Searching for Computational Creativity

Geraint A. WIGGINS

Centre for Cognition, Computation and Culture
Goldsmiths' College, University of London
New Cross, London SE14 6NW, UK

g.wiggins@gold.ac.uk

Received 15 December 2005

Abstract Boden's[1,2] philosophical account of creativity has been criticised on the grounds that it does not properly capture some aspects of creative situations[5]. Wiggins[13] has presented a formalisation of Boden's account, which allows such issues to be examined more precisely. We explore the relationship between traditional AI search methods and Boden's abstraction of creative behaviour, and revisit Bundy's argument in the context of that exploration.

Keywords Computational Creativity; Search; Complexity

§1 Introduction

In this paper, we discuss the relationship between traditional AI state space search and creative systems, as introduced by Boden[1] and formalised by Wiggins[13]. We then consider one of the arguments presented by Bundy[5], who proposes the need for complexity as part of creative value.

In Section 2, we review the standard AI state space search algorithm, Boden's model and Wiggins' formalisation of it. In Section 3, we compare traditional AI state space search with Boden's creative system model and suggest that the latter is a generalisation of the former in four important ways. In Section 4, we revisit Bundy's argument that complexity is a necessary feature of created artefacts, and argue that this is not necessarily the case; rather, we suggest that an artefact is more likely to be deemed creative by subjective observers if there is complexity in the conceptual space in which the artefact is found. We summarise and conclude in Section 5.

For the purposes of this paper, we define *Computational Creativity* as follows:

> "The study and support, through computational means and methods, of behaviour exhibited by natural and artificial systems, which would be deemed creative if exhibited by humans."

This "study and support" may, of course, include simulation. A *Creative System* is a system, natural or artificial, which exhibits such behaviour.

It is worth mentioning that, though this definition is sometimes decried as being "circular", it is not really so: rather, it aims to capture a concept or set of concepts which is difficult to define intensionally by means of explicit rules in terms of its extension into the real world of human behaviour.

§2 Background

2.1 AI Search and Problem Solving

AI search has been characterised in a simple and elegant agenda-based search algorithm[11].

The approach is usually to formulate one's computation as a *problem* to be *solved*. The search task is then to find a complete solution among a set of partial and complete solutions, expressed in a fixed, symbolic representation. The output of the algorithm is usually either the solution itself or the path through the search space which found it. The partial and complete solutions make up the *search space*; when viewed as a set of representations of the state of the search, or of the state of construction of a solution, rather than partial and complete solutions, the space is sometimes called the *state space*.

All the standard search algorithms (depth-first search, breadth-first search, best-first search and Algorithms A and $A^\star$) may be expressed in terms of the following components:

Representation allowing expression of partial and complete solutions;
Solution Detector allowing termination of the search when a solution is found;
Agenda keeping track of the states waiting to be examined in an ordered list;
Expansion Operator producing one or more new states from each existing one.

The general algorithm is then simple:

1. Apply the solution detector to the first item on the agenda. If it succeeds, output the first item and terminate; if it does not, continue.
2. Remove the first item from the agenda, and apply the expansion operator to it. Combine the new states so produced into the remainder of the agenda.
3. Go to step 1.

The key to producing different search algorithms is in the combination of new states with old ones, in step 2, above. For depth-first search, the new states are placed at the front of the agenda; for breadth-first, at the back. For best-first search and Algorithms A and $A^\star$, the agenda is sorted according to a *heuristic*, or "rule of thumb".

The heuristic value is produced by a function applicable to states which measures how good a (partial) solution is for best-first, or how costly it is to produce for Algorithms A and $A^\star$. The latter two use a heuristic which both accounts for the cost of producing the current intermediate state and an estimate of the cost of reaching a finishing state from it. In Algorithm $A^\star$, the heuristic is *admissible*, which means that its value at any state is known to be bounded above by the actual cost of reaching the least expensive solution reachable from that state; in these circumstances, $A^\star$ search approaches optimality, in the sense that it visits the fewest number of states possible *en route* to the solution, as the estimate approaches that upper bound.

For completeness, we mention iterative deepening search, which is a compromise between depth- and breadth-first search favouring space-complexity at the expense of time-complexity, and $IDA^\star$, which does the same in the context of Algorithm $A^\star$. These algorithms are not expressible in the framework above, and are not helpful to our argument here.

Before proceeding, it is worth pausing to note that the reason for doing all this is to solve problems whose state space is sufficiently large as to be, first, unenumerable by conventional exhaustive methods, and, second, formally unpredictable as to the existence and location of the solutions sought—if this were not so, we could simply write analytical algorithms to solve those problems. This point will be crucial below, when we consider the relationship between complexity and value.

Minsky[8] suggests that

> "We can program a computer to solve any problem by trial and error, without knowing how to solve it in advance, provided only that we have a way to recognize when the problem is solved."[8,4]

and it is here that the heuristic really comes into play. Minsky's claim is, strictly, true, but its implication—that "program[ming] a computer to solve any problem" is the same as actually obtaining the solution to that problem—is in general false, because we need the additional assurance that the solution we seek will be found in finite time by our trial and error strategy. This we cannot have, in general, because of the Halting Problem[12].

This, then, is one of the reasons why search algorithms are important in the study of AI: they allow us to solve problems which are inscrutable to analytical methods, and even to give partial solutions for problems which are known to be uncomputable when considered in their entirety.

As stated above, the key to formulating different search algorithms in this framework is the choice of representation, agenda manipulation and heuristic. For example, a significant difference in algorithmic power can be obtained by using a representation where the *path* to a (partial) solution through the search space is explicitly recorded in the agenda, as opposed to a representation where the (partial) solution only is represented, without any of its construction history. The difference is that in the former, meta-information, about the construction of the (partial) solution may be used to guide the construction of future agenda members; in the latter it may not. Of course, this is not useful in all cases, but in some it may be—for example, in avoiding cycles in the solution construction which are not manifest as cycles in the search space, and are therefore undetectable at that level.

A key restriction of the framework is that it does not admit construction of new (partial) solutions from more than one old one: the notion is of a stepwise construction of each solution, independently. This is because all versions of the algorithm operate only on the first member of the agenda. Consequently, the framework is unable to capture search methods which work on the basis of *combining* agenda members, such as metaphorical reasoning or evolutionary algorithms.

2.2 Boden's Model of Creativity

Boden's[1] model of creativity revolves around the notion of a *conceptual space* and its exploration by creative agents. The conceptual space is a set of artefacts (in Boden's terms, *concepts*) which are in some quasi-syntactic sense deemed to be acceptable as examples of whatever is being created[1]. Implicitly, the conceptual space may include partially defined artefacts too. *Exploratory creativity* is the process of exploring a given conceptual space; *transformational creativity* is the process of changing the rules which delimit the conceptual space. Boden[2] also makes an important distinction between mere membership of a conceptual space and the *value* of a member of the space, which is extrinsically defined, but not precisely.

Bundy[5] and Buchanan[4] both join Boden in citing reflection, and hence meta-level reasoning, as a requirement for "real" or "significant" creativity (though the definition of such creativity is so far left imprecise). For completeness, we mention here that there are other views: Ritchie[9], for example, presents a completely different account of what is going on in "transformational" creativity, in which the notion of transformation is not so clearly present. However, since the current paper is primarily focused on the relationship between Boden's model and search, we defer discussion of Ritchie's alternative to future work.

2.3 Wiggins' Formalisation of Boden's Model

The central plank of the formalism presented by Wiggins[13] is that an exploratory creative system, in Boden's[1] terms, may be abstractly represented by a septuple, thus:

$$\langle \mathcal{U}, \mathcal{L}, [\![.]\!], \langle\!\langle ., ., . \rangle\!\rangle, \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle.$$

The symbols here are defined in Table 1. The function of each is briefly explained below (Wiggins[13] gives more detail).

$\mathcal{U}$ is the (abstract) set of all possible partial and complete artefacts describable in the creative system being modelled. $\mathcal{R}$ is a set of rules, expressed using the language $\mathcal{L}$, which select an "acceptable" or "relevant" subset of $\mathcal{U}$ which corresponds with Boden's[1] conceptual space. In Wiggins' formulation, selection is *permissive* in the sense that it admits partial artefacts, even some of whose completions may eventually turn out not to be admitted. So applying a selector function generated from $\mathcal{R}$ by $[\![.]\!]$ and a suitable real comparator (*e.g.,* 0.5) gives Wiggins' formalisation of Boden's conceptual space:

$$\{c \mid c \in \mathcal{U} \wedge [\![\mathcal{R}]\!](c) \geq 0.5\}.$$

Table 1 The symbols used in Wiggins' description of Boden's exploratory-creative system.

$\mathcal{U}$	a universe of possible concepts (artefacts), both partial and complete
$\mathcal{L}$	a language in which to express concepts (artefacts) and rules
$[\![.]\!]$	a function generator, which maps a subset of $\mathcal{L}$ to a function which associates elements of $\mathcal{U}$ with a real number in [0,1]
$\langle\!\langle ., ., . \rangle\!\rangle$	a function generator, which maps three subsets of $\mathcal{L}$ to a function that generates a new sequence of elements of $\mathcal{U}$ from an existing one
$\mathcal{R}$	a subset of $\mathcal{L}$
$\mathcal{T}$	a subset of $\mathcal{L}$
$\mathcal{E}$	a subset of $\mathcal{L}$

The ruleset, $\mathcal{R}$, then, defines what it is to be an artefact of the kind we are interested in creating: a piece of music, a joke, and so on. (Alternatively, the output of $[\![\mathcal{R}]\!]$ might be used directly in a fuzzy selector; we postpone discussion of this for now.)

$\mathcal{T}$ is a set of rules which, when interpreted, perhaps along with those in $\mathcal{R}$ and $\mathcal{E}$, by $\langle\!\langle ., ., . \rangle\!\rangle$, describe the behaviour of a creative agent as it traverses the conceptual space from known artefacts to unknown ones (much as the standard AI search framework described in §2.1), and possibly back again. The first argument of $\langle\!\langle ., ., . \rangle\!\rangle$ takes a concept/artefact-definition ruleset, such as $\mathcal{R}$, above, and the second a rule set such as $\mathcal{T}$, which is the specification of the traversal strategy. The third argument is $\mathcal{E}$, the rules by which *value* is attributed to a created artefact, new or otherwise (see below). $\mathcal{R}$ and $\mathcal{T}$ are included so that it is possible for $\mathcal{T}$ to include reasoning about them, but this is not a requirement; thus, $\mathcal{T}$ can in principle generate artefacts which do not conform to the rules of $\mathcal{R}$ and this can be used to trigger subsequent reasoning and reflection about the creative system under simulation[13]. There is no explicit equivalent of $\mathcal{T}$ in Boden's writing, though it is implicitly present at all times. To distinguish between transformation of $\mathcal{R}$ and transformation of $\mathcal{T}$ we write "$\mathcal{R}$-transformation" and "$\mathcal{T}$-transformation".

$\mathcal{E}$ is a set of rules which define the evaluation of the creative outputs resulting from the agent's activity, appropriately contextualised. The formalism does not specify what this context is; it might be the subjective judgement of the creating agent, or the subjective combined judgements of other agents, or comparison with some objective measure. $\mathcal{E}$ allows us to express the notion of value proposed by Boden[2]. (For completeness, we mention that we would expect $\mathcal{E}$ to be amenable to transformation, also, in particular ways, especially if this theory were applied in the context of a multi-agent system. However, for the moment we leave the interesting question of how usefully to formalise $\mathcal{E}$-transformation to future work.)

Wiggins[13] gives examples to elucidate how the framework may be used, and shows how *transformational* creativity can be cast as exploratory creativity at the meta-level, where the conceptual space is the set of possible rule sets, generated by a given language, as informally suggested by Bundy[5].

A substantive difference between Boden's formulation and that of Wiggins is the addition of the rule set, $\mathcal{T}$, which describes the actual behaviour of a creative agent as it goes about its business: Boden is not concerned with this level of detail. The difference gives Wiggins' formulation more power to describe the behaviour of *implemented* creative systems. Thus, it may be compared in detail with existing similar methods, such as those of AI state space search. Further, the introduction of $\mathcal{T}$, as an explicit component, admits a new kind of transformational creativity, in which an agent modifies its own behaviour by reflective reasoning. This may be appropriate for the description of behaviours exhibited by Lenat's AM, for example[10].

§3 The Relationship between the Creative System Framework and AI Search

There are clearly strong similarities between the more traditional AI search and Boden's ideas: in particular, the conceptual space, thought of from an AI perspective, seems to be very like the traditional state space. In this section, we explore these issues in detail. This exploration will lead us to revisit Bundy's[5] ideas in Section 4.

3.1 Representation and Conceptualisation

Implicit in any traditional, symbolic approach to AI is the need for a representation, usually with a semantics which is in some sense compositional (that is, the meaning of the parts of the representation are somehow composed into the meaning of the whole). A crucial point is that the syntax of the representation is deemed to be in one-to-one correspondence with its semantics. Such a representation is explicit both in the standard AI search algorithm and in Wiggins' formalisation of Boden's work. It is strongly implicit in Boden's original writing. As many AI researchers (and indeed computer scientists in general) have found out the hard way, representation and algorithm (search or otherwise) are tightly bound together, and a simple change of representation can sometimes reduce a problematically time-complex solution to a perfectly tractable one.

Here, we consider "representation" not in the sense of *data* representation, but of *knowledge* representation, in the traditional AI style. So we are not concerned with issues like whether something is a pointer or an integer, but primarily with what the symbols mean with respect to an extrinsic interpretation, and what can be thence inferred from them. In this way of thinking, as we traverse a state space, we are constructing a formal description of the solution we seek (or perhaps we are constructing a description of how to construct it—the difference is immaterial here). In this context, it is perhaps worth mentioning one particular aspect of creative behaviour which might be thought to be excluded in this model: logical (or quasi-logical) connections made through ambiguity in reference—which is excluded from the representation because of the one-to-one mapping between syntax and semantics. This model certainly does not attempt to deny the value of such linguistic ambiguity. However, neither does it attempt to model language; rather, the aim is to model thought at a post-linguistic level, notwithstanding the (usual) use of linguistic symbols to do so. Therefore, any such ambiguity would properly be modelled as part of $\mathcal{R}$, $\mathcal{T}$ and/or $\mathcal{E}$.

Beyond this there is little to say: the techniques and precepts that apply to traditional AI knowledge representation apply wholesale in the context of implementing Boden's ideas. We can therefore sit comfortably at the current level of detail, which simplifies matters.

In traditional AI search, the representation defines, in practice at least, the universe of possibilities to be exactly that contained by the search space. Boden's model, however, implicitly but necessarily allows possibilities to exist outside the conceptual space, for, otherwise, transformational creativity of the conceptual space would be a vacuous operation. So the representation used in Wiggins' universe, $\mathcal{U}$, is more general than that used to do problem-solving in AI: it is not restricted to the problem at hand, but is capable of representing more general spaces, some of which are, in the sense of conceptual space membership defined by $\mathcal{R}$, not relevant to (the current formulation of) the current "problem", but which may become relevant should transformation be applied to $\mathcal{R}$.

Therefore, to implement traditional AI search in terms of Wiggins' framework, $\mathcal{R}$ would be defined so as to encode the syntactic rules of the representation, and $\mathcal{T}$ would be defined to generate all and only the search space. (Of course, this renders $\mathcal{R}$ unnecessary.)

What does this mean in terms of the creative system framework? In state space search, the syntax of the representation is assumed to be externally defined, whereas in the framework, it is explicitly internally defined (which is a requirement for it to be amenable to transformation within the system). So creative system framework system is, at least in principle, capable of inferring from and operating on an internal description of its representation. In other words, it is capable of reflection, and such *reflection* is one of the requirements for creativity suggested independently by Bundy[5] and Buchanan[4]. When it does so, it may exhibit transformational creativity, in Boden's $\mathcal{R}$-transformational sense, of the conceptual space, or in Wiggins' $\mathcal{T}$-transformational sense, of the traversal rules, or both. The search framework is not capable of such transformation, because its rules are not available for reasoning an inference *within the search.*

Linked with this point is one aspect of the relationship between the termination detector of state space search and the evaluation rules, $\mathcal{E}$, of the creative system framework. In their respective contexts, these two operators are semantic interpreters, in the sense that the former can detect a representation of a solution to a problem and the latter can interpret the value of the representation of a created artefact with respect to a particular value system. The binary output of the solution detector can be trivially implemented by applying a threshold to the output of the same interpreter, $[\![.]\!]$, used to produce the value (presumably with more detail than just binary values), as was done for the syntax detector, $[\![\mathcal{R}]\!]$, above.

So, in both of these aspects, the creative system framework is more general than the state space formulation, in the formal sense that the former is capable of implementing the latter and more, but the converse is not true: the key difference being the internal description of the system's representation. In this sense, at least, therefore, the creative system framework is not "just" AI search; rather, it is a very significantly more expressive framework, because it is capable of reflection and of changing itself as a result of that reflection.

3.2 Searching the Spaces

In the state space framework, the computation is characterised as operation of an agenda, with an expansion operator, which is capable of generating new states from existing ones. The framework is elegantly simple in being able to express several different algorithms simply in terms of the operation of an agenda; the creative system framework is necessarily more complicated, though not much. In the same sense as above, it turns out to be a fairly straightforward generalisation of AI search; the generalisation, however, allows the description of significantly more algorithms.

The key features of the creative system traversal mechanism, $\langle\langle \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle\rangle$, are as follows. First, it incorporates a process for exploring (or enumerating) the space of possibilities. Second, there are rules which may be applied to check the validity of the (partial) artefacts generated, by which we mean to check whether they really are artefacts of the kind to be considered ($\mathcal{R}$). Third, there are rules which may be applied to (partial) artefacts to test their value ($\mathcal{E}$). The traversal rules may or may not refer to $\mathcal{R}$ and/or $\mathcal{E}$ in their operation, in much the same way as the depth- and breadth-first search algorithms do not resort to a heuristic, but best-first does; also, the rules of $\mathcal{T}$ may themselves encode heuristic decisions which are different from those made on the basis of $\mathcal{R}$ and $\mathcal{E}$. For simplicity and clarity here, we suppose that there is no overlap between the rules of $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{E}$, in the sense that no rule or combination of rules in any one of the sets makes a decision for the same set of (semantic) reasons as a rule or combination of rules in any other. In computational terms, of course, this is arbitrarily hard to enforce, given that, for reasons of uniformity, we need them all to expressible in the same language; however, it will help keep the current argument clear.

In the next section, then, we consider the relationship between the generation of search agenda members (step 2 in the framework of Section 2.1) and the generation of (partial) artefacts by the traversal mechanism of the creative system framework. We separate out aspects of heuristic control, which are covered in Section 3.5.

3.3 Generation of states and artefacts

Since the syntax of the representation is not available as data to the state space search algorithm, the expansion operator must be assumed always to generate members of the search space; to do otherwise would be deemed an error in the program, and thence treated as external to the "world" of the problem. In the creative system framework, the situation is quite the reverse: if an artefact is created which does not conform to the rules of $\mathcal{R}$, then it is simply a new thing which does not conform to expectation, rather than an incorrect thing. Wiggins[13] names this situation *aberration*, and notes that there are conditions in which it may be *advantageous* to the goal of creation: aberration which produces valued artefacts (*perfect* or *productive* aberration, in Wiggins' terms) may constitute a trigger for transformational creativity. However, for the current purpose, we emphasise that the actual aberration is firmly at the (exploratory) object level: a construct has simply been constructed, and no rules or representations have been changed.

Of course, we must also note that there is an underlying syntax in the creative system framework. The key point is that the syntax of that representation is not in one-to-one correspondence with the semantics of *one particular problem* whose solution is sought.

3.4 Traversal Strategies

The most elegant aspect of the state space formulation lies in the simplicity of the interaction between the expansion operator and the agenda to produce the various search algorithms. In the creative system framework, this simplicity and transparency is to some extent sacrificed, as often happens when something is made more powerful.

In particular, both formulations maintain an ordered list, which serves as an agenda—indeed, it is hard to see how they could be characterised without one. But there is a key difference: the creative system framework is not limited to operating only on the single, first element of the agenda. This is precisely why it is less transparent than the state space formulation: we can no longer naturally express its behaviour in such simple, elegant terms. In the creative system framework, the elements of the agenda are accessible to the traversal operation in whatever way is required, and, importantly, not necessarily just one at a time. The agenda is ordered simply so that the traversal operation has a record, in its own ordering, of what has been produced when, should this be necessary. On a practical level, this is needed to admit modelling by the framework of such search-related approaches as genetic algorithms and conceptual blending, where properties of two or more existing artefacts are recombined to create another. On a philosophical level, it is necessary for a properly transparent model of what is going on, in the terms of the original (Boden) formulation.

To unpack this last claim a bit, consider the fact that any sequence of artefact-building steps performed by a so-called "creative" system may be modelled as a traditional state space search, where the representation is the language of the creative system, and the states are the states of the creative system as a whole (as opposed to the partial and complete artefacts being created). The simple fact is that any closed-world representation can be formulated and explored in this way, which is the point behind Minsky's[8] claim, quoted in Section 2.1 above. Now, while it is not currently possible to engineer a symbolic, open-world representation (the CYC project[7] is one attempt to do so), the step from this particular application of the state space formulation to the creative system framework begins to make explicit the need for (and utility of) a world which is at least bigger than the problem to be solved. Not only does this facilitate our understanding, but also it allows us to specify some formal requirements on creative systems[13] and to study them more specifically, so long as we are careful not to be misled by artefacts of the formulation itself.

Notwithstanding the possibility of re-representing everything as state space search, we now compare the state space formulation with the creative system framework on the most level playing field available: that where the conceptual space is viewed as correspondent with the search space, as in Section 3.1, above. We choose this level of comparison, in particular, also because Boden[1] herself comments on this particular similarity, and criticism of her framework, such as that of Bundy[5], is often couched in its terms.

It is almost trivial to note that the strategy of taking the first member of the agenda, expanding it, and then combining the results with the rest of the agenda, as used in state space search, is a specialisation of the more general creative system strategy: rather than use many members of the agenda, use one; rather than using any member, use the first one; exhaustively generate new artefacts from the first one; append or prepend the results to the agenda. This gives us depth- and breadth-first directly, while the heuristic methods must wait for the next section. Note here, though, that, where those two algorithms are concerned, the creative system framework is, as above, more general than the state space search formulation, in the sense that the latter can be expressed completely in terms of the former, but not the other way around.

3.5 Heuristics, Traversal Strategies and Value

The final comparison to be made between the two frameworks is in the heuristics, on the one hand, and traversal strategy and value on the other. Here things become rather less clear-cut, not least because the notion of value is relatively ill-defined[2].

In the state space formulation, the heuristic is used to place an order on the expansion of states in the search. This is easy to understand, because there is always one focus of attention in the search space: the next node to be expanded (*viz.* the first on the agenda). On the other hand, the selection of nodes to be used for traversing the space of the creative system framework is not a blind, predefined process, but needs to be defined explicitly by the rules of $\mathcal{T}$. Of course, nothing precludes, for example, a sorting régime, in which one takes the first n elements of the agenda and uses them, but this may not be entirely general. In particular, heuristics which are dependent on the relationships between two or more existing artefacts are inexpressible in the state space formulation. However, it is easy to see from the above that the best-first algorithms (including A and $A^\star$) can be implemented within the framework, with (at least in the simplest case) value being the heuristic. So, again, the creative system framework is more general than the state space search formulation in the sense that the former can simulate the latter, but not the other way around.

3.6 The Status of the Creative System Framework

The conclusion we draw from this argument is that the creative system framework is both a generalisation and a clarification of the state space search formulation, in a number of crucial ways:

1. it opens the closed world of the state space formulation, with respect to the problem being solved or the artefact being created, in as far as a symbolic representation can;
2. it admits expansion steps which simulate the combination of existing ideas, as well as their development;
3. it makes explicit the difference between heuristics which evaluate the quality of a (partial) solution and heuristics which can be used to find a solution efficiently; and
4. it broadens the notion of value to one of comparison, as opposed to the Boolean solution test, where only “correct and complete” answers are valued.

Again, it is to be noted that any of these features can be implemented by applying state space search at a meta-level, as Minsky says; this, however, is not a useful comparison in this context, because to do this is to make a comparison at between inappropriate levels of representation. Indeed, to do one's search at the meta-level to cover a significant part of the route towards a creative system anyway.

Having laid out this stall, we now proceed to consider Bundy's[5] argument concerning some of the details of Boden's[1] proposal.

§4 Complexity and Value

Bundy[5] presents (among other things) a proposal that complexity is perhaps a requirement for value in a creative system. He backs up this proposal with arguments from the field of mathematical reasoning. The substance of the argument is that for something to be creative, as opposed to merely novel, some kind of complexity needs to be involved. This argument had been presented elsewhere in terms of *aesthetic* value[6], but not in terms of *scientific* value.

The difficult with this argument (as usual) is the ineffability of the notion of "creativity", which Bundy does not explicitly define. Since the publication in question, an unwritten consensus seems to have developed in the field that a measure of creativity needs at least two dimensions: *novelty* and *value*; both of these are context dependent. Given this, it seems likely that we should interpret Bundy's paper as not being about the relationship between "creativity" and "novelty", but being about *value*.

Given this, the suggestion that *complexity* is a contributory factor to scientific value is not a straightforward one. While it is certainly true that large and complex scientific theories can be greatly valued, it is generally the simple and elegant solution that is valued the most; a complex solution is not so much usually valued for itself as for the difficulty of the problem it solves—and, in particular, a complex solution to a simple problem is generally valued less than a simple one, and even rejected outright!

Bundy's suggestion is based on selections from Boden's book, and his interpretation differs from clarifications Boden herself subsequently issued[3]. The exact argument revolves around the question of whether exploratory creativity is "really" creative or not, and it is here that a potential confusion arises. There is general agreement (including Boden and Bundy) that mere automatic enumeration of artefacts, complex or otherwise, is not creative. However, we suggest that the ability to *recognise* a valued artefact which has been thus created *is* indeed part of the faculties of a creative agent—for example, some accepted and famous human artists work by the *selection* of randomly or chaotically generated forms.

Furthermore, enumeration of a conceptual space, like the enumeration of a state space, is just as subject to the Halting Problem as any other algorithm (see Section 2.1): even if we are sure that valued solutions exist, we cannot be sure that we will ever find them. So the fact that a space is *theoretically* enumerable does not mean that it is *practically* enumerable, and this undermines the argument that the *exploration* of a conceptual space is necessarily routine, because if we cannot give a terminating effective procedure to enumerate a space then we cannot guarantee to find the solutions which exist in it.

Furthermore—and here is where complexity *does* matter—the reason for formulating problems as search problems, as Minsky implies, is generally that one cannot predict the solution of complex tasks in complex worlds. So, in a real usage of this technology, it may well be the case that, while we know some of the properties of our solutions/artefacts, we do not have any clear idea of what they actually are as wholes; this is especially true in the more open world of the creative system framework.

Therefore, we argue, it is perfectly possible for an exploratorily creative system to generate real novelty (because it can produce results which are unpredictable) and in some circumstances for those results to be valued, *even if it works by brute enumeration*, and for the system to be capable of judging the value of the artefact for itself. These are the criteria for "creativity" in humans—why, objectively, should machine creativity be different?

Now, note that a human artist who produces valued outputs by exhaustive enumeration is generally less likely to be heralded as an important artist than an artist who is capable of using his or her own heuristics to arrive directly at a valued artefact. This is equally true of a creative computer system: a system which is capable of using heuristics to navigate its conceptual space directly to a valued solution is likely to be deemed more successful than one which is not.

In summary, Bundy's solution to the difference between, for example, prosaic prose and creative, exciting prose is the addition of complexity. We argue that complexity is just one facet of value, which may apply in some circumstances, but not in others (*i.e.,* it is context-dependent), and, more importantly, we argue that such a specific criterion is not necessary to justify the exploration of a conceptual space as "creativity", so long as we have a notion of value and it is available to the creative system. The complexity, then, appears not necessarily in the created artefact, as Bundy seems to suggest, but in the conceptual space which contains it: the more complex the space is, the less predictable its contents and their locations are, so the more value we, as observers, are likely to place on them when—if—they are found.

§5 Conclusion

We have addressed the status of exploratory creativity, as defined by Boden, and considered the question of whether it is reasonable to call it "creativity" at all. We have argued that creative systems, as characterised by Wiggins' formalisation of Boden's work, are more than simply state space search engines, in particular in that they work in a more open world and are capable of more complex search strategies involving combinations of existing partial solutions.

We have also discussed Bundy's proposal that complexity is necessary for behaviour to be creative, and suggested that such complexity may lie in the search space, and not only in the created artefact.

Given this, and a sufficiently complex conceptual space, we have argued that enumeration of that conceptual space, especially when guided by heuristics, can be a valid simulation of human creativity, and is therefore a candidate for the attribution of machine creativity.

Acknowledgment We gratefully acknowledge the advice of several anonymous reviewers, of Simon Colton, Graeme Ritchie and the other participants in the IJCAI'05 Workshop on Computational Creativity, and of our treasured colleagues in the ISMS Group at Goldsmiths College, University of London.

Bibliography

1) M. Boden. *The Creative Mind*. Abacus, 1990.

2) M. Boden. Creativity and artificial intelligence. *Journal of Artificial Intelligence*, 1998.

3) M. Boden. Evaluating creativity. Unpublished Keynote Talk at the AISB Convention, Edinburgh, 1999.

4) B. Buchanan. Creativity at the meta-level. *AI Magazine*, 2000. Reprint of keynote lecture to AAAI-2000.

5) A. Bundy. What is the difference between real creativity and mere novelty? *Behavioural and Brain Sciences*, 1994.

6) P. Machado, A. Dias, and A. Cardoso. Learning to colour greyscale images. *AISB Journal*, 1(2):209–219, 2002.

7) J. Masters and Z. Güngördü. Structured knowledge source integration: A progress report. In *Integration of Knowledge Intensive Multiagent Systems*, Cambridge, MA, 2003.

8) M. Minsky. The puzzle principle. In M. Minsky, editor, *The Society of Mind*. Simon and Schuster, New York, 1986.

9) G. D. Ritchie. Assessing creativity. In G. A. Wiggins, editor, *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*. AISB, 2001.

10) G. D. Ritchie and F. K. Hanna. AM: A case study in AI methodology. *Artificial Intelligence*, 23:249–268, 1984.

11) S. Russell and P. Norvig. *Artificial Intelligence – a modern approach*. Prentice Hall, New Jersey, 1995.

12) A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.

13) G. A. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems*, 2006. To appear.