity as the AI subfield in which we study how to build software that models creative thought in science and the arts. In order to have a notion of progress, and to set an agenda for researchers who are modelling aspects of creative thought, it is essential to agree practical evaluation measures, based on sound theoretical foundations, which we can apply to our programs to help to identify aspects which are satisfactory and those which should be improved. We have discussed the use of the Turing Test, and different versions of it, for such purposes, and argued that it is largely inappropriate in this context. This is because attempts to pass the Turing Test may result in losing differing, and valuable, styles of creativity; might fail to take into account the importance of background and contextual information for a creative act; encourage superficial and uninteresting advances in front-ends; and result in rewarding creativity which adheres to a certain style over that which creates something which is genuinely novel. We suggest that although there may be some place for Turing-style tests for Computational Creativity at some point in the future, it is currently untenable and impractical.

As an alternative to Turing-style tests, we introduce two descriptive models for evaluating creative software, the FACE model which describes creative acts performed by software in terms of tuples of generative acts, and the IDEA model which describes how such creative acts can have an impact upon an ideal audience, given ideal information about background knowledge and the software development process. We believe that these alternative measures constitute a beginning in our efforts to avoid some of the pitfalls of the TT: they do not discriminate against a creativity which may be specific to computers, they take contextual information into account via the framing aspect of the FACE model, they reward genuine advances in CC and the genuinely novel over pastiche. Perhaps most importantly, we believe that they are workable measures which will enable us to measure intermediate progress and make falsifiable claims about our programs. We demonstrate the practicability of the descriptive models in [31], where we use them within comparison studies of existing software built for creative purposes. In particular, we compare and contrast mathematical invention software including the AM [39], HR [40] and HRL [41] programs. We similarly compare and contrast various pieces of generative art software, including the AARON program [12], The Painting Fool [42] and the NEvAr evolutionary art software [43]. Moreover, in [32], we further motivate the FACE and IDEA models by appealing to some of the authors mentioned above, and others like Sloman [44] and Thagard [45], who suggest criteria against which these descriptive models might be judged. We place the work in the context of existing approaches to the assessment of creativity in software, and in a wider context of creativity studies, in addition to providing a case study: the Basel problem from mathematics, described in [46] as the "best known problem of the time".

In [47], we suggest methods, methodologies and paradigms within which creative software might be written. In particular, we propose some ways in which to manage the public perception of creativity (or lack thereof) in computers. The descriptive models presented above are intended as a complement to these public perception guidelines, whereby AI practitioners can rely on concrete assessment methods

for the usually difficult topic of apportioning creativity to software. The FACE and IDEA descriptive models are not yet particularly acute tools for a full assessment of creativity in software, and we plan to develop sub-models for various notions which have been used to describe the creativity (or lack thereof) in computer systems in recent years. These terms include, but are not limited to, the following: affect, analogy, appreciation, audience, autonomy, blending, community, context, curiosity, exploration, framing, humanity, humour, idea formation, imagination, intentionality, interaction, interpretation, knowledge, metaphor, novelty, obfuscation, personality, physicality, playfulness, problem solving, process, programming, search, surprise, transformation and trust. Using the foundational terminology for creative acts and impact described above, we plan to expand each term into a formalism containing conceptual definitions and concrete calculations using those definitions which can be used for the assessment of creativity in software. In doing so, we hope to contribute a *Computational Creativity Theory* which will provide a strong foundation for objectively measured progress in our field.

Acknowledgements

We are very grateful to John Charnley for his thoughts on the FACE and IDEA descriptive models. We would also like to thank two anonymous reviewers for their helpful comments. This work is supported by EPSRC grants EP/F035594/1 and EP/F036647/1.

REFERENCES

[1] A. M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[2] A. Kozbelt, R. A. Beghetto, and M. A. Runco. Theories of creativity. In J. C. Kaufman and R. J. Sternberg, editors, *The Cambridge Handbook of Creativity*, pages 20–47. Cambridge University Press, USA, 2010.

[3] M.A. Boden. *The Creative Mind: Myths and Mechanisms*. Weidenfield and Nicholson, London, 1990.

[4] J. A. Plucker and M. C. Makel. Assessment of creativity. In J. C. Kaufman and R. J. Sternberg, editors, *The Cambridge Handbook of Creativity*, pages 48–73. Cambridge University Press, USA, 2010.

[5] A. Rothenberg. *Creativity and Madness*. The John Hopkins University Press, Baltimore, USA, 1990.

[6] A. Newell, J. G. Shaw, and H. A. Simon. The process of creative thinking. In H. E. Gruber, G. Terrell, and M. Wertheimer, editors, *Contemporary Approaches to Creative Thinking*, page 63119. Atherton, New York, 1963.

[7] M. T. Pearce and G. A. Wiggins. Towards a framework for the evaluation of machine compositions. In G. A. Wiggins, editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*, 2001.

[8] A. Garnham. Art for arts's sake. *Behavioural and Brain Sciences*, 17(3):543–544, 1994.

[9] A. Bundy. What is the difference between real creativity and mere novelty? *Behavioural and Brain Sciences*, 17(3):533–534, 1994. Open peer commentary on [3].

[10] D. Hofstadter. Epilogue: on computers, creativity, credit, brain mechanisms, and the Turing test. In D. Hofstadter and the Fluid Analogies Research Group, editors, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, pages 467–491. Basic Books, 1995. Epilogue in [25].

[11] M. A. Boden. The Turing test and artistic creativity. *Kybernetes*, 39(3):409–413, 2010.

[12] P. McCorduck. *AARON's Code: Meta-Art, Artificial Intelligence, and the Work of Harold Cohen*. Freeman, New York, 1991.

[13] M. A. Boden and E. A. Edmonds. What is generative art? *Digital Creativity*, 20(1-2):21–46, 2009.

[14] D. Cope. *Virtual Music: Computer Synthesis of Musical Style*. The MIT Press, Cambridge, Massachusetts, 2001.

[15] D. Cope. *Computer Models of Musical Creativity*. The MIT Press, Cambridge, Massachusetts, 2006.

[16] S. C. Todd and W. Latham. *Evolutionary Art and Computers*. Academic Press, London, 1992.

[17] K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4):319–328, July 1991.

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

[19] S. Colton. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*, 2008.

[20] J. C. Kaufman and R. J. Sternberg, editors. *The International Handbook of Creativity*. Cambridge University Press, Cambridge, New York, USA, 2006.

[21] J. Diamond. Animal art: Variation in bower decorating style among male bowerbirds *amblyornis inornatus*. *Proceedings of the National Academy of Sciences in the USA*, 83(9):3042–3046, May 1, 1986.

[22] A. Karmiloff-Smith. Constraints on representational change: Evidence from children's drawing. *Cognition*, (34):57–83, 1990.

[23] M. Negrotti. *Understanding the Artificial*. Springer-Verlag, London, 1991.

[24] J. Bedworth and J. Norwood. The Turing test is dead. In *Proceedings of the 3rd conference on creativity and cognition*, 1999.

[25] D. Hofstadter and the Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, NY, USA, 1995.

[26] C. Askew. Blind faith and anonymity. *Anon*, 7:pp. 55–58, 2010.

[27] F. Rowland. The peer-review process. *Learned Publishing*, 15(4), 2002.

[28] S. Bringsjord, P. Bello, and D. Ferrucci. Creativity, the Turing test, and the (better) Lovelace test. *Minds and Machines*, (11):3–27, 2001.

[29] D. E. Knuth. The concept of a meta-font. *Visible Language*, 16(1):3–27, 1982.

[30] 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):15–22, 2009.

[31] S. Colton, A. Pease, and J. Charnley. Computational creativity theory: The FACE and IDEA descriptive models. In *2nd International Conference on Computational Creativity*. 2011, 2011.

[32] A. Pease and S. Colton. Computational creativity theory: Inspirations behind the FACE and the IDEA models. In *2nd International Conference on Computational Creativity*. 2011, 2011.

[33] M Boden. *The Creative Mind: Myths and Mechanisms (second edition)*. Routledge, 2003.

[34] G. A. Wiggins. Searching for computational creativity. *New Generation Computing*, 24(3):209–222, 2006.

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

[36] G. Wallas. *The Art of Thought*. Harcourt Brace, NY,USA, 1926.

[37] R Finke, T Ward, and S. Smith. *Creative cognition: Theory, research and applications*. MIT press, Cambridge, 1992.

[38] G. Ritchie. Assessing creativity. In G. A. Wiggins, editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*, pages 3–11. SSAISB, 2001.

[39] D. B. Lenat. Automated theory formation in mathematics. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 833–842, Cambridge, MA, 1977. Morgan Kaufmann.

[40] S. Colton. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag, 2002.

[41] A. Pease. *A Computational Model of Lakatos-style Reasoning*. PhD thesis, University of Edinburgh, 2007.

[42] A Krzeczkowska, J El-Hage, S Colton, and S Clark. Automated collage generation - with intent. In *Proceedings of the 1st International Conference on Computational Creativity*, 2010.

[43] P Machado and A Cardoso. NEvAr – the assessment of an evolutionary art tool. In *Proceedings of the AISB00 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, 2000.

[44] A. Sloman. *The Computer Revolution in Philosophy*. The Harvester Press, Ltd., 1978.

[45] P. Thagard. *Computational Philosophy of Science*. MIT Press, Cambridge, Mass, 1993.

[46] C. E. Sandifer. *The early mathematics of Leonhard Euler*. The Mathematical Association of America, 2007.

[47] S Colton. Seven catchy phrases for computational creativity research. In *Proceedings of the Dagstuhl Seminar: Computational Creativity: An Interdisciplinary Approach*, 2009.