Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes

Diarmuid P. O'Donoghue, James Power, Sian O'Briain, Feng Dong[*], Aidan Mooney, Donny Hurley, Yalemisew Abgaz, Charles Markham,

Department of Computer Science, NUI Maynooth, Co. Kildare, Ireland.

[*] Department of Computer Science and Technology, University of Bedfordshire, Luton, UK.

Abstract

This paper begins by briefly looking at two of the dominant perspectives on computational creativity; focusing on the creative *artefacts* and the creative *processes* respectively. We briefly describe two projects; one focused on (artistic) creative artefacts the other on a (scientific) creative process, to highlight some similarities and differences in approach. We then look at a 2-dimensional model of Learning Objectives that uses independent axes of *knowledge* and (cognitive) *processes*. This educational framework is then used to cast artefact and process perspectives into a common framework, opening up new possibilities for discussing and comparing creativity between them. Finally, arising from our model of creative processes, we propose a new and broad 4-level hierarchy of computational creativity, which asserts that the highest level of computational creativity involves processes whose creativity is comparable to that of the originating process itself.

Introduction

Creativity is frequently seen through the "search space" metaphor (Boden, 1992; O'Donoghue and Crean, 2002; Wiggins, 2006; O'Donoghue *et al*, 2006; Ritchie, 2012; Veale, 2012; Pease *et al*, 2013). The space of possible *products* is represented as physical space, where each location represents a different product. Other search processes have been through this space previously, so a creative search process attempts to focus on regions of this space that have not yet been explored. The space of all search products carries different, often unpredictable values (including novelty). Boden (1992) identified three levels of creativity with *improbable* creativity exploring regions of this search space that are unlikely to have been visited previously. *Exploratory* creativity deliberately attempts to explore the boundaries of that search space. *Transformational* creativity attempts to identify and explore new search spaces, to identify products that did not exist in the original search space.

Viewing computational creativity through this search space metaphor, we can see that many artistic forms of creativity are adequately described. Artistic styles of creativity can be seen to explore the space of possible creative *artefacts* from one of the traditional creative domains like art, music, creative writing *etc.* (as used in Carson *et al*, 2005). Highly creative individuals transform accepted search spaces to create new possibilities – such as impressionism or cubism.

Creative artefacts and creative processes are generally discussed quite separately, with creative products/artefacts attracting the most attention. One criticism often levelled at the discipline of computational creativity, is that it is overly focused on creative products – paying too little attention to the process (Stojanov and Indurkhya, 2012; O'Donoghue and Keane, 2012). Analogy, metaphor are often seen as the dominant approaches to processes centred creativity, though evolutionary computing approaches are also popular. These creative processes appear to be generally associated with creativity within scientific or engineering types of disciplines. Thus, the starting point for this paper concerns the two distinct perspectives on computational creativity, focusing on *artistic products* and *scientific processes*. Later in this paper we shall use an educational assessment framework to cast both perspectives into a common framework, in order to bring resolution to these apparently conflicting perspectives.

It should be noted that even the basic distinction between artistic creativity and scientific creativity is not universally accepted. The noted 18th century mathematician (and poet) W. R. Hamilton regarded mathematics *"as an aesthetic creation, akin to poetry, with its own mysteries and moments of profound revelation"* (from Hankins, 1980). Mathematicians have also compared the aesthetic beauty of various equations, with Euler's identity ($e^{i\pi} + 1 = 0$) ranked the most beautiful equation in mathematics (Wells, 1990). Conversely, the process of analogical reasoning is generally seen as a driving force of scientific creativity (Brown, 2003), but at least one study has shown that analogical reasoning appears to play a part in some contemporary artistic creativity (Okada *et al*, 2009). Despite these overlaps, we shall proceed with the two basic categories of creative products and creative processes for the purposes of this paper.

Creative Products and Creative Processes

We briefly compare and contrast creative products (or artefacts) and creative processes using two projects that serve to highlight some commonalities and help identify some differences. The first is *ImageBlender* that creates new images using complex transformations of two given input images. The second *RegExEvolver* represents simple pro-

cesses (a finite automaton) as regular expressions, creating new regular expressions from that expression.

Another criticism often levelled at computationally creative systems is that *"Most of them are given, in advance, a detailed (hardcoded) description of the domain"* (Stojanov and Indurkhya, 2012). The two models presented in this paper make minimal assumptions about their relevant problem domains. ImageBlender is based on the assumption that the inspiring set contains images - regardless of what those images depict. RegExEvolver assumes only that the input is a valid regular expression – again with no additional limits. Additionally, both models take a very small inspiring set of two and just one items respectively.

Both systems use the search and evaluate strategy of evolutionary computation to explore the space of possible outputs. Both adopt a multi-objective selection strategy (Luke, 2013) to promote the emergence of high quality outputs. Multi-objective evaluation uses several independent objective functions to evaluate individuals in the population. Evolution then proceeds under the guidance of a Pareto-optimal selection strategy.

Finally, both projects use *interesting-ness* as one of the objective functions to guide evolution towards the creation of solutions. In both cases interestingness is estimated by the Kolmogorov complexity of the created output. This use of Kolmogorov complexity is slightly different to that discussed by McGregor (2007). Other metrics are used to ensure that the results have some measurable *novelty* compared to the given inspiring set – by measuring the dissimilarity between an evolved output and the given input(s).

These two metrics of interestingness and novelty are used as simple, general purpose estimates of the quality and novelty (Ritchie, 2001) that are sought by creative systems. We shall now see if these minimal assumptions can prove useful for computational creativity – in the absence of more detailed information on the problem domain.

Creative Artefacts from *ImageBlender* *ImageBlender* creates new images by combining two given input images. Well known techniques exist for combining two images using techniques like; super-positioning those images; selecting and combining sub-regions of the images using image manipulators like rotation, translation, scale, reflection *etc*. Many such techniques can be considered as collage generation that selectively combine parts of (two or more) given images.

However, ImageBlender does not operate directly upon the images but explores the space of possible images produced by combining transformed representations of those images. This process might be considered transformational in that it explores a space of possible images that has not been explicitly explored before (as far the authors can ascertain). ImageBlender currently focuses on the *Fast Fourier Transform (FFT)* of those images, creating a new image by combining portions of the *phase* and *frequency* information from those images. ImageBlender explores the space of possible images produced by various combinations of FFT's and then using the inverse transform (FFT^{-1}) to produce the resulting image. No restrictions are placed on the input images – other than those inherent to the FFT transform. Thus images may be black and white, greyscale, or colour; representing geometric figures, paintings, photographs *etc*. or any combination of these.

ImageBlender uses evolutionary computation to produce creative images, guided by a Pareto-optimal selection strategy. Among the metrics used are a number of estimates of the Kolmogorov complexity of the output image – ensuring there is some appropriate level of interestingness associated with the output images. Other metrics favour new images that are different from both input images.

Interestingly, some of these measures also have a role in assessing the beauty of images. Forsythe *et al* (2010) found that visual complexity can be adequately assessed using GIF compression and that the fractal dimension of an image often appears to be an adequate predictor of people's judgements of beauty.

Figure 1 shows two input images formed from black and white pixels only; a "checkerboard" of alternating black and white pixels (top left) and a black circle on a white background (top right of Figure 1). The grey appearance of the first image is caused by the low resolution reproduction of alternating black and white pixels. The final image was formed by combining the phase information from one image with the frequency information from the other, forming the third (bottom) image in Figure 1. Surprisingly, the output image has a far higher Kolmogorov complexity than either input image, suggesting a more interesting product. We argue that this output is creative in that it has the properties most frequently associated with creativity, it is: novel, interesting, unexpected and (arguably) has some aesthetic if geometric beauty. Appexdix 1 contains a few more sample images created by ImageBlender.

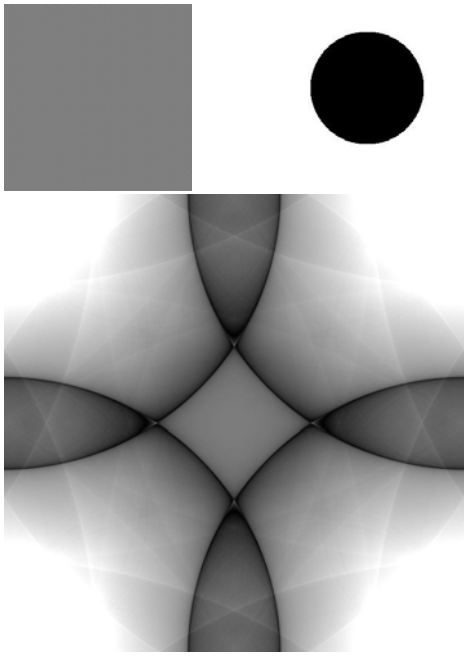

Figure 1: The two input images (above) and the new image (below) formed by blending the FFT of these images.

Creative Processes with RegExEvolver Computational creativity has addressed process centred creativity under three main categories: traditional GOFAI (Good Old Fashioned Artificial Intelligence) search processes, evolutionary search and analogy/metaphor/blending (Veale and O'Donoghue, 2000) approaches. However, instead of focusing on specific processes we look instead at general Turing Machine models of computational processes.

In this section we consider the case of creating outputs that are themselves processes. Creating a process rather than an "artefact" shouldn't in principle be that much of a change since computational processes are easily represented as strings of characters, parse trees or other structures. Such representations can allow "traditional" creativity e search to explore the space of possible artefacts/processes. In fact, evolutionary programming, genetic programming and grammatical evolution regularly output new programs in some executable programming language, though their focus in not normally on creative outputs. This situation where the creative output is itself a process also underpins the later section (below) that integrates creative processes and products through a theory of Educational Assessment.

A number of previous project have looked at creating outputs that are themselves processes. Procedural content generation (Togelius *et al*, 2011) is an emerging area devoted to the creation of game content for playable computer games. Cook *et al* (2013) discuss the Mechanic Miner system that generates the game mechanics for platform games using evolutionary computation. However Mechanic Miner and other procedural content generators are very focused on the domain of platform games and not on general purpose software development.

The Arís model (Pitu *et al*, 2013) creates formal specifications (in Spec#) for a given implementation (in C#) using analogical reasoning. Due to the creative and arguably unreliable nature of analogical reasoning, Arís uses a theorem prover to validate the inferences it automatically accepts. But unverified specifications may also spur the workaday *little-c creativity* (Gardner, 1993) of human specification writers. Finally, we note that Arís is also (potentially) capable of operating in the reverse direction, creating new source code (a process) for a given specification.

Many practitioners of computational creativity use the concept of *inspiring sets* to describe both the creative domain and (a sample of) the artefacts that have already been generated within that domain. In this section we briefly look at the creation of simple computational processes, as represented by *Regular Expressions (RegEx).* Each regular expression defines a language, and any regular expression can be converted to a *Finite State Machine (FSM)* that recognises strings from this language. The RegExEvolver project uses just one regular expression for its inspiring set and attempts to create new and potentially useful expressions from it.

As a simple example, a regular expression for the registration numbers of Irish vehicles before 2013 would be:

```
[0-9]{2}[A-Z]{1,2}[0-9]{1,5}
```

After this date, a new system was introduced conforming to the following regular expression:

```
[0-9]{2}[1-2]{1}[A-Z]{1,2}[0-9]{1,5}
```

As a second example we consider the rules for valid passwords used in a computer system. Valid passwords may be specified by a regular expression, with different "strengths" associated with different expressions. A weak expression might accept any combination of letters and numbers, but a stronger expression might require at least one of each of: a lower case letter, an upper case letter and a digit. RegExEvolver could also be used to create a new password specification given a pre-existing expression.

The process is similar to that used in fuzz-testing (Godfried *et al*, 2012), a software engineering technique used to find bugs in a program. One approach to ('black-box') fuzz testing involves analysing existing test inputs and then generating different, new inputs that may expose previously unknown vulnerabilities. A more sophisticated ('white-box') approach involves analysing the program's source code in order to generate test inputs that cause unexpected combinations of the program's flow of control. Common to both approaches is the goal of creating new combinations that had not been previously envisaged by the testers.

RegExEvolver uses evolutionary computation techniques to guide formation of the new RegEx under the guidance of a Pareto-optimal selection technique. The objective functions focus on the original and evolved expressions and also assess the languages that are generated by these expressions. To this end RegExEvolver uses the Xeger tool to generate random strings for any given RegEx. This is achieved by employing standard algorithms to convert the RegEx to an equivalent FSM and then choosing random transitions through this machine. Although repetition (denoted by the Kleene Star '*') in a RegEx can theoretically generate a string of infinite length, this is not an issue in practice as it would require the same transition to be chosen every time.

In addition to evaluating the generated strings (products) we also evaluate the processes themselves. The generated RegEx is compared to the original (input) RegEx by calculating the intersection of their corresponding FSM using the `dk.brics.automaton` package. In this way, evaluation of the new process (RegEx) itself ensures it overlaps the input expression, while also ensuring it contains some *novelty* compared to the input expression. However, in the absence of a problem domain, we do not evaluate the *usefulness* quality of the generated expressions.

RegExEvolver is focused on generating *novel* and potentially *useful* "processes" at level 3 of the Chomsky hierarchy. However, it is easy to see that other computationally creative processes could generate creative processes at any level from the Chomsky Hierarchy. It has been shown that the set of regular languages corresponding to regular expressions (or produced by a regular grammar) at level 3 are a subset of the set of context free languages at level 2, which in turn are a subset of the set of content sensitive languages at level 1, and that these in turn are a subset of

the set of recursively enumerable languages at level 0 (Chomsky, 1959).

Evaluating Creativity

Both ImageBlender and RegExEvolver create new outputs without the benefit of any specific context or the constraints and values that frequently arise from such contexts. Thus, evaluating their outputs can be considered all the more difficult. While this might be seen as a weakness, we see it as positive support for the generality of our approach. That is, some creativity is possible without making detailed assumptions about the target domain – without committing to some low level detail that will later limit the breadth or *flexibility* (Guilford, 1950) of our creative system.

Defeasible Creativity Newell, Shaw and Simon (1963) highlighted that one criterion for creativity is that a given answer should cause us to reject an answer that we had previously accepted. From this perspective computational creativity should place its highest value on creativity that contradicts some existing belief, leading to the "shock and amazement" often associated with H-Creativity.

Evaluation plays a central role in computational creativity. We identify two distinct types of evaluation: Subjective evaluation and Objective evaluation. Subjective evaluation is carried out by a computationally creative process to ensure the quality and novelty of the output. However that real value of a creative output can only ever be truly determined by an independent group of evaluations. A true determination of the qualities of novelty and/or quality can only ever be made by an independent adjudicator.

Objective evaluation relies heavily on consensus reality and thus on some target population of evaluators – either the general public or some target group of critics. To this end, a comprehensive model of computational creativity must incorporate a model of the beliefs of that target group. Thus a creative system must either implicitly or explicitly, incorporate a model of the beliefs of that target group of evaluators. Thus, a *Theory of Mind* (ToM) is a fundamental issue in computational creativity – be that either an explicit theory or one implicitly instantiated in the model and its use of data (such as the inspiring set). Any ToM will suffer inaccuracies and other problems, especially when it is used within the context of creative reasoning. Thus, we conclude that a defining characteristic of computational creativity is that the output can only be truly evaluated and assessed by an independent adjudicator.

In effect, the objective metrics used in the two projects described above implicitly incorporate a simple ToM in terms of the *interestingness* value estimated by multi-valued pareto-optimal values, including the Kolmogorov complexity of the created products.

Integrating Creative Products and Creative Processes

Creative *products* and creative *processes* appear to bring different perspectives to computational creativity. Often, it appears that these perspectives are almost irreconcilable in terms of their values and objectives. We now explore one means of resolving the apparent differences between the product and process perspectives of computational creativity. The integration we explore is at the cognitive level, but it also bares relevance to other levels of creativity; from the neurological to the sociological.

In this section we review some work on education, as this is another discipline that values the creativity of its outputs – promoting the creativity of students produced by educational systems. Bloom's (1956) taxonomy of Learning Objectives (top of Figure 2) tried to get away from simple rote learning and promote higher forms of learning such as evaluating and analysing. The taxonomy was primarily aimed at informing education and assessment activities. The taxonomy was aimed at supporting objective assessment of educational activities and thus focuses on measurable and quantifiable properties.

While rote learning was seen as the lowest form of education attainment, synthesis and evaluation were seen as the highest achievements in the original (1956) taxonomy. "Creation" was only included in this original taxonomy as part of the "Synthesis" category and surprisingly, Synthesis was seen as a lower level of attainment than "Evaluation".

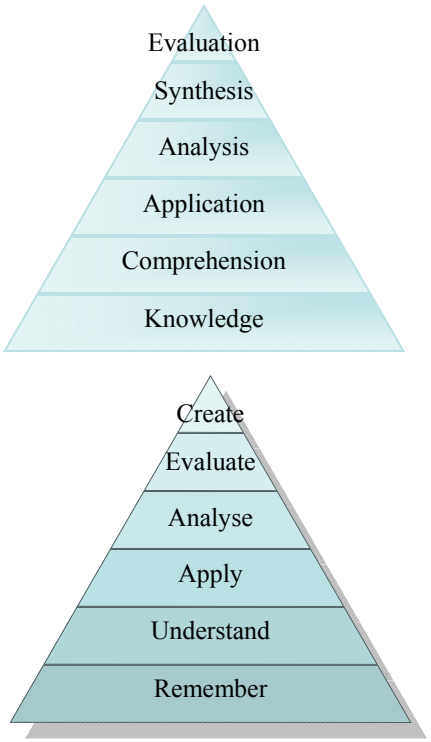

Figure 2: Bloom's Revised Taxonomy (below) places greater emphasis on the role of creativity in educational attainment

Bloom's Revised Taxonomy A subsequent revision of this taxonomy (Anderson and Krathwohl, 2001) (bottom of Figure 2) introduced a number of changes as moving from noun based to a verb based form and other changes. One of the most significant changes involved the introduction of "Create" as the highest level of educational attainment and a "demotion" of Evaluation below the Create level.

Learning Objectives Matrix

As noted by Krathwohl (2002) the unidimensional hierarchy of Bloom's Revised Taxonomy incorporated both noun/knowledge and verb/process and thus was essentially dual in nature. Anderson and Krathwohl (2001) overcame this problem by separating the noun (knowledge) dimension from the verb (process) dimension.

This resulted in a two dimensional matrix, with one axis called The Knowledge Dimension representing the noun related information. The other axis is called The Process Dimension and this represents verb related information. Towards the origin of this array are found some of the simplest forms of educational attainment, involving rote learning and the listing of facts. Furthest from the origin then, are the highest forms of educational attainment - notably including "create".

At this point we should acknowledge that Krathwohl's original diagram was a simple 2D matrix. However, in Figure 3 we depict a representation due to Rex Heer's model[1] that uses the third dimension to highlight the difference between the simpler and higher forms of educational attainment. Thus, the simpler forms of learning are depicted with the least height, while the highest forms of learning are depicted by greater heights. Heer's model and this paper make the assumption by using the third dimension that the Cognitive Process and Knowledge Dimensions are represented to the same scale. However, relative heights are merely suggestive of the levels of educational attainment.

Learning Objectives are typically stated in the form "*The learner will be able to do X with Y*" where X is a verb representing the relevant cognitive process and Y is a noun representing the corresponding knowledge. Of course, both the X and Y are sourced from the two axes of Figure 3. For example, "*The learner will be able to remember the law of supply and demand*" where X is "*remember*" and Y is "*the law of supply and demand*". The nouns and verbs on the two axes, along with the verbs contained in each vertex of the matrix, provide a terminology and reference points to describe and discuss different creative systems.

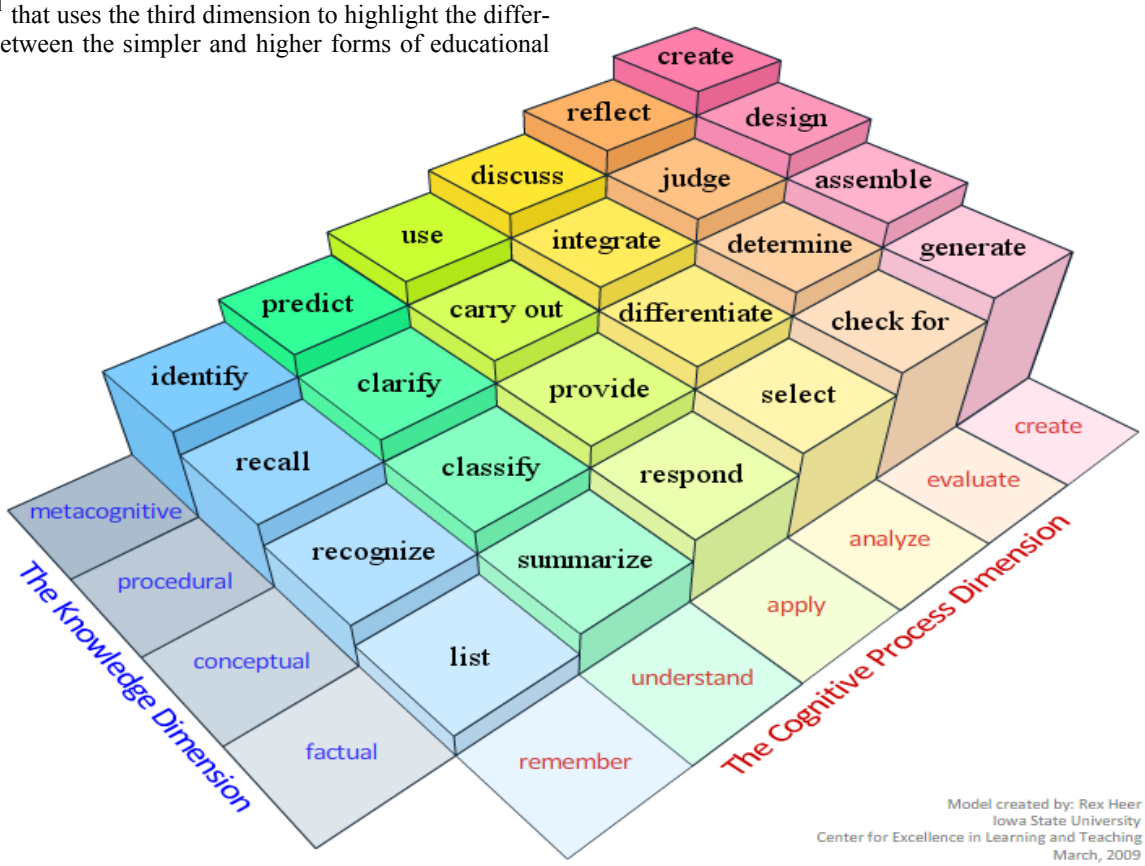

Figure 3: A 3D representation[1] of Krathwohl's 2D Matrix of Educational Assessment. This is used to view the artefact and process perspectives of computational creativity within a common framework.

[1] This image has been reproduced from: A Model of Learning Objectives–based on *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives* by Rex Heer, Center for Excellence in Learning and Teaching (CELT), Iowa State University.

We propose an adaptation of this taxonomy for the purposes of informing work on computational creativity. Adapting the typical statement of Learning Objectives to the domain of computational creativity, we suggest that we read this as *"A computationally creative system should be able to do X with Y",* where X and Y are identified from the diagram in Figure 3.

Of course, we acknowledge that adopting this matrix is contingent upon accepting some similarity between an *artefact* and the *knowledge* that it embodies. We feel that allowing this comparison may provide a new and useful perspective on computational creativity.

The Knowledge Dimension Firstly we look at the Knowledge Dimension of Figure 3. This we liken to the artefact perspective of computational creativity, as both are concerned with the production of new ideas in the form of knowledge or artefacts that represent that knowledge.

Factual: Knowledge of the basic elements of the discipline, essential facts, terminology and details. Factual knowledge details the basic elements required to function in some discipline – music, art, maths etc.

Conceptual: knowledge of classifications, categories and generalisations; knowledge of theories, models, and structures. Knowledge about how factual elements can be related and combined to form low level structures; this might include ontological and other knowledge (*warm* colours, *emotive* words).

Procedural: knowledge of genre-specific skills, algorithms and techniques, knowledge of criteria for determining when to use appropriate procedures, details how to do something; skills, algorithms, techniques and method, including their use.

Metacognitive: strategic knowledge, knowledge about the cognitive tasks including appropriate contextual and conditional knowledge, self-knowledge, and awareness of one's own cognition (or the systems own cognition).

The Cognitive Process Dimension depicted in Figure 3 highlights different levels of cognitive processes. While simple cognitive process are identified (like remember and understand), our concern is with the create level. Figure 3 depicts "create" as the highest level of cognitive process. However, it is interesting to note that *creative* and *evaluate* are seen as distinct regions on the cognitive dimension, given their joint roles in many creative systems.

We shall examine how the creative process interactions with (or relies upon) the previous four levels of knowledge: *factual, conceptual, procedural* and *metacognitive*.

Cognitive Processes and "Create"

Before we look at the "create" level of Cognitive Processes Dimension, we note that the adjacent level of process is "evaluate". This would appear to highlight the close relationship between creation and evaluation. For example, at the metacognitive level of evaluation we see the "reflect" verb – with reflection often being seen as a precursor to creativity. However, this paper is focused on the differing levels of the "create" cognitive process.

Generate: Create Factual Outputs

While we may not frequently think of producing new facts as a creative challenge, we can see creativity as sometimes being involved - even when there is a known technique to help generate these facts. Let us consider the domain of prime numbers, whole natural numbers divisible only by themselves and 1. Prime numbers play an important role in cryptography and other domains. A non-creative process may simply list the known prime numbers. However, looking at the creative dimension we can see that "generating" a new prime number might be considered a creative task. Let us restrict the set of numbers even further to the set of Mersenne primes – that is, a prime number that is also a Mersenne number of the form ($M_n = 2^n - 1$). While this equation looks like it can "generate" arbitrary prime numbers, in fact most Mersenne numbers are not prime. The "Great Internet Mersenne Prime Search" project is devoted to discovering ever larger Mersenne prime numbers. Among the reasons for considering this to be a creative task is the enormity of the space of numbers and Mersenne numbers and the enormity of verifying that a given candidate is actually prime.

Assemble: Create Conceptual Outputs Creating new concepts might be achieved by combining previously existing concepts, by appropriately assembling a new construct using the lower factual level of knowledge. This could involve finding or creating new similarities between existing knowledge. Here the creation process is already known or relatively straightforward, with the focus being on the concepts and their creation. That is the "assembly" process is already known and is used to create the new knowledge.

Many creative systems appear to produce artefacts that introduce new concepts and facts, using systems that do not change while that artefact is being created. Even powerful systems like analogical reasoning and evolutionary computation typically create new concepts in an "assembly" like manner.

Design: Create Procedural Outputs The next level of creativity aims to design new procedures that might operate on existing or new facts. This level of creativity introduces additional flexibility and creative power, in that the range of possible outputs and artefacts is greatly increased upon the lower level.

Analogical reasoning, evolutionary computation and other approaches might be seen as involving metacognitive creation were they to reflect upon their own processes – and use this reflection to guide further progress (while evolutionary strategies take their progress into account through strategies like adaptive mutation and others, reactions do not (usually) take the form of metacognitive or reflective modifications to the creative process).

Create: Create Meta-Cognitive Outputs These often involve self-knowledge and reflection on that knowledge. The authors are not aware of any computational models addressing this level of computational creativity. Metacognitive and reflective processes may well encompass a Theory of Mind (ToM) as mentioned earlier. However, metacognitive aspects are generally not made explicit in most creative systems.

Levels of Computational Creativity

In this section we build on this joint perspective of knowledge/artefacts and processes. We begin by re-visiting computational creativity, but bearing in mind that creativity is also valued among thinking, processing students.

Creating Outputs that themselves Create Artefacts One significant feature of the generated RegEx is that it has a dynamic productive quality. The created product is itself, capable of generating products. In this case the created regular expression is at the lowest level of the Chomsky hierarchy, however a similar approach can in principle be adopted to generate automata at any level from the Chomsky hierarchy.

Interestingly, from a creativity perspective it is relatively straightforward to generate an output process that is at a more complex level that the input expression. That is an FSA can be easily transformed into a pushdown automaton by introducing an additional rule from a higher level automaton or by introducing higher level rules that overlap with the pre-existing grammar.

While there has been some discussion on the Turing Test and its potential use and adaptation for computational creativity (Boden, 2010; Pease *et al*, 2012), there have been surprisingly few references to Turing Machines in the various discussions on computational creativity.

What limits can we see on the artefacts that are produced by a computationally creative process? Similarly, what limits can we see in the creative processes generated by a creative system? Let us consider a creative system that outputs new and interesting Turing Machines. Earlier in this paper we saw a creative system that created a very simple Turing Machine (a regular expression). Is it possible to generate a creative Turing machine whose output could be (or at least include) a creative Turing Machine?

Turing Machine TM1 can be considered creative only if it generates an output string that was not produced by other machines in its inspiring set. Or alternatively, it produced the same output but did so using a different grammar. That is, either the language or the grammar must be different in some novel and useful way.

We now look at four levels of computationally creative system that arise from our focus on creative processes.

1. Direct Computational Creativity (DCC): In direct computational creativity the outputs (artefacts or processes) display the *novelty* and *quality* attributes associated with creativity. This category includes the majority of work in computational creativity where the (direct) output of the computational process is seen as creative. The directly created output might be an image, a poem, a piece of music, a recipe, or it might be a computational process such as a regular expression or an evolved program.

In terms of the search space metaphor, *direct* computational creativity searches through the space of novel and useful outputs.

2. Direct Self-Sustaining Creativity (DSC): In direct self-sustaining creativity, the outputs are added to the inspiring set and serve to drive subsequent creative episodes. Supporting this type of creativity involves two distinct factors. Firstly, the process must be capable of generating multiple creative artefacts and secondly the quality of the creative outputs must be adequately judged before inclusion in the inspiring set.

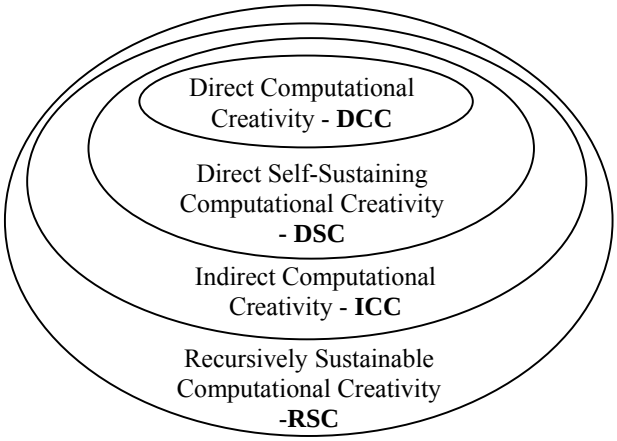

Figure 4: Levels and Limits of Computational Creativity

3. Indirect Computational Creativity (ICC): Indirect computational creativity outputs a creative process and that creative process is creative. That is, ICC outputs processes and those creative processes can be considered as computationally creative systems. We see this as a form of *indirect* computational creativity, where we attribute creativity to the created process.

We do not see these created processes as simple variants on some successful template – outputting a family of closely related creative models. But instead, the ICC should also itself display an ability to produce processes with the attributes of novelty and quality.

4. Recursively Sustainable Creativity (RSC): This is a further restriction on the previous level, where the created process itself creates processes at the level of RSC. This would appear to be a very challenging level of computational creativity, creating highly creative processes. RCS represents the most significant challenge for computational creativity arising from this discussion. It would appear that techniques like evolutionary and genetic programming are best suited to producing such creative models.

Conclusion

The search space metaphor pervades most work on computational creativity but appears to have led towards a divide, between a focus on creative artefacts and less of a focus on the creative *processes*. Two projects are briefly described to highlight some differences between artefact centred and process centred computational creativity. ImageBlender creates new images by combining two input images in complex mathematical transformation of those images. RegExEvolver takes just one regular expression as its input and creates new expressions that differ from their expressions, either in terms of the language it produces or in terms of the expression itself.

Kolmogorov complexity and other general purpose compression algorithms appear to offer very useful and widely applicable mechanisms for assessing the *quality* of output artefacts. In particular they offer a means of assessing the interestingness of creative outputs. In recent work it has been shown that interestingness as estimated by the fractal dimension has been closely correlated with judgements of artistic quality (Forsythe *et al*, 2010).

To help clarify the apparent friction between artefact and process centred creativity we turned to educational assessment – as this is another discipline that values creativity among its outputs. We suggest that the 2-dimensional model of Learning Objectives by Anderson and Krathwohl (2002) can offer guidance in comparing creative artefacts and processes. Among its advantages are its 2D matrix, elucidating different levels of attainment achieved along the "Cognitive Process Dimension" and the "Knowledge Dimension". We argue that these two dimensions can be seen as loosely analogous to the "Creative Process" and the "Creative Artefact" perspectives that are common to computational creativity. Four increasing levels of creative process were identified, described using the verbs; *generate, assemble, design* and *create*. Each of these four levels impacts on increasing levels of the knowledge (or artefact) dimension.

Finally, our focus on computationally creative processes allowed us to identify a four-level hierarchy of computational processes. We suggest that the majority of work on computational creativity is at the level of "Direct Computational Creativity" and arguably some work approaches the level of "Direct Self-Sustaining Computational Creativity". However, we also define two higher levels, the first being "Indirect Computational Creativity" that outputs processes that themselves are creative. The final level we call "Recursively Sustainable Computational Creativity" and only this highest level is capable of outputting creative processes that are akin in their creative potential to the originating process.

Acknowledgements

Some of the research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement 611383. We would like to thank John McDonald, Tom Naughton, Ronan Reilly and Stephen Brown for their contributions to the *ImageBlender* project and we would like to thank Amy Wall for her assistance with *RegExEvolver*.

References

Anderson, L.W.; Krathwohl, D.R.; Airasian, P.W.; Cruikshank, K.A.; Mayer, R.E.; Pintrich, P.R.; Raths, J.; and Wittrock, M.C. (eds) (2000). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives,*

Anderson, L. W. and Krathwohl, D. R., *et al* (Eds.) (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon, Boston, MA.

Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; and Krathwohl, D. R. (1956). *Taxonomy of educational objectives: the classification of educational goals; Handbook I: Cognitive Domain,* New York, Longmans, Green.

Boden, M.A. (1992). *The Creative Mind,* Abacus.

Boden, M.A. (2010). The Turing Test and Artistic Creativity, *Kybernetes*, 39 (3), pp 409-413.

Carson, S.H.; Peterson, J.B.; and Higgins, D.M. (2005) Reliability, validity, and factor structure of the creative achievement questionnaire, *Creativity Research Journal*, 17, pp 37–50.

Chomsky, N., (1959). On certain formal properties of grammars, *Information and Control*, 2(2), pp 137-167.

Cook, M.; Colton S.; Raad, A.; and Gow J (2013). Mechanic Miner: Reflection-Driven Game Mechanic Discovery and Level Design, *LNCS Vol.* 7835, pp 284-293.

Forsythe, A.; Nadal, M.; Sheehy, N.; Cela-Conde, C.J.; and Sawey, M. (2010). Predicting beauty: Fractal dimension and visualcomplexity in art, *British Journal of Psychology,* 102(1), pp 49-70.

Gardner, H. (1993). Creating Minds, Basic Books, NY.

Godefroid, P.; Levin, M.Y.; and Molnar, D. (2012). SAGE: Whitebox Fuzzing for Security Testing, *Communications of the ACM*, 55(3), pp 40-44.

Guilford, J.P. (1950). Creativity, *American Psychologist*, 5(9), pp 444-454.

Hankins, T. (1980). *Sir William Rowan Hamilton,* Johns Hopkins University Press.

Krathwohl, D.R.(2002). A Revision of Bloom's Taxonomy: An Overview, *Theory Into Practice*, 41(4), pp 212-218.

Luke, S. (2013). *Essentials of Metaheuristics*, 2nd Edn. Lulu, http://cs.gmu.edu/~sean/book/metaheuristics/

McGregor, S. (2007). Algorithmic Information Theory and Novelty Generation, *Proc. 4th Intl. Joint Workshop on Computational Creativity (IJWCC),* London, June.

Okada, T.; Yokochi, S.; Ishibashi, K.; and Ueda, K. (2009). Analogical modification in the creation of contemporary art, *Cognitive Systems Research*, 10, pp 189–203.

O'Donoghue, D. and Crean, B. (2002) Searching for Serendipitous Analogies, *ECAI - Workshop on Creative Systems*, Lyon, France.

O'Donoghue, D.P.; Bohan, A.; and Keane, M.T. (2006). Seeing things: Inventive reasoning with geometric analogies and topographic maps, *New Generation Computing,* 24(3), pp 267-288.

O'Donoghue, D.P. and Keane, M.T. (2012). A Creative Analogy Machine: Results and Challenges, *International Conference on Computational Creativity (ICCC),* UCD, Dublin, Ireland, pp 17-24.

Pease, A.; Colton, S.; Ramezani, R.; Charnley, J.; and Reed, K. (2013). A Discussion on Serendipity in Creative Systems, *International Conference on Computational Creativity (ICCC),* Sydney, Australia, pp 64-71.

Pitu, M.; Grijincu, D.; Li, P.; Saleem, A.; Monahan, R.; and O'Donoghue, D.P. (2013). Arís: Analogical Reasoning for reuse of Implementation & Specification, *AI for Formal Methods (AI4FM) Workshop,* Rennes France, 22 July.

Ritchie, G. (2012). A Closer look at Creativity as Search, *4th International Conference on Computational Creativity (ICCC),* UCD, Dublin, Ireland, pp 41-48.

Newell, A.; Shaw, J.G.; and Simon, H.A. (1963). The Process of Creative Thinking, in *Contemporary Approaches to Creative Thinking,* pp 63-119. New York: Atherton.

Stojanov, G. and Indurkhya, B. (2012). Perceptual Similarity and Analogy in Creativity and Cognitive Development, *1st International Workshop on Similarity and Analogy-based Methods in AI (SAMAI),* at ECAI, France.

Togelius, J.; Yannakakis, G.N.; Stanley, K.O.; and Cameron Browne C. (2011). Search-based Procedural Content Generation: A Taxonomy and Survey, *IEEE Transactions on Computational Intelligence and AI in Games* 3(3), pp 1-15.

Veale, T.; and O'Donoghue, D. (2000). Computation and Blending, *Cognitive Linguistics,* 11(3/4), pp 253-282.

Veale, T. (2012). *Exploding the Creativity Myth*, London: Bloomsbury Academic.

Wells, D. (1990). Are these the Most Beautiful? *The Mathematical Intelligencer,* 12(3), pp 37-41.

Wiggins, G.A. (2006). Searching for Computational Creativity, *New Generation Computing*, 24(3), pp 209–222.

Appendix

This appendix contains a small sample of the images created by ImageBlender.