

Creativity at the Metalevel

AAAI-2000 Presidential Address

Bruce G. Buchanan

■ Creativity is sometimes taken to be an inexplicable aspect of human activity. By summarizing a considerable body of literature on creativity, I hope to show how to turn some of the best ideas about creativity into programs that are demonstrably more creative than any we have seen to date. I believe the key to building more creative programs is to give them the ability to reflect on and modify their own frameworks and criteria. That is, I believe that the key to creativity is at the metalevel.

It is a widely held view that creative thought is a necessary part of the complex of behaviors we use to define intelligence. In the AAAI-1998 Presidential Address, for example, David Waltz said that creativity is a key topic for AI research because it is an essential element of human intelligence (Waltz 1999). If it truly is necessary, then it must be addressed for a resolution to the philosophical questions about whether machines can think. As Dartnall puts it in the *Encyclopedia of the Philosophy of Mind*,¹ if creativity is a human process that cannot be described mechanistically, then human minds cannot be symbol-manipulation machines.

From the point of view of an experimentalist, AI is the perfect medium for understanding creativity because implementing ideas in computer programs gives us the means to test these ideas. Therefore, I discuss some of the empirical work in psychology and experimental work in AI and what it seems to tell us about methods for building machines capable of creative activity.

Common Usage

I first note the obvious: There is no consensus, just considerable ambiguity, about what we call creative behavior or what is involved in this behavior.² In everyday speech, gifted people who create new ideas, new works of art, new music, and so on, are said to “think outside the box,” “break the rules,” “revolutionize the field,” “think intuitively,” “think different,” and “change the way we think.”

Psychologists use much the same kind of language but with longer words. They say that creative people “think in uncharted waters” (Lee [1957] quoted in Taylor [1988], p. 118), “go off in different directions” (Guildford [1950] quoted in Taylor [1988], p. 119), “destroy one Gestalt in favor of a better one” (Wertheimer [1945], quoted in Taylor [1988], p. 118), suppress conscious thought to let pre-conscious material emerge (Bellak 1958), arrange old elements into a new form (Koestler 1964; Harmon 1956), and add to the existing stored knowledge of mankind (Newell, Shaw, and Simon 1958).

These six samples of early psychological studies of creativity represent different aspects of creative behavior that have subsequently been studied and elaborated on (see, for example, Taylor [1988]). They also represent different aspects that made their way into contemporary models of creativity in computer programs. I have deliberately highlighted older work to underscore the fact that psychologists have been interested in understanding creative work at least since the 1950s; a recent overview with many personality traits associated with

creative people (and an extensive bibliography) is in Dacey and Lennon (1998). Sternberg (1988a) surveyed people to see what kinds of characteristics they (we) actually associate with creative behavior. Six major elements turned up: (1) lack of conventionality; (2) the recognition of similarities and differences and the making of connections; (3) appreciation for and ability to write or draw or compose music; (4) flexibility to change directions; (5) willingness to question norms and assumptions; and (6) motivation and energy.

Editors of a recent volume entitled *Creativity and Madness* (Panter et al. 1995, p. xiii) encapsulate much of the earlier writing by psychologists:

Creativity is the ability to bring something new into existence, by seeing things in a new way. Those who have this in the greatest degree are considered geniuses, and greatly honored and rewarded, but frequently considered strange, disturbing, and even mad.... Creativity is a constructive outlet for painful feelings and confused states of being.

In addition to melancholy and madness, some of the psychology literature also contains an undercurrent of mystery surrounding the creative process, for example:

The more one studies the subject of creativity, the more complex and bewildering it seems, and the closer one comes to accepting Freud's conclusion that it simply cannot be understood (Berman 1995, p. 59).

AI scientists, however, have always been more optimistic about understanding all aspects of human thinking (Simon 1979, 1967). Polya's (1945) pioneering work on understanding the process of problem solving helped remove the mystery. In one of the first descriptions of creativity in AI programs, Newell, Shaw, and Simon (1958) focused on characteristics of the product, the process, the person, and the problem that come together when we call someone's thinking about a problem to be creative:

Product novelty: The product of the thinking has novelty and value.

Process unconventionality: Thinking is unconventional; that is, it requires modification or rejection of previously held ideas.

Person's persistence: Thinking requires high motivation and persistence.³

Problem difficulty: The initial problem was vague and ill defined.

Margaret Boden (1994) has brought together many of these elements into an elegant working definition of creativity. She wrote that in art, music, design, problem solving, science, and so on, creativity involves generating ideas that are both novel and valuable. At the risk of oversimplifying, by *novelty*, she means that the idea is either new to the person or new in the history of the field; by *valuable*, she means that the idea meets the approval or satisfies the values of the social group.

Boden points out that *creativity* is a judgmental term; there is a value judgment associated with calling a person creative. People make judgments differently and can apply the same adjective differently to different people or use different adjectives to describe more or less the same kind of behavior in different people. In children and adults, for example, we might talk about their problem solving or their art making as "pedestrian" or "routine" at the lower levels and "clever," "imaginative," or "revolutionary" at the higher levels, with many things in between. Calling someone "creative" implies that they are more imaginative than is implied by calling them "merely clever," and their contributions are more important than those of routine, journey person thinkers or artists. Boden also emphasizes the importance of a generative mechanism.

Our own definitions of creativity also partly depend on how rare we think the phenomenon is. If we want to restrict the class to just a few people a century, or maybe a few in any discipline in a century, then we apply much stricter criteria—usually insisting that the person's ideas have caused the rest of society to change the way they think about things. Beethoven, for example, introduced human voices into symphonic music in ways that had never been done before and, thus, changed forever the nature of music. However, if we believe that every advertising agency and startup company has as many creative people as they claim, we must be willing to use criteria that many people satisfy in ordinary life, not just the singular geniuses. In part, this decision might have more to do with how much society values the product than with the process itself (Gedo 1990).

The process of creative thought, however, might not differ between the everyday thought of every individual and the rare thoughts that earn a place in history. Schank and Cleary (1995, p. 229), for example, write:

These small acts of creativity, though they differ in scope, are not different in kind from the brilliant leaps of an Einstein. Creativity is commonplace in cognition, not an esoteric gift bequeathed only to a few.

The question of whether failures can be creative also forces us to think about how we want to define the class of creative behaviors and creative people. Must an engineering design actually work before we call its designer creative? If none of Edison's inventions had been successful, would we refer to him as a creative genius or merely a crackpot?

It is often said that creative people "break the rules," but is rule breaking either necessary or sufficient?

As Johnson-Laird (1999) points out, it is certainly not sufficient:

We can be certain that high creativity is not just a matter of "breaking the rules." ... There are many ways to break the rules of any genre: almost all of them are uninteresting and aesthetically unappealing."

Similarly, background knowledge seems to be an essential element to distinguish deliberate acts of creation from what Weisberg (1993) calls "accidental" creativity. In science and other problem-solving activities, a novice with no technical expertise is less likely to be called creative for making pronouncements that break the rules than is an experienced person. A child might have written $e = mc^2$, for example. Without the background knowledge that puts the formula into context and suggests why it is important, it is more an exercise in penmanship than in physics (Kuhn 1970). Also, it is less likely we will call artists creative if they have not developed a set of skills and knowledge that lets us see their work as a departure from a solid base. Picasso's technical drawing skills, for example, gave him a base for departure from representational art. Chase and Simon (1973) suggested that years of preparation are essential for developing skills and knowledge that allow a person to perform at grand-master levels in any discipline.

Johnson-Laird (1988) sums this up in a sentence:

Geniuses need to know more, and to have this knowledge in a form that can control the generation of new ideas.

Sternberg (1988b, p. 137), too, puts it succinctly:

It is impossible to have novel ideas about something if one knows nothing about it.

However, many writers have pointed out the inhibiting effect of having too much knowledge or at least of believing too strongly in the framework that one already has. In describing Alexander Graham Bell's work, for example, his assistant, Watson (1913, p. 21), quotes his own mentor in electricity, Moses Farmer:

If Bell had known anything about electric-

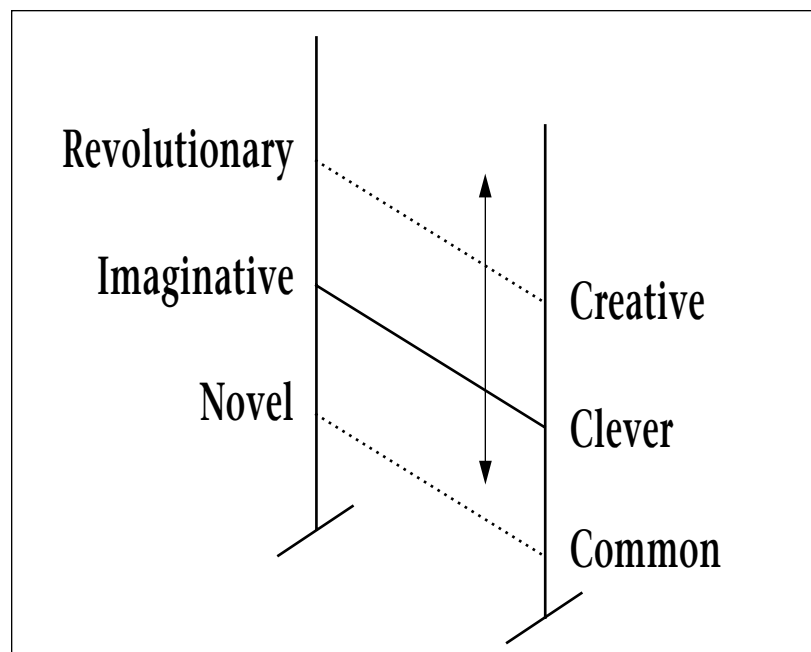


Figure 1. Raising the Bar on What We Call Creative Thought.

ity he would never have invented the telephone.

Although there is no consensus on exactly what we mean when we call a person *creative*,⁴ we do have a sense of when most of us would say that one person is more creative than another. We can use this sense as a guide to developing more creative programs and, thus, raise the bar on what we have demonstrated AI programs can do (figure 1).

Marvin Minsky links the criteria defining creative behavior with the possibility that machines can be creative. Once we can say what behavior we are looking for, he says, then there is no doubt that machines are capable of it. On what he calls the puzzle principle, he writes (Minsky 1986b):

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.

That is, defining the criteria for acting creatively assures us that a computer can be creative.

There are many other people in AI and psychology whose writing on creativity I am drawing on, and I would recommend to you. The literature in psychology is voluminous, and it is not entirely clear how we can get our arms around it, but the titles listed in the bibliography, starting with Sternberg (1999) and Boden (1994), might be good places to start.

Poetry	M. Masterman	Program for Haiku
Music	D. Cope	EXPERIMENTS IN MUSIC
Art	H. Cohen	AARON
Historical science	H. Simon et al.	BACON, GLAUBER, STAHL
Chemistry	E. Feigenbaum et al.	DENDRAL and METADENDRAL
Mathematics	D. Lenat	AM and EURISKO

Table 1. Examples of Programs That Have Been Called Creative.

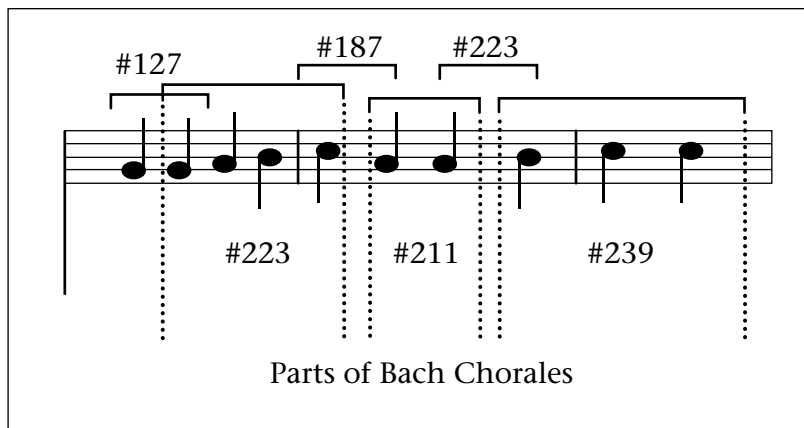


Figure 2. Composition by Substitution in D. Cope's Program.

In this example, the program has started with an initial phrase from Bach Chorale #127, added a phrase from #223 that starts on the ending note of #127, and so on.

Creative AI Programs

In the literature in and about AI, several well-known programs have been called creative (for example, Boden [1999]). In one of the papers on the logic theorist program (LT), Newell, Shaw and Simon (1958) wrote explicitly about creative behavior and the possibility that LT's discoveries of new proofs in logic were creative. Even in the mid-1950s, they were finding examples of several working programs they were willing to call creative. These included, besides LT, programs that were composing music, playing chess, and designing electric motors. What many of these programs have in common is the ability to put together known elements in new ways.

Examples of some other programs that have been called creative include those listed in table 1. Three are from literature and the arts, three from science and mathematics. If you keep in mind the combinatorial aspects of these programs, you will understand why they are often used as illustrations.

In haiku poetry, perhaps the most creative person was the person a few centuries ago who changed an art form called hokku into the art form called haiku by adding one more constraint on the semantics:

Hokku Syntax:

1. 3 lines
2. Lines one and three = 5 syllables
3. Line two = 7 syllables

Hokku Semantics:

4. Poem suggestive or epigrammatic

Haiku Constraint:

5. The need to refer in some way to one of the seasons, often very obliquely.

Margaret Masterman (1971, p. 180) nearly 30 years ago explored natural language generation in the context of a haiku writing program. One example from this program is

All black in the mist,
I trace thin birds in the dawn.
Whirr! The crane has passed.

She describes how this haiku satisfies the constraints, how it could be modified, and what it means, and she explicitly talks about the oblique reference to the seasons in this haiku (birds migrating). However, is the art form so constrained that it leaves no room for creativity at all, or are we willing to call a haiku master creative? It is a question that we all have to answer for ourselves because the notion of structure is terrifically important in every activity. In particular, is it possible for creative problem solving to be carried on without any structure at all, and what would it mean?

In an invited talk at AAAI-1998, the composer David Cope (1998) described his program EXPERIMENTS IN MUSIC, which is distinctly combinatorial in nature. By the account of most everyone who listened to examples from the program, the program is creative indeed. The program considers specific elements of Bach chorales, Rachmaninoff concertos, or Cope's own compositions and links small pieces together, adding one at a time, with the semantic rules that preserve phraseology and an overall sense of musical flow. In one case, which I have oversimplified in figure 2, Cope's program has taken the first two notes from Chorale 127 and then, using some linking rules, the next four notes from Chorale 223, and so on. The program continues combining overlapping fragments, guided by its sense of overall continuity and phrase, to create a pleasing chorale that Bach never wrote but might have.

Even though we can explain quite mechanically how the piece is assembled, the result appears to be creative.

Harold Cohen's art program, named AARON (McCorduck 1991), is a much more talented and creative artist than most of us would claim to be. AARON is an especially interesting example because Cohen himself does not think that it is creative for reasons that I try to explain.

Figure 3 shows an example of one of AARON's more recent paintings—and yes, AARON designed and executed everything in this drawing, including the choice of colors and the coloring itself.⁵

This pseudoportrait, like every painting produced by AARON, is unique, although it is executed within a framework that Cohen has specified in the code that avoids, for example, coloring faces green or purple. Therefore, AARON will never make a choice to break the rules, nor will it reflect on those constraints as something that it might want to change. There certainly is a framework, which might even be called a rigid framework, that includes a fixed palette of 800 colors to work with, but it is clearly less rigid than the haiku structure. Within this framework, AARON is moving through an immense combinatorial space where each thing that it does is predicated on what it has just done.

Now, one of the limiting factors in both Cohen's use of the term *creativity*, and his need to avoid it for this program, is that AARON has no sense of continuity or sense of experience from one drawing to the next. Even though each one is unique, not one of them is based on any prior experience from previous drawings or a previous history in a way that Cohen believes creative visual artists need to be. Part of his reluctance to say that AARON exhibits creativity is philosophical:

“Creative” is a word I do my very best never to use if it can be avoided (Cohen 1999a).

However, when pressed to talk about this kind of behavior, which he calls *behavior X*, Cohen says it is “the ability of the individual ... to move forward, to develop, to introduce new material...” (Cohen 1999b). Some of the components of behavior *X*, according to Cohen, are the abilities to construct new territory, reflect on the predicates and the criteria, modify his/her own—and the accepted—criteria of the field, and learn from past actions.

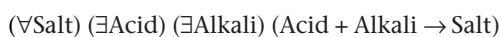
AARON does none of these. To someone who is a creative artist, AARON's work might seem “merely clever”; to me, it seems imaginative enough within its large, albeit fixed, framework to be called creative.

In the realm of science, programs have demonstrated unusual abilities to produce novel solutions to problems. Programs in the family of BACON, GLAUBER, STAHL, and DALTON by Simon, Langley, Zytkow, and others, create explanations of historical facts in science (Langley et al. 1987). By most every account, they are exercising creativity in proposing hypotheses that scientists have become famous for.



Figure 3. A Painting Conceived and Executed by H. Cohen's Program AARON (Reproduced with permission).

The task of each of these programs was to postulate some laws of science that explain the facts as known at the time, without the benefit of modern science. BACON was working with quantitative laws, and GLAUBER, among others, was working on formulating qualitative laws. For example, given a set of facts, such as the fact that hydrochloric acid (HCl) and sodium hydroxide (NaOH) react to form sodium chloride (NaCl), GLAUBER formulates the general rule that every salt is the reaction product of an acid and an alkali:



As an example of combinatorial behavior that is considered creative, Boden cites DENDRAL's problem solving in chemistry (Lindsay et al. 1980). DENDRAL produced plausible solutions to new problems through what we have come to call the *exclusion paradigm* (Brinkley et al. 1988): With a complete definition of the search space, generate all plausible combinations of the elementary units (that is, those combinations that fit explicit constraints), and

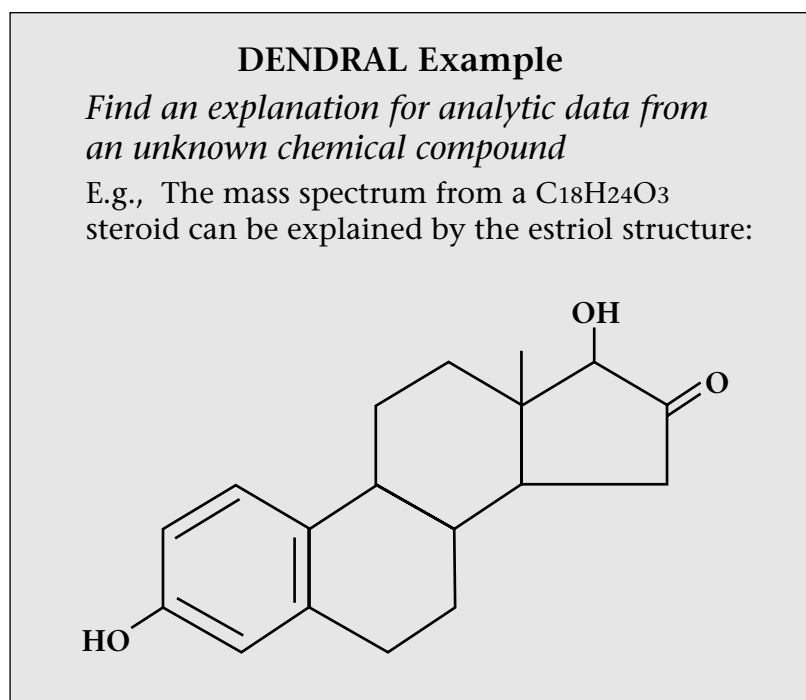


Figure 4. From the Mass Spectrum of Some Compound with 18 Carbon and 3 Oxygen Atoms.

DENDRAL finds the structure that explains the empirical data. The compound estriol is postulated as such a structure that will explain the data.

prune items that are not plausible—then all items not excluded remain plausible. Figure 4 shows an example.

Although the problems that DENDRAL solves are difficult, it is easy to see how the distinctly combinatorial method can put chemical atoms and groups of atoms into chemical graphs to find solutions to new problems. (See Buchanan [1985] for other early examples.)

With the METADENDRAL program, the task was to formulate the rules that allowed the DENDRAL program to determine which explanations were plausible. From a set of data linking known chemical structures and known data, METADENDRAL finds the general rules that explain the whole collection. For the class of steroid structures, of which estriol is a particular example, figure 5 shows some of the kinds of rule that METADENDRAL found, where each arrow corresponds to a rule indicating which chemical bonds would be broken in the instrument and where the electronic charge would be located. (There are a number of primitives available besides breaking and charging; for example, neutral molecules would be lost, and various numbers of ions could be transferred from one half to the other.) All the primitives are combined in understandable ways in a space that is immense but produces a result

that looks plausible and could sometimes be called creative.

Some of METADENDRAL's results became the first examples of a scientific result discovered by a program that were new and useful enough to be published in the refereed scientific literature (Buchanan et al. 1976).

Doug Lenat's program classics, AM and EURISKO, are two of the most creative programs in the AI literature (Davis and Lenat 1982; Lenat 1983). Both programs are more than "merely clever." From a number of primitive axioms of set theory, AM was asked to find interesting conjectures and found many important concepts and conjectures in number theory. Among other things, it rediscovered the concept of numbers with exactly two divisors, that is, prime numbers, as being an interesting set of numbers, quite correctly so. (For another example in mathematics, see Colton, Bundy, and Walsh [2000]).

Lenat's EURISKO program moved a level up to explore a space of heuristics so that a program such as AM would have the ability to move through its own well-defined space and make better conjectures than it was able to before. EURISKO could, as most of you know, learn heuristics for playing games. Looking at Extrema, for example, it was able to find loopholes in the rules of a tournament game that allowed a game-playing program to beat all human opponents. Although the framework for the problem-solving (game playing) program is fixed, another program is able to change it to "break the rules."

Models for Creative Programs

Much of the work on creativity in psychology and AI has been analyzed and distilled into various models that can be used either to recognize creative work or produce it. Of course, a starting point for developing AI models is believing that creativity is another facet of cognitive activity that can be explained, that is, it is not inexplicable. Early in the history of AI, Minsky (1963, p. 447) made this point that there might be nothing mysterious about explaining intelligent behavior in the first place:

It may be ... that when we understand finally the structure and program, the feeling of mystery (and self-approbation) will weaken.

The models of creativity developed by AI scientists, psychologists, and cognitive scientists seem to fall into four classes: (1) combinatorial = generate and test; (2) heuristic search = push local test criteria into the generator, add additional global test criteria; (3) transforma-

Meta-DENDRAL Example

Find rules explaining analytic data from a class of known chemical compounds

E.g., Mass spectra from estriol and other estrogenic steroids can be explained by:

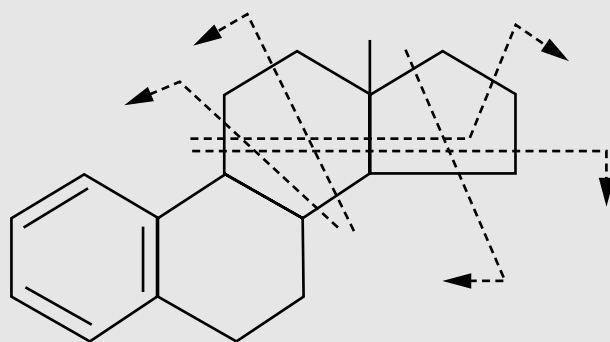


Figure 5. The METADENDRAL Program Finds General Rules for a Collection of Chemical Structures and Associated Mass Spectra.

A dotted line indicates that these chemical bonds would be broken. An arrow indicates which fragment carries the charge.

tional = add analogies and other transformations to the space of hypotheses; and (4) layered search = include bias space search at the metalevel.

These four classes of models suggest a structure that helps to organize the vast literature. There is considerable written discussion about generate and test with a combinatorial generator. As a baseline, it provides a good starting point for problem solvers and has been suggested as a starting point for people who feel a "creative block." For example, in lateral thinking, people are encouraged to list properties of objects, synonyms of concepts, and so on (deBono 1970; Osborn 1953). In divergent thinking (Baer 1993), people are encouraged to look for (we would say "generate") many alternatives from a wide range, including unusual ones.

Combinatorial search can be made smarter, however, by introducing heuristics into the generator. Providing some guidance and pruning before actual generation of the full space is the essence of heuristic search, which is the predominant problem-solving paradigm for AI programs. As an extension of the fundamental axiom that all problem solving is search, Newell, Shaw, and Simon emphasize that heuristic search might well be sufficient for explaining creative behavior. According to the

associationist model of creative thought in psychology (Mednick 1962), people are helped greatly in generating new ideas by thinking of concepts loosely associated with concepts in the problem description. Loose association either augments or replaces a strictly combinatorial generator of ideas.

The psychologist Johnson-Laird (1988) wrote that there were only three classes of algorithms that he could conceive for creative behavior (figure 6), which all are variations on the first and second search models. He added, "the three classes of algorithm exhaust the computational possibilities," which leaves out the last two of the four classes.

The first one, "Neo-Darwinian," he believes is not computationally plausible because it depends on unguided, random combinations. Genetic algorithms, however, have been shown to be an effective computational procedure for finding new solutions in seemingly creative ways (Holland 1992). The second class of models is essentially generate and test. The third is heuristic search, with the heuristics moved back up into the generator for guidance during generation. There is no mention here of either analogy or reflection at the metalevel, which I take to be important computational possibilities.

Weisberg, a contemporary psychologist,

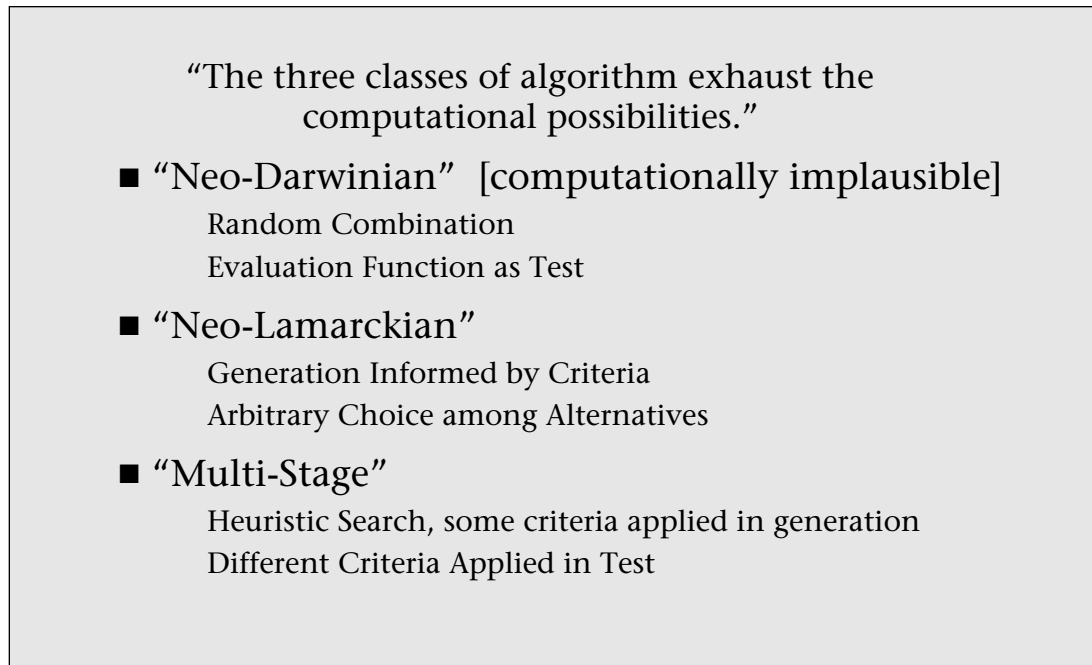


Figure 6. Johnson-Laird's Three Models for Creative Thought.

underscores the point that Newell, Shaw, and Simon were making—that creative problem solving is, after all, problem solving that makes essential use of background knowledge in the process (Weisberg 1999). In his model (Weisberg 1993), the same thought processes are going on in creative problem solving as in the ordinary kind but with some additional level of commitment that leads to the persistence that many people have mentioned. Thinking time, in his model, gets put into the background, becoming the incubation time that many psychologists have talked about. During this time, a person can incorporate a lot more background knowledge into problem solving, which most of us in AI would believe to be essential. Background knowledge, in the Weisberg model, then gives the person or the program the ability to explore a rich space much more efficiently. Overall, the model is one of successive modifications of ideas, generating successors by processes that are not elaborated. His emphasis on background knowledge, however, suggests both heuristic search and analogy as the generating mechanisms.

Feigenbaum's and Lederberg's model of hypothesis formation in science implemented in DENDRAL was a variation on heuristic search, which, as mentioned earlier, is sometimes called we have come to call the exclusion paradigm. In this model, the legal move generator is capable, in principle, of generating the complete space of possibilities with a combinatorial successor function. However, whole classes of

plausible hypotheses are excluded by the heuristics, which are based on general knowledge of the domain as well as the data of a specific problem. In principle, the program is exploring every plausible conjecture—either individually or as a class—and ruling out the ones that are not plausible. The scientist is left in a position of much more strength than if we use a generator that is not complete, where it is not entirely explicit what has been left out. Knowledge of chemistry and the data at hand guide the program to generate small numbers of hypotheses in the combinatorial space. The emphasis on background knowledge is even stronger and more explicit in this model than in Weisberg's.

By way of contrast, AM's search space of conjectures in elementary set theory was open ended and, therefore, not amenable to defining a complete move generator (Davis and Lenat 1982). One of the most interesting parts of Lenat's work was the introduction of a plausible move generator, which worked for AM partly because the space of interesting conjectures in this domain was very rich. A plausible move generator could find so many interesting things that the ones it left out became less important.

Another approach is offered by Boden (1994), who discusses three basic models for programs and people: (1) combinatorial, (2) exploratory, and (3) transformational, as listed in figure 7.

The combinatorial and exploratory models

- producing novel combinations of familiar ideas
[= combinatorial creativity]
- exploring complex conceptual spaces
[= exploratory creativity]
- transforming a conceptual space by adding
dimensions or dropping constraints
[= transformational creativity]

Figure 7. Boden's Three Models for Generating New Ideas.

are very much in the genre of heuristic search. We understand these models well, and what needs considerably more work is the transformational. She also talks about transformational creativity as involving analogical reasoning, for example, some transfer of concepts from one domain to another that allows some extension to the fixed space that the program is exploring. For example, it has been helpful to transform a discussion of biological mechanisms acting on the sequence of nucleotides in DNA from the domain of biology to the domain of string manipulation in computers. Under this metaphor, the repertoire of copying, matching, and editing procedures on strings suggests easily understood biological mechanisms.

Roger Schank's (1988, p. 227) model is a combination of search and alteration (close to Boden's transformation) involving three kinds of heuristics:

Creativity in a computer means supplying the computer with three types of heuristics, namely:

Creativity heuristics

1. Heuristics for the intentional reminding of XPs (explanation patterns)
2. Heuristics for the adaptation of old patterns to the current situation
3. Heuristics for knowing when to keep alive seemingly useless hypotheses

This model emphasizes analogy and transformation almost to the exclusion of combinatorial search.

Three sources of insights in creative thinking (figure 8) are offered by psychologist Robert

Sternberg (Sternberg and Davidson 1983), who has done considerable work on creativity and who has pulled together the literature into various cogent collections.

Insightful thinking comes from selective encoding, selective combination, and selective comparison (Sternberg and Davidson 1983). In AI terms, we would call these three sources correctly choosing an ontology, searching heuristically, and finding an effective evaluation function and criteria for ranking plausible hypotheses. Once again, the essential procedural element is search, but the choices of ontology and evaluation function might be seen as outside (that is, "meta" to) the search program.

The last of the four classes of models is layered search models. Although Sternberg recognizes the importance of choices at the metalevel, he stops short of suggesting that these choices can be made by search, which is the essence of layered search models. Newell, Shaw, and Simon talk explicitly about modifying the generator as a way of changing the framework within which search is conducted. Introducing new heuristics, for example, can make problem solving more effective, and seemingly more creative.⁶

McCarthy's (1958) advice-taker paper, *Programs with Common Sense*, is another early piece of wisdom that anticipated the layered search model:

In order for a program to be capable of learning something it must first be capable of being told it.⁷

That is, there is immense power in a declar-

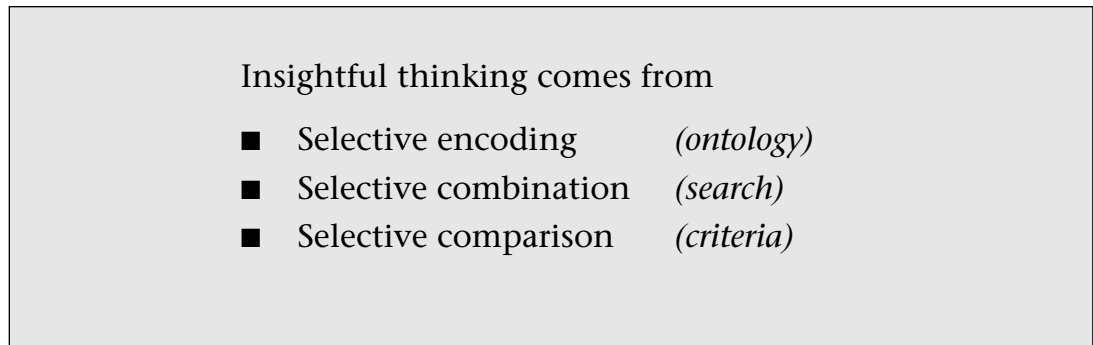


Figure 8. The Sternberg and Davidson Model.

ative, explicit representation of everything that the program knows, so that it can be changed from the outside. In this paper, McCarthy provides features that we can build into programs that allow them to be more intelligent, and we would say more creative, problem solvers. For example, all aspects of the behavior should be made as explicit and modular as possible, so they can be adaptive.

Arthur Samuel's (1959) work was a tour de force; no work in machine learning came close to Samuel's checker player for at least 25 years. A simple polynomial representation of the evaluation function for scoring positions on a checkerboard made explicit what needed to be changed to improve the program, namely, the coefficients in the polynomial. Samuel's approach was especially creative in that he introduced a second-level decision in which the program itself selected the terms in the polynomial that were used to evaluate checkerboard positions.

Lenat also implemented a layered search model in his EURISKO program, which automates the discovery and introduction of new heuristics, so that the program itself was making changes that allowed a problem-solving or game-playing program to be more effective. A central part of both the AM and EURISKO programs is Lenat's sets of heuristics for defining interestingness. Metalevel thinking also plays a large role in Hofstadter's (1985) writings on creativity.

In some of my own work at the University of Pittsburgh, and previously at Stanford University, I and creative collaborators have been looking at what it means to do metalevel reasoning to improve the performance of a problem solver or learning program. Randy Davis's dissertation research explored the power of explicit, metalevel reasoning (Davis and Buchanan 1977; Davis and Lenat 1982). Our early model of metalevel reasoning for learning

programs was encapsulated in Buchanan et al. (1978). Provost's research on searching a bias space for learning continues in this vein (Provost and Buchanan 1995), demonstrating that at the metalevel a program can decide, for example, what is the best set of features for learning.

Gary Livingston's (2001) dissertation program called HAMB extends the AM model of exploration into empirical science. HAMB's plausible move generator is a rule-induction program that suggests plausible generalizations from data. It is capable of generating an arbitrarily large number of conjectures with heuristics that guide the search for interesting sets of data, interesting attributes, and interesting attribute-value pairs to focus on as well as for interesting general rules. One of the most important features of HAMB is its ability to define which problems to work on by reflecting (at the metalevel) about the tasks and objects it might work on.

One preliminary result from HAMB is shown in figure 9. Starting with data from protein crystallography, namely, data about experiments that grow successful crystals of proteins and those that were failures, the exploratory question was whether the program could find any interesting conjectures that would promote the crystal growth of new proteins. It is a very large space. At one point, over 8000 conjectures had been generated that were plausible. The collaborating X-ray crystallographer was not willing to look through 8000 things, certainly not numerous times. Livingston introduced domain-independent criteria for what makes a conjecture interesting in science. For example, singularities and exceptions are interesting, and attributes that have a great deal of explanatory power are also interesting. Thus, using these domain-independent criteria, HAMB found 219 conjectures, about three-quarters of which were

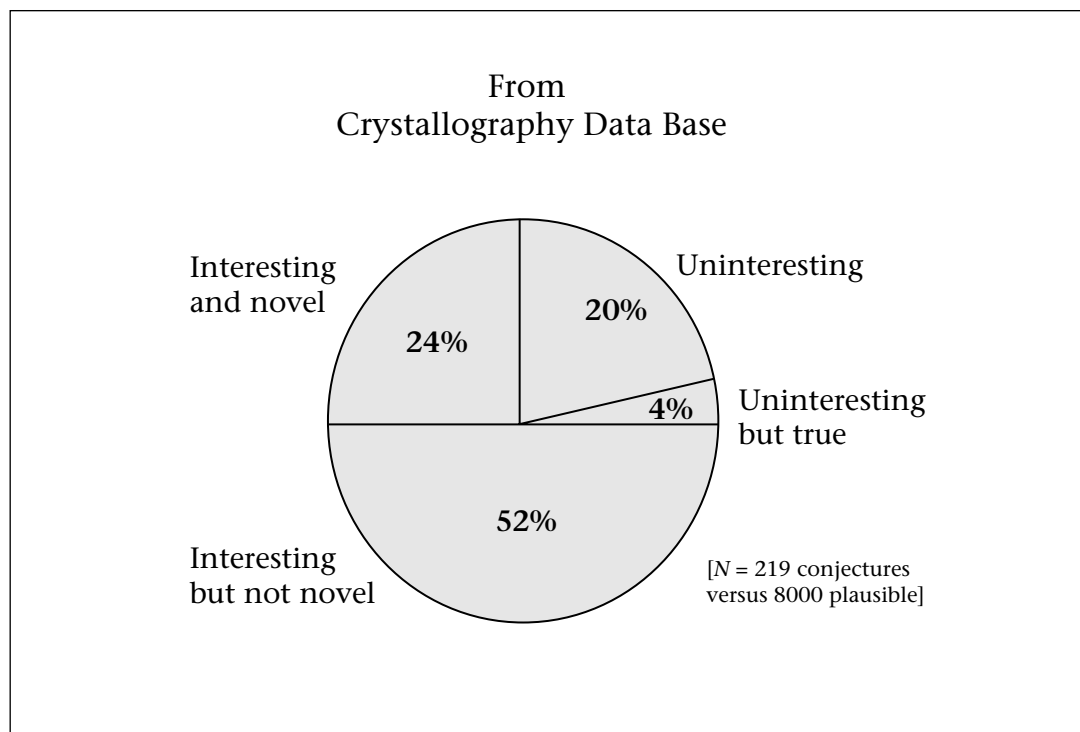


Figure 9. One Expert's Assessment of 219 Conjectures That HAMB Judged to Be "Interesting."

judged by this expert to be actually interesting.

Because HAMB's heuristics, like EURISKO's, are domain independent, this result can be repeated in other domains, which we are in the process of confirming. We are also working on criteria of novelty that encode relationships in existing models so they are not rediscovered (Ludwig et al. 2000). By generating three times as many interesting conjectures as not interesting ones, HAMB has encouraged us to believe a program can find interesting, novel, and useful hypotheses in science.

The schematic model of what I am suggesting as a key to creative problem solving, shown in figure 10, is that search at the metalevel gives us a means for identifying the choices that are most effective for performing a specific task. In the sphere of machine learning, this model is, essentially, what Mitchell (1997) and others call *bias space search*. The model of creativity at the metalevel is shown in figure 10. At the performance level, a program works within a fixed ontology, fixed criteria, and fixed methods. At the metalevel are libraries of ontologies and criteria and methods or generators of them. Searching multiple variations on the performance program might find just that variation of the performance program that will solve a class of problems most efficiently or with the highest-quality solutions—or at least find a satisficing alternative.

Challenges

The end of a millennium is a fitting time to take stock of what we have learned about creating creative programs. With respect to the language we use, there appears to be a great deal of variability in our use of the term *creative*. However, there also appears to be no mystery in this: Creative problem solving is problem solving (ditto science, mathematics, logic); creative art is art (ditto music, poetry, literature). Once we can define the criteria we would use ourselves to judge a person or a person's work to be creative, we can map them into programs.

With respect to the important abilities involved in creative work, cognitive scientists have identified several things: (1) knowledge, skill, and prior experience; (2) the ability to modify the ontology, the vocabulary, and the criteria that are used; (3) motivation, persistence, time on task; (4) the ability to learn from prior experience and adapt old solutions to new problems; and (5) the ability to define new problems for oneself. Some of these items we understand better than others, but there has been substantial research on all these items (Colton and Steel 1999; Simon, Valdez-Perez, and Sleeman 1997).

With respect to the methods used in problem solving, we certainly understand heuristic

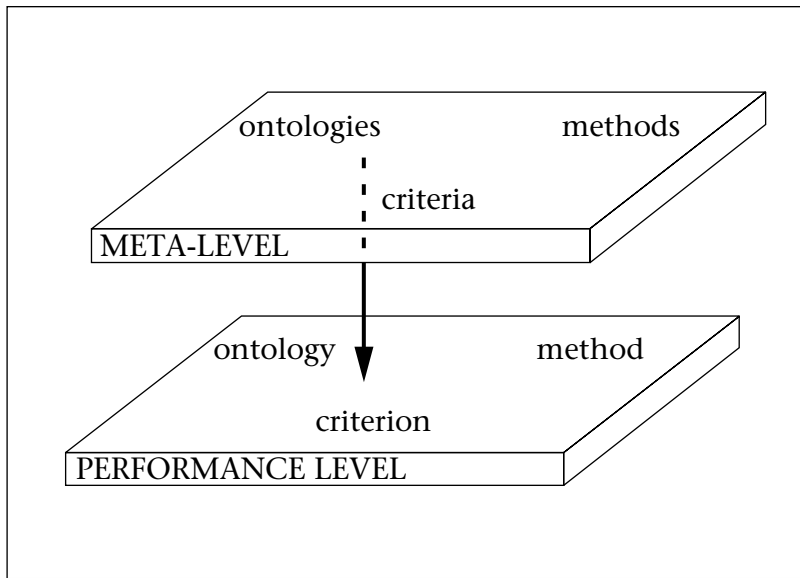


Figure 10. Creativity at the Metalevel.

search. The essential components that we need are a generator that combines the primitive elements into new ones, a goal test that determines when we are done, and an evaluation function that determines which partial paths are more likely to yield a solution.

Two main reasons account for my optimism for progress in creating more creative programs. First, in the last 50 years, we have come to understand the models of creativity that Boden calls combinatorial and exploratory, that is, heuristic search. Second, we have orders of magnitude more computing power available. It is not unthinkable to solve the same problem in thousands of different ways to find the best framework and assumptions empirically even when we do not know enough to choose the right ontology, evaluation criteria, methods, and so on, at the onset. To achieve more creativity in problem solving, however, it is also important that we can reason analogically and carry on a search at the metalevel.

We are poised to challenge the creativity bar once again. However, as we look at today's programs, it seems they fall short of what Boden and others describe in three important ways: (1) they do not accumulate experience and, thus, cannot reason about it; (2) they work within fixed frameworks, including fixed assumptions, methods, and criteria of success; and (3) they lack the means to transfer concepts and methods from one program to another. Therefore, I place the major challenges to creating more creative programs into three categories: (1) accumulation, (2) reflection, and (3) transfer.

Accumulation

Today's programs do not routinely accumulate experience or use past experience to accumulate knowledge. Art Samuel demonstrated the power of rote learning, yet we do not routinely include even this simple mechanism in our programs. Although it is the heart of case-based reasoning (CBR) (Leake 1996), the transfer aspects of CBR are still highly problem dependent. A few programs record and save users' preferences, but most do not accumulate knowledge of current practices and cultural norms. This knowledge is given within a given framework, which is our best guess as to the "best" one, and programs rarely are given the capability to modify the framework.

By accumulating knowledge of what is known and building models of the world, a program is better equipped to solve new problems. One difficulty in building a library of problems and associated methods for solving them is that we don't have a representation of problems and procedures that is both flexible enough to be changed and structured enough to be explicit. By accumulating and explicitly representing knowledge of cultural norms, including moral preferences and aesthetic criteria, a program reflecting at the metalevel can know, for example, that some solutions to problems are morally unacceptable, even though technically feasible. The fundamental problem here is understanding and defining the criteria for ourselves in the first place.

Reflection

Reflection is thinking at the metalevel; in psychology, it is referred to as "metacognition" (Baer 1993). It includes knowledge about tasks and problem-solving procedures, knowledge about strategies, and knowledge about applicability conditions for procedures. We now know enough about explicitly representing some parts of a program's knowledge to enable a second-order program to reason about it, which can take several forms, for example, (1) shift representation = transform the space, (2) introduce new ways to satisfy constraints, (3) introduce "just enough" randomness, (4) reflect on and change values, and (5) define new problems.

Shifting the representation has been described in the AI literature as a hard and essential problem for several decades (Amarel 1972). We have more computing power now than was available even 10 years ago, and we understand much better how a program can make some minor shifts in its representation. It is possible, therefore, for a program to look at the consequences of thousands of modifications to find

a solution to a problem. For example, finding a representation that transforms a hard problem into an easy one can be cast as metalevel search. With a library of millions of problems solved by computers around the world—plus the assumptions and methods used to solve them—a second-order CBR system would find numerous things to try.

When the constraints in problems are represented explicitly, it is possible for programs to reflect on new ways to solve them, which is the essence of much of the work on creativity in design (Brown 1993; Gero 1992; Goel 1997), pointing the way to finding novel ways of satisfying constraints in any problem area.

The element of randomness in problem solving is captured in the mutation operator of genetic algorithms (Holland 1992), for example. The rate of random moves often needs to be tuned from the outside, or by a second-order program. At the metalevel, this is another explicit strategic element. It can be changed again and again until there is a satisfactory result from the performance program. Insofar as random perturbations of an idea can lead to surprising and novel suggestions, genetic algorithms can produce creative products. The fact that random search is so inefficient in people as to be ruled out as an explanation for human creativity should not preclude its use by computers. But we do not need to fall back on totally random changes. Guiding the random changes by associations in a semantic net, for example, would be consistent with descriptions of lateral thinking, divergent thinking, or madness.

Values can also be considered to be high-level constraints on an activity. For example, different values are reflected in designing an automobile for fuel efficiency or road rage or nostalgia. In medicine, the relative value of true positive predictions against the cost of false positive errors influences much of medical practice. One point that Harold Cohen makes is that artists are considered to be more creative when they change the criteria used for judging their work. The impressionist painters, for example, rejected realism as a primary objective in painting and sought instead to convey an impression of the play of light, a change in values they made more readily than their academic critics. Reflecting on values at the metalevel will give programs new capabilities for creative behavior.

One of the strongest criticisms of good-old-fashioned-AI programs is that they only solve the problems they are designed to solve. METADENDRAL only constructed rules of mass spectrometry using planar graph descriptions of

chemical molecules within a fixed framework of chemical atoms and bonds. Cohen's AARON program only creates portraits within the framework he set. Both programs are capable of producing arbitrarily many variations within their frameworks, but neither one is capable of inventing a new framework. By searching through variations on conceptual frameworks at the metalevel, programs will partly overcome this objection.

Transfer

Transferring what is known in one problem domain to another one is not a common part of AI programs, although the state of our understanding has been advanced by substantial research in the last two decades. Representation, once again, is a key to improving the capabilities of programs. Four suggestions for experimenting with transfer are based on progress in the following four areas: (1) share ontologies, (2) use analogy engines, (3) import concepts from an old domain, and (4) modify previously successful methods.

It is difficult to see how one program can use the concepts and methods of another if their representational frameworks are radically different. Knowledge sharing is an active and productive area of AI research (Gruber 1993; Neches et al. 1991) that points the way to transferring knowledge from one problem area to another. Representing much of the critical information in the form of probabilities, as in Bayesian nets, gives programs a simpler representation than, say, structured objects do. Programs with simple, heterogeneous representations would seem to be candidates for straightforward applications of sharing ontologies routinely.

Gentner's work on analogical reasoning (Falkenhainer, Forbus, and Gentner 1989) and work by Winston (1980), Hofstadter (1985), Carbonell (1986), and Langley and Jones (1988) provide many insights into the computational mechanisms of analogy. In addition, work on case-based reasoning (for example, Leake [1996]) also offers working models of programs that can transfer concepts and methods from one problem area to another. These methods need to be routinely incorporated in problem solvers that we want to be more creative.

A program supported by a library of programs and their ontologies might well be able to look for new concepts to import. It need not invent wholly new concepts anew, if there is accumulated knowledge elsewhere that can help solve a problem. Kekule's importing the concept of cyclic graphs into chemistry (suggested by his famous dream of a snake chasing

... as we look at today's programs, it seems they fall short of what Boden and others describe in three important ways: (1) they do not accumulate experience and, thus, cannot reason about it; (2) they work within fixed frameworks, including fixed assumptions, methods, and criteria of success; and (3) they lack the means to transfer concepts and methods from one program to another.

its tail), for example, helped him explain the puzzle about the structure of benzene. We care less about the number of other concepts that did not work than we do about whether the puzzle was solved.

Similarly, with a library of methods suitably encoded, programs such as EURISKO can try changing a method to see if they get better results. One of the reasons AM and EURISKO worked so well was that their representation of methods as Lisp procedures was well suited to modification from the outside.

Conclusion

AI is ready for a renewed push on the important problem of creativity as we start a new millennium. Our goal should not be refining a definition, but rather demonstrating methods that achieve higher levels of creativity by some or all of the criteria already proposed.

Creativity is not a mystery; it does not require any noncomputable elements collectively called intuition or gestalt by some philosophers. It does require persistence, background knowledge, programming skill, and considerable experimentation, that is, creativity, on the part of the researchers.

I have tried to provide some background information and outline several areas where experimentation might lead to understanding how to build more creative programs. I would hope the experiments either demonstrate creative behavior in machines or suggest mechanisms that will.

I conclude with three questions. Once we know how to create creative programs, first, will they be able to help us with some of today's problems that seem to be impossibly hard? For example, can they create a user-friendly operating system? Second, what can they tell us about human creativity? Will we understand our own thinking processes better or gain deeper insight as to our own place in the universe? Third, how will we use them to benefit humankind? Will we be able to put all this effort to use in ways that alleviate human suffering and safeguard the planet?

The challenge is yours.

Acknowledgments

My deepest gratitude is extended to Ed Feigenbaum and Joshua Lederberg who taught me to think in computational terms and to numerous students whose creativity has inspired me. No one has caused me to think more about creativity in art than Harold Cohen. I am also grateful to the many friends who suffered my questions, to other colleagues who read earlier

drafts and made suggestions, and to those who suggested additional readings after this talk was presented. Jeremy Ludwig, in particular, was very helpful in discussing some of the landmark publications. Financial support was provided in part from grants from the National Library of Medicine (LM-6625) and the National Science Foundation (9412549).

Notes

1. T. Dartnall, 1994. Creativity. In *Dictionary of the Philosophy of Mind*, ed. C. Eliasmith. Available from artsci.wustl.edu/~philos/MindDict/creativity.html.
2. Sternberg & O'Hara (1999) survey the literature on the relationship of creativity and intelligence and conclude there is still a need for clarification. Taylor (1988) lists over 50 different definitions of creativity.
3. Motivation seems to be more of a problem for people than for computers. For more on motivation, see Collins and Amabile (1999).
4. It is not altogether clear that we apply the same criteria to products of artists, musicians, and writers as we do to the products of scientists and mathematicians. The former are visual works of art, auditory performances, or written words; the latter are abstract ideas. It is also not clear when we should be talking about products or people (and programs). Creative products must be novel and interesting; creative persons must either produce new and valuable products or have the capacity to produce them.
5. AARON has been made available as a screensaver, so a larger audience can now enjoy its unique drawings.
6. Newell, Shaw, and Simon also talk about the need for adaptation, anticipating to some extent some of Cohen's reservations about the AARON program. They point to the power of rote learning as in Samuel's checker-playing program. You can also read into their 1958 paper some early words anticipating SOAR's chunking mechanism, which routinely stores generalized solutions to subproblems during search.
7. Also see the articles that accompany this article.

References

- Amarel, S. 1972. Representations and Modeling in Problems of Program Formation. In *Machine Intelligence 6*, eds. B. Meltzer and D. Michie, 411–466. New York: Elsevier.
- Baer, J. 1993. *Creativity and Divergent Thinking: A Task-Specific Approach*. Hillsdale, N.J.: Lawrence Erlbaum.
- Bellak, L. 1958. Creativity: Some Random Notes to a Systematic Consideration. *Journal of Projective Techniques* 22:363–380.
- Berman, L. E. A. 1995. An Artist Destroys His Work: Comments on Creativity and Destructiveness. In *Creativity and Madness: Psychological Studies of Art and Artists*, eds. B. Panter, M. L. Panter, E. Virshup, and B. Virshup. Burbank, Calif.: AIMED Press.
- Boden, M. A. 1999. Computer Models of Creativity. In *Handbook of Creativity*, ed. R. J. Sternberg, 351–372. Cambridge, U.K.: Cambridge University Press.

- Boden, M. A. 1994. What Is Creativity? In *Dimensions of Creativity*, ed. M. A. Boden, 75–117. Cambridge, Mass.: MIT Press.
- Brinkley, J. F.; Altman, R. B.; Duncan, B. S.; Buchanan, B. G.; and Jardetzky, O. 1988. The Heuristic Refinement Method for the Derivation of Protein Solution Structures: Validation of Cytochrome-b562. *Journal of Chemical Information and Computer Science* 28(4): 194–210.
- Brown, D. C. 1993. Intelligent Computer-Aided Design. In *Encyclopedia of Computer Science and Technology, Volume 28, Supplement 13*, eds. A. Kent and J. Williams. New York: Marcel Dekker. Rev. Article available at www.cs.wpi.edu/~dcb/Papers/EofC-SandT.html.
- Buchanan, B. G. 1985. Steps Toward Mechanizing Discovery. In *Logic of Discovery and Diagnosis in Medicine*, ed. K. F. Schaffner, 95–114. Berkeley, Calif.: University of California Press.
- Buchanan, B. G.; Mitchell, T. M.; Smith, R. G.; and Johnson, Jr., C. R. 1978. Models of Learning Systems. In *Encyclopedia of Computer Sciences and Technology, Volume 11*, ed. J. Belzer, 24–51. New York: Marcel Dekker.
- Buchanan, B. G.; Smith, D. H.; White, W. C.; Gritter, R. J.; Feigenbaum, E. A.; Lederberg, J.; and Djerassi, C. 1976. Application of Artificial Intelligence for Chemical Inference XXII: Automatic Rule Formation in Mass Spectrometry by Means of the META-DENDRAL Program. *Journal of the American Chemical Society* 98:6168–6172.
- Carbonell, J. G. 1986. Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In *Machine Learning: An Artificial Intelligence Approach, Volume 2*, eds. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, 371–392. San Francisco, Calif.: Morgan Kaufmann.
- Chase, W. G., and Simon, H. 1973. Perception in Chess. *Cognitive Psychology* 4(1): 55–81. Reprinted in Simon, H. E. 1979. *Models of Thought*. New Haven, Conn.: Yale University Press.
- Cohen, H. 1999a. Colouring without Seeing: A Problem in Machine Creativity. *AISB Quarterly* 102:26–35.
- Cohen, H. 1999b. A Self-Defining Game for One Player. Invited Talk presented at the ACM/SigCHI Conference on Creativity and on Cognition, 10–13 October, Loughborough, United Kingdom. Available at acm.proxy.nova.edu/pubs/articles/proceedings/chi/317561/p14-cohen/p14-cohen.pdf.
- Collins, M. A., and Amabile, T. M. 1999. Motivation and Creativity. In *Handbook of Creativity*, ed. R. J. Sternberg, 297–312. Cambridge, U.K.: Cambridge University Press.
- Colton, S., and Steel, G. 1999. Artificial Intelligence and Scientific Creativity. *AISB Quarterly, Journal of the Society for the Study of Artificial Intelligence and the Simulation of Behavior*, 102:44–50.
- Colton, S.; Bundy, A.; and Walsh, T. 2000. Automatic Invention of Integer Sequences. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 558–563. Menlo Park, Calif.: American Association for Artificial Intelligence.
- Cope, D. 1998. A Musical Turing Test between the Works of Bach and Compositions by an AI Program. Invited talk presented at the Fifteenth National Conference on Artificial Intelligence, 26–30 July, Madison, Wisconsin.
- Dacey, J. S., and Lennon, K. H. 1998. *Understanding Creativity: The Interplay of Biological, Psychological, and Social Factors*. San Francisco, Calif.: Jossey-Bass.
- Davis, R., and Buchanan, B. G. 1977. Meta-Level Knowledge: Overview and Applications. In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, 920–927. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.
- Davis, R., and Lenat, D. 1982. *Knowledge-Based Systems in Artificial Intelligence*. New York: McGraw-Hill.
- deBono, E. 1970. *Lateral Thinking: Creativity Step by Step*. New York: Harper Collins.
- Falkenhainer, B.; Forbus, K.; and Gentner, D. 1989. The Structure Mapping Engine: Algorithms and Examples. *Artificial Intelligence* 41(1): 1–63.
- Gedo, J. E. 1990. More on Creativity and Its Vicissitudes. In *Theories of Creativity*, eds. M. A. Runco and R. S. Albert, 35–45. Newbury Park, Calif.: Sage.
- Gero, J. 1992. *Artificial Intelligence in Design*. Dordrecht, The Netherlands: Kluwer.
- Ghiselin, B. 1952. *The Creative Process*. New York: Mentor.
- Goel, A. 1997. Design, Analogy, and Creativity. *IEEE Expert* 12(3): 62–70.
- Gruber, T. R. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5(2): 199–200.
- Guildford, J. P. 1950. Creativity. *American Psychologist* 5:444–454.
- Harmon, L. R. 1956. Social and Technological Determiners of Creativity. In The 1955 University of Utah Research Conference on the Identification of Creative Scientific Talent, ed. C. W. Taylor, 42–52. Salt Lake City, Utah: University of Utah Press.
- Hofstadter, D. R. 1985. *Metamagical Themes: Questing for the Essence of Mind and Pattern*. New York: Basic.
- Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems*. 2d ed. Ann Arbor, Mich.: University of Michigan Press.
- Johnson-Laird, P. N. 1999. Jazz Improvisation: A Theory at the Computational Level. In *Representing Musical Structure*, ed P. N. Laird. San Diego, Calif.: Academic.
- Johnson-Laird, P. N. 1988. Freedom and Constraint in Creativity. In *The Nature of Creativity: Contemporary Psychological Perspectives*, ed. R. J. Sternberg, 202–219. Cambridge, U.K.: Cambridge University Press.
- Koestler, A. 1964. *The Act of Creation*. London: Hutchinson.
- Kuhn, T. 1970. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.
- Langley, P.; Simon, H. A.; Bradshaw, G. L.; and Zytkow, J. M. 1987. *Scientific Discovery: Computational Explorations of the Creative Process*. Cambridge, Mass.: MIT Press.

- Langley, P., and Jones, R. 1988. A Computational Model of Scientific Insight. In *The Nature of Creativity: Contemporary Psychological Perspectives*, ed. R. J. Sternberg, 177–201. Cambridge, U.K.: Cambridge University Press.
- Leake, D. B., ed. 1996. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, Calif.: AAAI Press.
- Lenat, D. B. 1983. EURISKO: A Program That Learns New Heuristics and Domain Concepts. *Artificial Intelligence* 21(1–2): 61–98.
- Lindsay, R. K.; Buchanan, B. G.; Feigenbaum, E. A.; and Lederberg, J. 1980. *Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project*. New York: McGraw-Hill.
- Livingston, G. R. 2001. A Framework for Autonomous Knowledge Discovery from Databases. Ph.D. dissertation, Department of Computer Science, University of Pittsburgh.
- Ludwig, J.; Fine, M.; Livingston, G.; Vozalis, E.; and Buchanan, B. G. 2000. Using Prior Models as a Measure of Novelty in Knowledge Discovery. Paper presented at the American Medical Informatics Association 2000 Annual Symposium, 4–8 November, Los Angeles, California.
- McCarthy, J. 1958. Programs with Common Sense. Paper presented at the Teddington Conference on the Mechanization of Thought Processes, December, Teddington, United Kingdom. Reprinted in 1968. *Semantic Information Processing*, ed. M. Minsky. Cambridge, Mass.: MIT Press.
- McCorduck, P. 1991. *AARON'S Code*. San Francisco, Calif.: Freeman.
- Masterman, M. 1971. Computerized Haiku. In *Cybernetics, Art and Ideas*, ed. Jasia Reichardt, 175–183. Greenwich, Conn.: New York Graphic Society.
- Mednick, S. A. 1962. The Associative Basis of the Creative Process. *Psychological Review* 69(3): 220–232.
- Minsky, M. 1963. Steps toward Artificial Intelligence. In *Computers and Thought*, eds. E. A. Feigenbaum and J. Feldman, 406–450. New York: McGraw-Hill.
- Minsky, M. 1986a. Genius. In *The Society of Mind*, ed. M. Minsky, 80. New York: Simon and Schuster.
- Minsky, M. 1986b. The Puzzle Principle. In *The Society of Mind*, ed. M. Minsky, 73. New York: Simon and Schuster.
- Mitchell, T. M. 1997. *Machine Learning*. New York: McGraw Hill.
- Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; and Swartout, W. 1991. Enabling Technology for Knowledge Sharing. *AI Magazine* 12(3): 36–56.
- Newell, A.; Shaw, J. C.; and Simon, H. 1958. The Processes of Creative Thinking. Report P-1320. Santa Monica, Calif.: The RAND Corp.
- Osborn, A. 1953. *Applied Imagination*. rev. ed. New York: Scribner's.
- Panter, B.; Panter, M. L.; Virshup, E.; and Virshup, B. 1995. *Creativity and Madness: Psychological Studies of Art and Artists*. Burbank, Calif.: AIMED Press.
- Polya, G. 1945. *How to Solve It*. Princeton, N.J.: Princeton University Press.
- Provost, F. J., and Buchanan, B. G. 1995. Inductive Policy, The Pragmatics of Bias Selection. *Machine Learning* 20(1–2): 35–61.
- Samuel, A. L. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3:211–229. Reprinted in 1963. *Computers and Thought*, eds. E. A. Feigenbaum and J. Feldman. New York: McGraw-Hill.
- Schank, R. 1988. Creativity as a Mechanical Process. In *The Nature of Creativity: Contemporary Psychological Perspectives*, ed. R. J. Sternberg, 220–232. Cambridge, U.K.: Cambridge University Press.
- Schank, R., and Cleary, C. 1995. Making Machines Creative. In *The Creative Cognition Approach*, eds. S. Smith, T. Ward, and R. Finke, 229–247. Cambridge, Mass.: MIT Press.
- Simon, H. E. 1979. *Models of Thought*. New Haven, Conn.: Yale University Press.
- Simon, H. A. 1967. Understanding Creativity. In *Creativity: Its Educational Implications*, eds. J. C. Gowan, G. D. Demos, and E. P. Torrance. New York: Wiley.
- Simon, H. A.; Valdez-Perez, R. E.; and Sleeman, D. 1997. Scientific Discovery and Simplicity of Method (Guest Editorial). *Artificial Intelligence* 91(2): 177–181.
- Sternberg, R. J., ed. 1999. *Handbook of Creativity*. Cambridge, U.K.: Cambridge University Press.
- Sternberg, R. J. 1988a. A Three-Facet Model of Creativity. In *The Nature of Creativity: Contemporary Psychological Perspectives*, 125–147. Cambridge, U.K.: Cambridge University Press.
- Sternberg, R. J. 1988b. *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge, U.K.: Cambridge University Press.
- Sternberg, R. J., and Davidson, J. E. 1983. Insight in the Gifted. *Educational Psychologist* 18:51–57.
- Sternberg, R. J., and O'Hara, L. A. 1999. Creativity and Intelligence. In *Handbook of Creativity*, ed. R. J. Sternberg, 251–272. Cambridge, U.K.: Cambridge University Press.
- Taylor, C. W. 1988. Approaches to and Definitions of Creativity. In *The Nature of Creativity: Contemporary Psychological Perspectives*, 99–121. Cambridge, U.K.: Cambridge University Press.
- Waltz, D. L. 1999. The Importance of Importance: Presidential Address, AAAI-98. *AI Magazine* 20(3): 18–35.
- Watson, T. A. 1913. The Birth and Babyhood of the Telephone. Address to the Third Annual Convention of the Telephone Pioneers of America, 17 October, Chicago, Illinois.
- Weisberg, R. W. 1999. Creativity and Knowledge: A Challenge to Theories. In *Handbook of Creativity*, 226–250. Cambridge, U.K.: Cambridge University Press.
- Weisberg, R. W. 1993. *Creativity: Beyond the Myth of Genius*. New York: Freeman.
- Wertheimer, M. 1945. *Productive Thinking*. New York: Harper and Row.
- Winston, P. H. 1980. Learning and Reasoning by Analogy. *Communications of the ACM* 23(12): 689–703.



Bruce Buchanan is university professor of computer science and professor of medicine, philosophy, and intelligent systems at the University of Pittsburgh. He received his Ph.D. in philosophy from Michigan State University in 1966 and has worked in AI since then. He is president and a fellow of the American Association for Artificial Intelligence, a fellow of the Medical Informatics Association, and a member of the National Academy of Science Institute of Medicine. His e-mail address is buchanan@cs.pitt.edu.