Computer Models of Creativity

Margaret A. Boden

■ *Creativity isn't magical. It's an aspect of normal human intelligence, not a special faculty granted to a tiny elite. There are three forms: combinational, exploratory, and transformational. All three can be modeled by AI—in some cases, with impressive results. AI techniques underlie various types of computer art. Whether computers could "really" be creative isn't a scientific question but a philosophical one, to which there's no clear answer. But we do have the beginnings of a scientific understanding of creativity.*

Creativity is a marvel of the human mind, and an obvious goal for AI workers. Indeed, the proposal that led to the famous 1956 Dartmouth Summer School often remembered as the time of AI's birth mentioned "creativity," "invention," and "discovery" as key aims for the newly named discipline (McCarthy et al. 1955, 45, 49*ff*.). And, 50 years later, Herb Simon—in an e-mail discussion between AAAI Fellows (quoted in Boden 2006, 1101)—cited a paper on creativity (Simon 1995) in answer to the challenge of whether AI is a science, as opposed to "mere" engineering.

But if its status as an AI goal is obvious, its credibility as a potential AI achievement is not. Many people, including many otherwise hard-headed scientists, doubt—or even deny outright—the possibility of a computer's ever being creative.

Sometimes, such people are saying that, *irrespective of its performance* (which might even match superlative human examples), no computer could "really" be creative: the creativity lies entirely in the programmer. That's a philosophical question that needn't detain us here (but see the final section, below).

More to the point, these people typically claim that a computer simply could not generate *apparently* creative performance. That's a factual claim—which, in effect, dismisses AI research on creativity as a waste of time.

However, it is mistaken: AI scientists working in this area aren't doomed to disappointment. It doesn't follow that they will ever, in practice, be able to engineer a new Shakespeare or a neo-Mozart (although the latter goal has been approached more nearly than most people imagine). But lesser examples of AI creativity already abound. And, crucially, they help us to understand how human creativity is possible.

What Is Creativity?

Creativity can be defined as the ability to generate novel, and valuable, ideas. *Valuable*, here, has many meanings: interesting, useful, beautiful, simple, richly complex, and so on. *Ideas* covers many meanings too: not only ideas as such (concepts, theories, interpretations, stories), but also artifacts such as graphic images, sculptures, houses, and jet engines. Computer models have been designed to generate ideas in all these areas and more (Boden 2004).

As for *novel*, that has two importantly different meanings: psychological and historical. A psychological novelty, or P-creative idea, is one that's new *to the person who generated it*. It doesn't matter how many times, if any, other people have had that idea before. A historical novelty, or H-creative idea, is one that is P-creative *and* has never occurred in history before.

So what we need to explain, here, is P-creativity—which includes H-creativity but also covers more mundane examples. And our explanation must fit with the fact that creativity isn't a special faculty, possessed only by a tiny Romantic elite. Rather, it's a feature of human intelligence in general. Every time someone makes a witty remark, sings a new song to his or her sleepy baby, or even appreciates the topical political cartoons in the daily newspaper, that person is relying on processes of creative thought that are available to all of us.

To be sure, some people seem to be better at it than others. Some newspaper cartoonists have an especially good eye, and brain, for the delectable absurdities of our political masters. And a few people come up with highly valued H-creative ideas over and over again. Alan Turing is one example (he did revolutionary work in mathematics, computer science, cryptography, and theoretical biology [Boden 2006, 3.v.b–d, 4.i–ii, 15.iv]). But some people are better at tennis, too. To understand how Wimbledon champions manage to do what they do, one must first understand how Jo Bloggs can do what he does at the municipal tennis courts. P-creativity, whether historically novel or not, is therefore what we must focus on.

Computer models sometimes aim for, and even achieve, H-creativity. For example, a quarter century ago, an AI program designed a three-dimensional silicon chip that was awarded a patent—which requires that the invention must not be "obvious to a person skilled in the art" (Lenat 1983). And the AARON program (mentioned below) that generates beautifully colored drawings is described by its human originator as a "world-class" colorist. So it's presumably H-creative—and it's certainly capable of coming up with color schemes that he himself admits he wouldn't have had the courage to use.

Often, however, computer models aim merely for P-creativity. Examples discussed below include drawing scientific generalizations that were first discovered centuries ago (Langley et al. 1987), or generating music of a type composed by long-dead musicians (Cope 2001, 2006)

Even P-creativity in computers need not match all the previous achievements of human beings. Years ago, in the early days of AI, Seymour Papert used to warn AI researchers, and their sceptical critics, against "the superhuman human fallacy." That is, we shouldn't say that AI has failed simply because it can't match the heights of human intelligence. (After all, most of us can't do that either.) We should try to understand mundane thinking first, and worry about the exceptional cases only much later. His warning applies to AI work on creativity, too. If AI cannot simulate the rich creativity of Shakespeare and Shostakovich, it doesn't follow that it can teach us nothing about the sorts of processes that go on in human minds—including theirs—when people think new thoughts.

Creativity without Magic

How is creativity possible? In other words, how is it possible for someone—or, for that matter, a computer program—to produce new ideas?

At first blush, this sounds like magic: literally, producing something out of nothing. Stage magicians seem to do that, when they show us rabbits coming out of hats. But of course it's not really magic at all: members of the Magic Circle know how it's done. In the case of creativity, the psychologist—and the AI scientist—need to know how it's done if there's to be any hope of modeling it on a computer.

If we look carefully at the many examples of human creativity that surround us, we can see that there are three different ways in which creativity happens. Novel ideas may be produced by combination, by exploration, or by transformation (Boden 2004).

Combinational creativity produces unfamiliar combinations of familiar ideas, and it works by making associations between ideas that were previously only indirectly linked. Examples include many cases of poetic imagery, collage in visual art, and mimicry of cuckoo song in a classical symphony. Analogy is a form of combinational creativity that exploits shared conceptual structure and is widely used in science as well as art. (Think of William Harvey's description of the heart as a pump, or of the Bohr-Rutherford solar system model of the atom.)

It is combinational creativity that is usually mentioned in definitions of "creativity" and that (almost always) is studied by experimental psychologists specializing in creativity. But the other two types are important too.

Exploratory creativity rests on some culturally accepted style of thinking, or "conceptual space." This may be a theory of chemical molecules, a style of painting or music, or a particular national cuisine. The space is defined (and constrained) by a set of generative rules. Usually, these rules are largely, or even wholly, implicit. Every structure produced by following them will fit the style concerned, just as any word string generated by English syntax will be a gramatically acceptable English sentence.

(Style-defining rules should not be confused with the associative rules that underlie combinational creativity. It's true that associative rules generate—that is, produce—combinations. But they do this in a very different way from grammarlike rules. It is the latter type that are normally called "generative rules" by AI scientists.)

In exploratory creativity, the person moves through the space, exploring it to find out what's there (including previously unvisited locations)—and, in the most interesting cases, to discover both the potential and the limits of the space in question. These are the "most interesting" cases because they may lead on to the third form of creativity, which can be the most surprising of all.

In transformational creativity, the space or style itself is transformed by altering (or dropping) one or more of its defining dimensions. As a result, ideas can now be generated that simply *could not* have been generated before the change. For instance, if all organic molecules are basically strings of carbon atoms, then benzene can't be a ring structure. In suggesting that this is indeed what benzene is, the chemist Friedrich von Kekule had to transform the constraint *string* (open curve) into that of ring (closed curve). This stylistic transformation made way for the entire space of aromatic chemistry, which chemists would explore [sic] for many years.

The more stylistically fundamental the altered constraint, the more surprising—even shocking—the new ideas will be. It may take many years for people to grow accustomed to the new space and to become adept at producing or recognizing the ideas that it makes possible. The history of science, and of art too, offers many sad examples of people ignored, even despised, in their lifetimes whose ideas were later recognized as hugely valuable. (Think of Ignaz Semmelweiss and Vincent van Gogh, for instance. The one was reviled for saying that puerperal fever could be prevented if doctors washed their hands, and went mad as a result; the other sold only one painting in his lifetime.)

Transformational creativity is the "sexiest" of the three types, because it can give rise to ideas that are not only new but fundamentally different from any that went before. As such, they are often highly counterintuitive. (It's sometimes said that transformation is exploration on a metalevel, so that there's no real distinction here [Wiggins 2006]. However, in exploratory creativity none of the initial rules of the search space are altered, whereas in transformational creativity some are. We'll see below, for example, that the style may be varied by GAs, that is, metarules that change other rules, while remaining unchanged themselves.)

But combinational creativity is not to be sneezed at. Kurt Schwitters and Shakespeare are renowned for their H-creative collages and poetic images, which depend not on stylistic transformations but on associative processes for their origination (and their interpretation, too). Exploratory creativity, likewise, is worthy of respect—and even wonder. Indeed, the vast majority of what H-creative professional artists and scientists do involves exploratory, not transformational, creativity. Even Mozart and Crick and Watson spent most of their time exploring the spaces created by their (relatively rare) moments of transformation.

Despite what's been said above, it must also be said that there's no clear-cut distinction between exploratory and transformational creativity. That's because any rule change, however trivial, will result in structures that weren't possible before. So one must decide whether to count superficial "tweaking" as part of *exploration.* Since even the average Sunday painter may make slight changes to the style they've been taught, it's probably best to do so. And one will still have to judge, in any given case, whether the stylistic change is superficial or fundamental.

But if creativity isn't magic, it's not immediately obvious that it could be achieved or modeled by the particular types of *nonmagic* offered by AI. Nor is it immediately clear which of the three forms of human creativity would be the easiest for AI work to emulate, and which the most difficult.

Computer Combinations

That last question has a surprising answer. Contrary to what most people assume, the creativity that's most difficult for AI to model is the combinational type. Admittedly, there's no problem getting a computer to make novel combinations of familiar (already stored) concepts. That can be done until kingdom come. The problem, rather, is in getting the computer to generate and prune these combinations in such a way that most, or even many, of them are interesting—that is, valuable. What's missing, as compared with the human mind, is the rich store of world knowledge (including cultural knowledge) that's often involved.

Certainly, AI programs can make fruitful new combinations within a tightly constrained context. For instance, a program designed to solve alphabetical analogy problems, of the form *If ABC* *goes to ABD, what does MRRJJJ go to?* generates combinations that are valuable (that is, acceptable to us as answers) and sometimes surprising, and it can even judge one possible answer to be "better" than another (Hofstadter 2002). In this case, for instance, its answers included *MRRJJD, MRRDDD, MRRKKK,* and *MRRJJJJ*; and when given *XYZ* instead of *MRRJJJ,* it sometimes gave the surprisingly elegant *WYZ*. Again, the chess program Deep Blue came up with novel combinations of moves—in one case, so seemingly uncomputerlike that world champion Kasparov accused the computer scientists of cheating. And programs with chemical knowledge can come up with valuable new suggestions for molecules that might have pharmaceutical uses.

But no current AI system has access to the rich and subtly structured stock of concepts that any normal adult human being has built up over a lifetime. A few systems already have access to a significant range of concepts and factual knowledge, stored in databases such as Wordnet, Wikipedia, the CYC encyclopedia, and Google. And future programs may also have increased associative and inferential powers, based on the ontology of the semantic web. But using huge databases sensibly, and aptly, so as to match the combinations generated by linguistically—and culturally—sensitive human beings is a tall order. Not impossible in principle (after all, we don't do it by magic), but extremely difficult to achieve.

Consider, for instance, an example of H-creative combinational creativity from Shakespeare. Macbeth, sleepless because tormented by guilt, says this:

> Sleep that knits up the ravelled sleeve of care,
> The death of each day's life, sore labour's bath,
> Balm of hurt minds, great nature's second course,
> Chief nourisher in life's feast.

Describing sleep as someone knitting is certainly an unfamiliar combination of two boringly familiar ideas. And, assuming that Shakespeare didn't borrow this poetic imagery (like many of his plots) from some previous writer, such as Petrarch, it's H-creative too. But why is it apt?

Well, the knitter imagined here is not producing a new garment (the most obvious function of knitting), but mending ("knitting up") a torn sleeve. And sleep, likewise, mends the worried mind. Similarly, a bath (line 2) or balming ointment (line 3) can cure the soreness caused by one's daily work. Moreover, in Shakespeare's time, as in ours, the second course of a meal (line 3) was the most nourishing, the one best suited for replenishing the body. In short, sleep provides desperately needed relief.

It's because we know all these things that we can intuitively understand and appreciate Shakespeare's text. But a computer model needs its "intuition" to be spelled out. Think what world knowledge would be needed in the database even to recognize the aptness of these combinations, never mind originating them as Shakespeare did.

Of course, AI workers could cheat. They could build a toy system that knew *only* about worry-induced sleeplessness, knitting, sleeves, ointment, and dining customs and that—provided also with the relevant combinational processes—could decipher Shakespeare's meaning accordingly. And even that would be an achievement. But it wouldn't match the power of human minds to cope with the huge range of creative combinations that can assail us in a single day.

That power rests in the fact that our memories store (direct and indirect) associations of many different kinds, which are naturally aroused during everyday thinking. Shakespeare seemingly had access to more associations than the rest of us or to subtler criteria for judging the value of relatively indirect associations. But our ability to understand his poetry rests in this mundane fact about human memory. Moreover, this fact is biological (psychological), not cultural. The specific associations are learned within a culture, of course—as are the socially accepted styles of thought that ground exploratory and transformational creativity. But the making of associations doesn't have to be learned: it's a natural feature of associative memory. That's why combinational creativity is the easiest of the three types for human beings to achieve.

One of the best current computer models of combinational creativity is the joke-generating system JAPE (Binsted, Pain, and Ritchie 1997). One might call it a "toy" system when compared with the human mind, but it's more impressive than the imaginary system just described.

JAPE's jokes are based on combination but involve strict rules of structure too. They are punning riddles, of a type familiar to every eight year old. For example:

> What do you call a depressed train? A low-comotive.
>
> What do you call a strange market? A bizarre bazaar.
>
> What kind of murderer has fibre? A cereal killer.
>
> What's the difference between leaves and a car? One you brush and rake, the other you rush and brake.

Hilarious, these are not. But they're good enough for a Christmas cracker.

Those four riddles, along with many more, were created by JAPE. The program is provided with some relatively simple rules for composing nine different types of joke. Its joke schemas include: What kind of *x* has *y?* What kind of *x* can *y?* What do you get when you cross *x* with *y?;* and What's the difference between an *x* and a *y?*

The joke-generating rules are only "relatively" simple—and much less simple than most people would expect. That's par for the course: AI has repeatedly shown us unimagined subtleties in our psychological capacities. Think for a moment of the complexity involved in your understanding the jest (above) about the *cereal killer,* and the rather different complexities involved in getting the point of the *low-comotive* or the *bizarre bazaar.* Sounds and spellings, for instance, are crucial for all three. So making (and appreciating) these riddles requires you to have an associative memory that stores a wide range of words—not just their meanings, but also their sounds, spelling, syllabic structure, and grammatical class.

JAPE is therefore provided with a semantic network of over 30,000 units, within which new—and apt—combinations can be made by following some subset of the links provided. The network is an extended version of WordNet, a resource developed by George Miller's team at Princeton University and now exploited in many AI programs. WordNet is a lexicon whose words are linked by semantic relations such as *superordinate, subordinate, part, synonym,* and *antonym.* Dimensions coding *spelling, phonology, syntax,* and *syllable-count* had to be added to WordNet by JAPE's programmer so that the program could do its work, for JAPE uses different combinations of these five aspects of words, in distinctly structured ways, when generating each type of joke.

It wasn't enough merely to provide the five dimensions: in addition, rules had to be given to enable JAPE to locate appropriate items. That is, the rules had to define what is appropriate (valuable) for each joke schema. Clearly, an associative process that obeys such constraints is very different from merely pulling *random* combinations out of the semantic network.

The prime reason that JAPE's jokes aren't hilarious is that its associations are very limited, and also rather boring, when compared with ours. But, to avoid the superhuman human fallacy, we shouldn't forget that many human-generated jokes aren't very funny either. Its success is due to the fact that its joke templates and generative schemas are relatively simple. Many real-life jokes are much more complex. Moreover, they often depend on highly specific, and sometimes fleetingly topical, cultural knowledge—such as what the prime minister is reported to have said to the foreign secretary yesterday. In short, we're faced with the "Shakespeare's sleep" problem yet again.

Computer Exploration

Exploratory creativity, too, can be modeled by AI—provided that the rules of the relevant thinking style can be specified clearly enough to be put into a computer program. Usually, that's no easy matter. Musicologists and art historians spend lifetimes trying to identify different styles—and they aim merely for verbal description, not computer implementation. Anyone trying to model exploratory creativity requires not only advanced AI skills but also expertise in, and deep insights into, the domain concerned.

Despite the difficulties, there has been much greater success here than in modeling combinational creativity. In many exploratory models, the computer comes up with results that are comparable to those of highly competent, sometimes even superlative, human professionals.

Examples could be cited from, for instance, stereochemistry (Buchanan, Sutherland, Feigenbaum 1969), physics (Langley et al. 1987, Zytkow 1997), music (Cope 2001, 2006), architecture (Koning and Eizenberg 1981, Hersey and Freedman 1992), and visual art. In the latter category, a good example is Harold Cohen's program, AARON (Cohen 1995, 2002).

AARON's creations have not been confined to the laboratory. On the contrary, they have been exhibited at major art galleries around the world—and not just for their shock value. Under development since the late-1960s, this program has generated increasingly realistic (though not photo-realistic) line drawings, followed by colored images. The latter have included paintings, wherein the paint is applied by AARON to its own drawings, using paint brushes (or, more accurately, rounded paint pads) of half a dozen different sizes. Most recently, AARON's colored images have been subtly multicolored prints.

It's especially interesting to note Cohen's recent remark, "I am a first-class colorist. But AARON is a world-class colorist." In other words, the latest version of AARON outstrips its programmer—much as Arthur Samuel's checkers player, way back in the 1950s, learned how to beat Samuel himself (Samuel 1959).

This example shows how misleading it is to say, as people often do, "Computers can't do anything creative, because they can do only what the program tells them to do." Certainly, a computer can do only what its program *enables it* to do. But if its programmer could *explicitly tell it* what to do, there'd be no bugs—and no "world-class" color prints from AARON surpassing the handmade productions of Cohen himself.

A scientific example—or, better, a varied group of scientific examples—of exploratory creativity can be found in the work of Pat Langley and Simon's group at CMU (Langley et al. 1987, Zytkow 1997). This still-burgeoning set of programs is the BACON family, a dynasty, including close relations and more distant descendants, that has been under development since the mid-1970s (Boden 2004, 208–222). And it is this body of research on which Simon was relying when he

© Franck Boston

defended AI's status as a *science* (see previous text).

Among the early achievements of the BACON family were the "discovery" of several important laws of classical physics (Boyle's law, Ohm's law, Snell's law, Black's law, and Kepler's third law) and some basic principles of chemistry (an acid plus an alkali gives a salt; molecules are composed of identifiable elements, present in specific proportions; and the distinction between atomic and molecular weight). These generalizations were gleaned from the experimental data used by human scientists hundreds of years ago (and recorded in their notebooks); initially, the data were cleaned up for BACON.1's benefit, but they were later provided in the original noisy form. The BACON suite also reinvented the Kelvin temperature scale, by adding a constant of 273 to the Celsius value in the equation, and "discovered" the ideal gas laws ($PV / t = k$).

The words *discovery* and *discovered* need to be in scare quotes here because this was P-creativity rather than H-creativity. Although a few results were historically new (for example, a version of Black's law that is more general than the original one), most were not.

Later members of this set of programs were aimed at genuine discovery, or H-creativity. Some, for instance, suggested new experiments, intended to provide new sets of correlations, new observations, with which the program could then work when testing a theory. Others could introduce new basic units of measurement, by taking one object as the standard (human scientists often choose water). And Simon foresaw a future in which programs modeling scientific creativity could read papers in the scientific journals, so as to find extra experimental data, and hypotheses, for themselves. (To some extent, that future has already come: some bioinformatics software, such as for predicting protein structure, can improve accuracy by reading medical publications on the web. But the ideas in more discursive scientific papers are less amenable to AI.)

There's an obvious objection here, however. These programs, including the more recent ones, all assume a general theoretical framework that already exists. The physics-oriented BACON programs, for instance, were primed to look for mathematical relationships. Moreover, they were instructed to seek the simplest relationships first. Only if the system couldn't find a numerical constant or linear relationship (expressible by a straight-line graph) would it look for a ratio or a product. But one might say that the greatest creative achievement of the famous scientists modeled here, and of Galileo before them, was to see that—or even to ask whether—some observable events can be described in terms of mathematics at all, for this was the real breakthrough: not discovering *which* mathematical patterns best describe the physical world, but asking whether *any* mathematical patterns are out there to be found.

In other words, these programs were exploratory rather than transformational. They were spoonfed with the relevant questions, even though they found the answers for themselves. They have been roundly criticized as a result (Hofstadter and FARG 1995, 177–179; Collins 1989), because of the famous names (BACON and the like) used to label them. To be sure, they can explore creatively. (And, as remarked above, exploration is what human chemists and physicists do for nearly all of their time.) However, the long-dead scientists whose discoveries were being emulated here did not merely explore physics and chemistry, but transformed them.

Could a computer ever do that?

Stylistic Transformations

Many people believe that no computer could ever achieve transformational creativity. Given a style, they may admit, a computer can explore it. But if you want it to come up with a new style, don't hold your breath!

After all, they say, a computer does what its program tells it to do—and no more. The rules and instructions specified in the program determine its possible performance (including its responses to input from the outside world), and there's no going beyond them.

That thought is of course correct. But what it ignores is that the program may include rules *for changing itself*. That is, it may contain genetic algorithms, or GAs (see Boden 2006, 15.vi).

GAs can make random changes in the program's own task-oriented rules. These changes are similar to the point mutations and crossovers that underlie biological evolution. Many evolutionary programs also include a *fitness* function, which selects the best members of each new generation of task programs for use as "parents" in the next round of random rule changing. In the absence of an automated fitness function, the selection must be made by a human being.

Biological evolution is a hugely creative process, in which major transformations of bodily form have occurred. This has happened as a result of many small changes, not of sudden saltations, and few if any of those individual changes count as transformations in the sense defined above. (Even small mutations can be damaging for a living organism, and larger—transformational—ones are very likely to be lethal.) Nevertheless, over a vast period of time, the evolutionary process has delivered unimaginable changes.

It's not surprising, then, that the best prima facie examples of transformational AI involve evolutionary programming. For example, Karl Sims's (1991) graphics program produces varied images (12 at a time) from which a human being—Sims, or a visitor to his lab or exhibition space—selects one or two for breeding the next generation. (There's no automatic fitness function because Sims doesn't know what visual or aesthetic properties to favor over others.) This system often generates images that differ radically from their predecessors, with no visible family resemblance.

Sims's program can do this because its GAs allow not only small point mutations (leading to minor changes in color or form) but also mutations in which (for instance) two whole image-generating programs are concatenated, or even nested one inside the other. Since one of those previously evolved programs may itself be nested, several hierarchical levels can emerge. The result will be an image of some considerable complexity. As an analogy, consider these two trios of sentences:

(1) The cat sat on the mat; The cats sat on the mat; The dog sat on the porch, and (2) The cat sat on the mat; Aunt Flossie went into the hospital; The cat given to me by Aunt Flossie last Christmas before she went into the hospital in the neighboring town sat on the mat. Clearly, the second trio displays much greater differences than the first.

So this program undeniably delivers transformations: images that are fundamentally different from their ancestors, sometimes even from their parents. But whether it delivers transformed *styles* as well as transformed *items* is less clear, for family resemblance is the essence of style. When we speak of styles in visual art (or chemistry, or cooking), we mean a general pattern of ideas/artifacts that is sustained—indeed, explored—over time by the artist concerned, and perhaps by many other people too. But Sims's program cannot sustain a style, because some equivalent of Aunt Flossie's trip to the hospital can complicate the previous image at any time.

In brief, Sims's program is almost too transformational. This lessens the importance of the selector. Even an automatic fitness function would not prevent highly unfit examples from emerging. And when human selectors try to steer the system toward certain colors or shapes, they are rapidly disappointed: sooner rather than later, unwanted features will appear. This is frustrating for anyone seriously interested in the aesthetics of the evolving images.

That's why the sculptor William Latham, a professional artist rather than a computer scientist, uses evolutionary programming in a less radically transformational way (Todd and Latham 1992). His GAs allow only relatively minor changes to the current image-generating program, such as altering the value of a numerical parameter. Nesting and concatenation are simply not allowed. The result is a series of images that, he admits, he could not possibly have imagined for himself, but that nevertheless carry the stamp of his own artistic style. The transformations, in other words, are relatively minor and concern relatively superficial dimensions of the original style (conceptual space).

It would be possible, no doubt, for an evolutionary program to be allowed to make "Aunt Flossie" mutations only *very* rarely. In that case, there would be a greater chance of its producing transformed styles as well as transformed items. Indeed, the minor mutations might then be regarded as *exploring* the existing style, whereas the nesting/concatenating mutations might be seen as transforming it.

Whether those stylistic transformations would be valued is another matter. By definition, a creative transformation breaks some of the currently accepted rules. It may therefore be rejected out of hand—as Semmelweiss and van Gogh knew only too well. But—as neither of them lived long enough to find out—even if it is rejected, it may be revived later. In biology, nonlethal mutations lead to viable organisms, which then compete as natural selection proceeds. In human thought, social selection takes the place of natural selection. So, since being valuable is part of the very definition of creative ideas, the identification of "creativity" is not a purely scientific matter but requires socially generated judgments.

Putatively creative ideas are evaluated by means of a wide range of socially determined criteria. The criteria for scientific evalutation are relatively straightforward, and also relatively stable—even though bitter disputes about new scientific theories can arise. Those for fashion and art are not. So if structural transformation is necessary for a novel idea to be hailed as a triumph of style-changing creativity, it certainly isn't sufficient.

Is Transformational AI Actually Possible?

I said, above, that the best prima facie examples of transformational AI involve evolutionary programming. Why that cautious "prima facie"?

Sims's program, after all, does generate radically transformed images. And Latham's program generates new visual styles, even if the family resemblances to the ancestor styles are relatively obvious. Moreover, we don't need to focus only on the contentious area of art, nor only on cases where a human selector decides on "fitness." Even a very early GA program was able to evolve a sorting algorithm that could put a random set of numbers into an increasing series, or order words alphabetically (Hillis 1992). Since then, many other highly efficient algorithms have been automatically evolved from inferior, even random, beginnings. If that's not transformation, what is?

Well, the objection here comes from people who take the biological inspiration for evolutionary programming seriously (Pattee [1985]; Cariani [1992]; see also Boden [2006, 15.vi.c]). They assume that AI is either pure simulation or abstract programming that defines what sort of interactions can happen between program and world (as in computer vision, for example). And they offer a version of the familiar argument that *A computer can do only what its program tells it to do*. Specifically, they argue that genuine transformations can arise in a system only if that system interacts *purely physically* with actual processes in the outside world, as biological organisms do.

Their favorite example concerns the origin of new organs of perception. They allow that once a light sensor has arisen in biology, it can evolve into better and better sensors as a result of genetic mutations that can be approximated in AI programs. So an inefficient computer-vision system might, thanks to GAs, evolve into a better one. But the *first* light sensor, they insist, can arise only if some mutation occurs that causes a bodily change that happens to make the organism sensitive to light for the very first time. The light—considered as a physical process—was always out there in the world, of course. But only now is it "present" for the organism. One might say that only now has it passed from the world into the environment.

That acceptance of light as part of the organism's environment depends crucially on physical processes—both in the world and in the living body. And these processes, they say, have no place in AI.

They grant that the generative potential of a computer program is often unpredictable and may even be indefinitely variable, as it is for most evolutionary programs. But still, it is constrained by the rules (including the GAs) in the program. And if it interacts with events in the outside world, as a computer-vision system or a process monitor does, the types of data to which it is sensitive are preordained. Certainly, they say, improved sensory artifacts can result from evolutionary computing. And those improvements may be so surprising that we naturally classify them as "transformations." But (so the argument goes) no fundamentally new capacities can possibly arise.

For instance, if the physical parameters foreseen by the programmer as potentially relevant don't happen to include light, then no artificial eye can ever emerge. In general, then, there can be no *real* transformations in AI.

That may be true of AI systems that are pure simulations. But it's demonstrably not true of all AI systems—in particular, of some work in so-called embodied AI, for recent research in this area has resulted in the evolution of *a novel sensor:* the very thing that these critics claim can happen only in biology.

In brief, a team at the University of Sussex were using a GA to evolve oscillator circuits—in hardware, not in simulation (Bird and Layzell 2002). To their amazement, they ended up with a primitive radio receiver. That is, the final (GA-selected) circuit acted as a primitive radio antenna (a "radio wave sensor"), which picked up and modified the background signal emanating from a nearby PC monitor.

On investigation post hoc, it turned out that the evolution of the radio-wave sensor had been driven by unforeseen physical parameters. One of these was the aerial-like properties of all printed circuit boards, which the team hadn't previously considered. But other key parameters were not merely unforeseen but unforeseeable, for the oscillatory behavior of the evolved circuit depended largely on accidental—and seemingly irrelevant—factors. These included spatial proximity to a PC monitor; the order in which the analog switches had been set; and the fact that the soldering iron left on a nearby workbench happened to be plugged in at the mains.

If the researchers had been aiming to evolve a radio receiver, they would never have considered switch order or soldering irons. Nor would either of these matters necessarily be relevant outside the specific (physical) situation in which this research was done. On another occasion, perhaps, arcane physical properties of the paint on the surrounding wallpaper might play a role. So we can't be sure that even research in *embodied* AI could confidently *aim* to evolve a new sensor. The contingencies involved may be too great, and too various. If so, doubt about (nonaccidental) genuine transformations in AI still stands. But that they can sometimes happen unexpectedly is clear.

Computer Models and Computer Art

All computer models of creativity are aimed at the production of P-creative ideas, and a few at H-creativity too. And many are intended also to throw some light on creativity in human minds. Some, however, function in ways that have no close relation to how the the mind works: it's enough that they generate creative outcomes.

Examples of the latter type include most of the AI programs employed in the various forms of computer art. (The different types are distinguished, and their varying implications for "creativity" outlined, in Boden and Edmonds [2009].) One might say that these aren't really computer "models" at all, but rather computer programs—ones that may sometimes seem to work in creative ways. (AARON was unusual in that Cohen—already a highly successful abstract painter—first turned to AI techniques in the hope of understanding his own creativity better.) Most computer artists are interested not in human psychology but in the aesthetic value of their program's performance.

That performance may be a stand-alone matter, wherein the computer generates the result all by itself. Having written the program, the human artist then stands back, hands off, to let it run. These are cases of *generative art,* or G-art (Boden and Edmonds 2009).

Where G-art is involved, it's especially likely that the AI system itself—not just its human originator—will be credited with creativity. In *evolutionary art* too (see the following text), much of the creativity may be credited to the program, for here, the computer produces novel results—images or melodies, for instance—that the human artist couldn't predict, or even imagine. In yet other cases of computer art, such as the *interactive* art described below, some or all of the creativity is attributed to the programmer or the human participants. The interactive program isn't designed to be (or even to appear to be) creative in its own right, but rather to produce aesthetically attractive/interesting results in noncreative ways.

The preeminent case of G-art in the visual arts is AARON, whose programmer tweaks no knobs while it is running. In music, perhaps the best-known example is the work of the composer David Cope (2001, 2006).

Cope's program Emmy (from EMI: Experiments in Musical Intelligence) has composed music in the style of composers such as Bach, Beethoven, Chopin, Mahler … and Scott Joplin, too. Some are pieces for solo instrument, such as a keyboard fugue or sonata, while others are orchestral symphonies. They are remarkably compelling, striking many musically literate listeners—though admittedly not all—as far superior to mere pastiche. That's sometimes so even when the listener approached Emmy's scores in a highly sceptical spirit. For instance, the cognitive scientist Douglas Hofstadter, a leading figure in the computer modeling of creativity (Hofstadter and FARG 1995, Rehling 2002), believed it to be impossible that traditional AI techniques could compose music of human quality. But on playing through some Emmy scores for new "Chopin mazurkas," a genre with which Hofstadter, a fine amateur musician, was already very familiar, he was forced to change his mind (Hofstadter 2001, 38*f*.).

Other examples of computer art are not standalone, but *interactive* (Boden and Edmonds 2009; Boden in press); that is, the computer's performance is continually affected by outside events while the program is running.

Those "outside events" may be impersonal matters such as wave movements or weather conditions, but usually they are the movements/actions of human beings. Given that what the system does on any occasion depends in part on the actions of the human audience, the causal relation may be obvious enough for the person to choose what effect to have on the program's performance. Sometimes, however, such predictions are impossible: the audience affects what happens but, perhaps because of the complexity of the causality involved, they don't know how. They may not even realize that this is happening at all—for instance, because of a built-in delay between (human-generated) cause and (computer-generated) effect.

One interactive program, written by Ernest Edmonds, was chosen by the curators of a Washington, D.C., art gallery to be run alongside the works of Mark Rothko, Clyfford Still, and Kenneth Noland, in a 2007 exhibition celebrating the 50th anniversary of the "ColorField" painters. (So much for the view that computer art can't really be art—see below.)

An interactive artwork commissioned for the huge millennial exhibition filling London's newbuilt Millennium Dome was described by a *Times* journalist as "the best bit of the entire dome." This was Richard Brown's captivating *Mimetic Starfish*. The starfish is a purely virtual creature: a visual image generated by a self-equilibrating neural network that's attached to sensors in the vicinity. The image is projected from the ceiling down onto a marble table, and appears to onlookers to be a large multicolored starfish trapped inside it. But despite being "trapped" inside the table, it moves. Moreover, it moves in extraordinarily lifelike ways, in response to a variety of human movements and sounds. If someone shouts, for instance, or suddenly pounds the table, the starfish instantly "freezes" as a frightened animal might do.

Interactive art isn't wholly new: Mozart's dice music is one ancient example. (Someone would throw a die to decide the order in which to play Mozart's precomposed musical snippets, and the result would always be a coherent piece.) But because of the general-purpose nature of computing, a very wide range of types of interaction can be accommodated, many of which were previously unimaginable.

In computer-based interactive art, the aesthetic interest is not only, or not even primarily, in the intrinsic quality of the results (images and sounds). Rather, it is in the nature of the interaction between computer and human beings (Boden in press). The "audience" is seen as a *participant* in the creation of the artwork—especially if the causal relations between human activity and computer performance are direct and intelligible. In the latter case, one can voluntarily shape the computer's performance so as to fit one's own preferences. But whether the the relatively direct cases are more artisticially interesting than the indirect ones is disputed: there's no consensus on just what type of interactions are best from an aesthetic point of view.

As for evolutionary art, two pioneering examples (Sims and Latham) have been mentioned already. Today, young computer artists are increasingly using evolutionary techniques in their work. One main reason is the potential for surprise that this (randomness-based) approach provides. Another is its connection with A-Life, and with life itself. Some evolutionary artists even claim that their work, or something like it, may one day generate "real" life in computers (Whitelaw 2004). (They are mistaken, because computers—although they use energy, and can even budget it—don't metabolize [Boden 1999].)

Additional types of computer art exist, which can't be discussed here. But there is a debilitating occupational hazard that faces all who work in this area, whichever subfield they focus on. Namely, many members of the general public simply refuse point-blank to take their work seriously.

Consider Emmy, for instance. I said, above, that Emmy *composes* music in the style of Bach and other composers. I should rather have said that it *composed* such music, for in 2005, Cope destroyed the musical database that had taken him 25 years to build and that stored musical features characteristic of the composers concerned (Cope 2006, 364). His reason was twofold. First, he had found over the years that many people dismissed Emmy's compositions (sometimes even refusing to hear them at all), failing to take them seriously because of their nonhuman origin. Second, even those who did appreciate Emmy's scores tended to regard them not as *music* but as *computer output.* As such, they were seen as infinitely reproducible—and devalued accordingly (Boden 2007). Now, however, Emmy has a finite oeuvre—as all human composers, beset by mortality, do.

Some of Emmy's detractors would be equally adamant in dismissing every other example of computer art. They might admit that the *Mimetic Starfish*, for example, is both beautiful and—for a while—intriguing. But they would regard it as a decorative gimmick, not as art. For them, *there can be no such thing as computer art*. Despite the fact that there is always a human artist somewhere in the background, the mediation of the computer in generating what's actually seen or heard undermines its status as *art*.

This isn't the place to attempt a definition of the notoriously slippery concept of art. (But remember that a computer artwork was chosen by traditional gallery curators for inclusion in Washington's "ColorField" exhibition; see above.) In other words, it's not the place for discussing whether computer art is "really" art.

But we can't wholly ignore a closely related question, which is often in people's minds when they deny the possibility of computer art. Namely, can a computer "really" be creative?

But Are Computers Creative, Really?

Whether a computer could ever be "really" creative is not a scientific question but a philosophical one. And it's currently unanswerable, because it involves several highly contentious—and highly unclear—philosophical questions.

These include the nature of meaning, or intentionality; whether a scientific theory of psychology, or consciousness, is in principle possible; and whether a computer could ever be accepted as part of the human moral community. Indeed, you can ignore *creativity* here, for many philosophers argue that no naturalistic explanation of *any* of our psychological capacities is possible, not even an explanation based in neuroscience. In short, the philosophical respectability of "strong" AI, and of cognitive science in general, is hotly disputed.

These are among the very deepest questions in philosophy. I've discussed them elsewhere (Boden 2004, chapter 11; Boden 2006, chapter 16). I've also argued that the ultrasceptical, postmodernist view is irrational and fundamentally self-defeating (Boden 2006, 1.iii.b, 16.vi–viii). But there's no knock-down refutation on either side. That being so, even the youngest readers of *AI Magazine* shouldn't expect to see these questions to be definitively answered in their lifetimes.

The scientific questions offer more hope. Enough advance has already been made in computational psychology and computer modeling to make it reasonable to expect a scientific explanation of creativity. Optimists might even say that it's already on the horizon. This doesn't mean that we'll ever be able to predict specific creative ideas, any more than physicists can predict the movements of a single grain of sand on a windswept beach. Because of the idiosyncracy and (largely hidden) richness of individual human minds, we can't even hope to explain all creative ideas post hoc. But, thanks in part to AI, we have already begun to understand *what sort of phenomenon* creativity is.

Still something of a mystery, perhaps. And certainly a marvel. But not—repeat, not—a miracle.

References

Binsted, K.; Pain, H.; and Ritchie, G. D. 1997. Children's Evaluation of Computer-Generated Punning Riddles. *Pragmatics and Cognition* 5(2): 305–354.

Bird, J., and Layzell, P. 2002. The Evolved Radio and Its Implications for Modelling the Evolution of Novel Sensors. In *Proceedings of Congress on Evolutionary Computation,* CEC-2002, 1836–1841. Pisacataway, NJ: Institute of Electrical and Electronics Engineers.

Boden, M. A. In press. Aesthetics and Interactive Art. In *Art, Body, Embodiment,* ed. R. L. Chrisley, C. Makris, R. W. Clowes, and M. A. Boden. Cambridge: Cambridge Scholars Press.

Boden, M. A. 2007. Authenticity and Computer Art. Special Issue on Creativity, Computation, and Cognition, ed. P. Brown. *Digital Creativity.* 18(1): 3–10.

Boden, M. A. 2006. *Mind as Machine: A History of Cognitive Science*. Oxford: Oxford University Press.

Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms,* 2nd ed. London: Routledge.

Boden, M. A. 1999, Is Metabolism Necessary? *British Journal for the Philosophy of Science* 50(2): 231–248.

Boden, M. A., and Edmonds, E. A. 2009. What Is Generative Art? *Digital Creativity* 20(1–2): 21–46.

Buchanan, B. G.; Sutherland, G. L.; and Feigenbaum, E. A. 1969, Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry. In *Machine Intelligence 4,* ed. B. Meltzer and D. M. Michie, 209–254. Edinburgh: Edinburgh University Press.

Cariani, P. 1992. Emergence and Artificial Life. In *Artificial Life II,* ed. C. J. Langton, C. G. Taylor, J. D. Farmer, and S. Rasmussen, 775–797. Reading, MA: Addison-Wesley.

Cohen, H. 1995. The Further Exploits of AARON Painter. In Constructions of the Mind: Artificial Intelligence and the Humanities, ed. S. Franchi and G. Guzeldere. Special edition of *Stanford Humanities Review,* 4(2): 141–160.

Cohen, H. 2002. A Million Millennial Medicis. In *Explorations in Art and Technology,* ed. L. Candy and E. A. Edmonds, 91–104. London: Springer.

Collins, H. M. 1989. Computers and the Sociology of Scientific Knowledge, Social Studies of Science, 19: 613–624.

Cope, D. 2006. *Computer Models of Musical Creativity.* Cambridge, Mass.: The MIT Press.

Cope, D. 2001. *Virtual Music: Computer Synthesis of Musical Style*. Cambridge, Mass.: The MIT Press.

Hersey, G., and Freedman, R. 1992. *Possible Palladian Villas Plus a Few Instructively Impossible Ones*. Cambridge, MA: The MIT Press.

Hillis, W. D. 1992. Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure. In *Artificial Life II,* ed. C. J. Langton, C. G. Taylor, J. D. Farmer, and S. Rasmussen, 313–324. Reading, MA: Addison-Wesley.

Hofstadter, D. R. 2002. How Could a COPYCAT Ever be Creative? In *Creativity, Cognition, and Knowledge: An Interaction,* ed. T. Dartnall, 405–424. London: Praeger.

Hofstadter, D. R., and FARG, The Fluid Analogies Research Group. 1995. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought.* New York: Basic Books.

Hofstadter, D. R. 2001. Staring Emmy Straight in the Eye—And Doing My Best Not to Flinch. In *Virtual Music: Computer Synthesis of Musical Style*, ed. D. Cope, 33–82. Cambridge, Mass.: The MIT Press.

Koning, H., and Eizenberg, J. 1981. The Language of the Prairie: Frank Lloyd Wright's Prairie Houses. *Environment and Planning, B,* 8(4): 295–323.

Langley, P. W.; Simon, H. A.; Bradshaw, G. L.; and Zytkow, J. M. 1987. *Scientific Discovery: Computational Explorations of the Creative Process*. Cambridge, MA: The MIT Press.

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

Lenat, D. B. 1983. The Role of Heuristics in Learning by Discovery: Three Case Studies. In *Machine Learning: An Artificial Intelligence Approach,* ed. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, 243–306. Palo Alto, CA: Tioga.

McCarthy, J.; Minsky, M. L.; Rochester, N.; and Shannon, C. E. 1955. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. *AI Magazine* 27(4): 12–14.

Pattee, H. H. 1985. Universal Principles of Measurement and Language Functions in Evolving Systems. In *Complexity, Language, and Life: Mathematical Approaches,* ed. J. Casti and A. Karlqvist, 168–281. Berlin: Springer-Verlag.

Rehling, J. A. 2002. Results in the Letter Spirit Project. In *Creativity, Cognition, and Knowledge: An Interaction,* ed. T. Dartnall. London: Praeger, 273–282.

Samuel, A. L. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3: 211–229.

Simon, H. A. 1995. Explaining the Ineffable: AI on the Topics of Intuition, Insight, and Inspiration. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* vol. 1, 939–948. San Francisco: Morgan Kaufmann Publishers.

Sims, K. 1991. Artificial Evolution for Computer Graphics. *Computer Graphics*. 25(4): 319–28.

Whitelaw, M. 2004. *Metacreation: Art and Artificial Life*. London: The MIT Press.

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

Zytkow, J. ed. 1997. *Machine Discovery.* London: Kluwer Academic.

Margaret A. Boden (research professor of cognitive science at the University of Sussex) was the founding dean of Sussex University's School of Cognitive and Computing Sciences. She is a Fellow (and former vice-president) of the British Academy, a member of the Academia Europaea, a Fellow of the Association for the Advancement of Artificial Intelligence, a Fellow of the European Coordinating Committee for Artificial Intelligence, a Life Fellow of the UK's Society for Artificial Intelligence and the Simulation of Behaviour, a member of Council of the Royal Institute of Philosophy, and the former chairman of Council of the Royal Institution of Great Britain. Boden has lectured widely, to both specialist and general audiences, in North and South America, Europe, India, the former USSR, and the Pacific. She has also appeared on many radio and TV programs in the UK and elsewhere. Her work has been translated into 20 foreign languages. She was awarded an OBE in 2001 (for "services to cognitive science"), and beside her Cambridge ScD, she has three honorary doctorates (from Bristol, Sussex, and the Open Universities).