ARTIST'S ARTICLE

A Self-Defining Game for One Player: On the Nature of Creativity and the Possibility of Creative Computer Programs

Harold Cohen

ABSTRACT

The AARON program has been generating original artworks for almost 30 years, but is denied by its own author to be creative. The author characterizes creativity as a directed movement towards an ill-defined but strongly felt end-state for the individual's work as a whole, not as a characteristic of any single work and profoundly knowledge-based in the sense of externalizing the individual's internal world-model and system of belief. He suggests that a creative program would be one that was able to modify the belief-based criteria that inform the rule-base in which expert knowledge is represented, not one that is able simply to modify the rule-base itself.

I once made a joke at a cocktail party to the effect that I would be the first artist in history to have a posthumous exhibition of new work. I should be more careful of what I say at cocktail parties. The joke has been quoted—though not, need I say, actually discussed—much more frequently than anything serious I have ever said.

I was referring, of course, to the fact that my computer program, AARON, is currently capable of generating about a quarter of a million unique, original images every year from now to eternity and, with computing power increasing over time, could soon be providing several new, original paintings a year for everyone on the planet if it were, in fact, generating all those images on paper, which fortunately it is not. It uses a mechanical painting machine to generate output in the real world, and I don't see its real world output ever getting much beyond one large painting per day. A good thing, too.

Note that I used the term "new, original images," not "creative." I use the word "creative," on those rare occasions when I use it at all, to refer to the ability of the individual—human right now, program potentially—to move forward, to develop, to introduce new material. To put it more precisely, I believe the word properly attaches to continuous change, not to single events. There is no question that AARON has moved forward and developed over its 30-year existence, but the agency of change and development has been me, not AARON. Unless it can pick up from where I leave off, developing new knowledge and new levels of capability for itself, it will go on generating images that are original and different one from the other—in the sense that any two faces in the human population are different from each other—but nevertheless they are all the same in the sense that there will have been no further development, no new material introduced.

This leaves an open question then: whether we will ever be able to claim that a computer program is creative. What would be involved in giving a program that capacity? What is creativity actually like?

If we survey the work of any major artist, we get the distinct impression of someone who knew exactly what he or she was doing and knew exactly where he or she was going. Mozart always sounds like Mozart. Matisse did not produce a Picasso on Monday and a Miró on Thursday, he produced Matisses every day. No doubt his admirers anticipated that whatever he did would be unmistakable as a Matisse, but they could never predict by looking at what he did one day what he would do the next day or next year, when some unexpected new element or quality would eventually appear that did not follow so obviously from what went before that we—or, we may suspect, the artist—could have predicted it.

Michelangelo, according to some accounts, said that the figure was inside the block of stone and that all he had to do was to remove the superfluous material. Picasso said, "I don't seek, I find." There's a curious similarity to these attitudes in the popularized view of the scientist's "search space" as a place where all possibilities pre-exist and need only to be found. But it is unlikely that Michelangelo's genius lay in spotting the right blocks of stone, and the movies of Picasso working made back in the 1960s certainly made it look as though he was seeking something pretty hard. I will argue that creativity is not a random walk in a space of interesting possibilities, but that it is directed. The difficulty in tracking how the individual proceeds is that it is directed less by what the individual wants the single work to

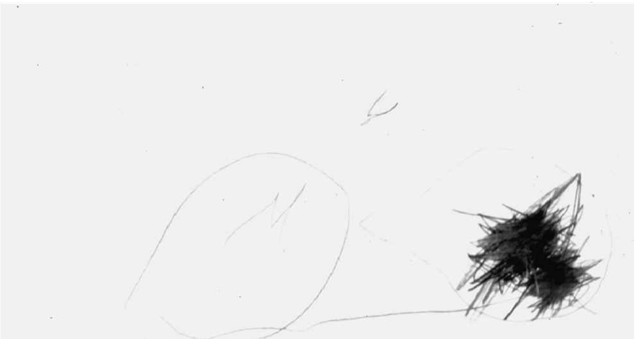

Fig. 1. Scribbles, drawing by Noah MacIlwaine, aged three.

Harold Cohen (artist, researcher, teacher), Center for Research in Computing and the Arts, University of California, San Diego, CA 92093, U.S.A. E-mail: <hcohen@ucsd.edu>.

An earlier version of this paper was presented at the third Creativity and Cognition Conference, Loughborough University, U.K., 10–13 October 1999, and published in the *Proceedings* (New York: ACM Press, 1999; ISBN: 1–58113–078–3). The complete text is also available on the web site of the Center for Research in Computing and the Arts at the University of California, San Diego: <http://crca.ucsd.edu\~hcohen>.

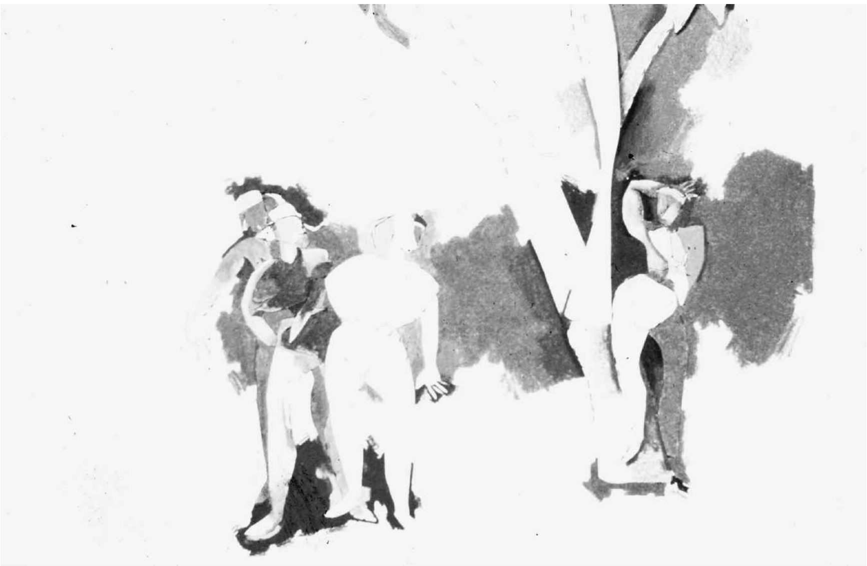

Fig. 2. AARON, drawing from *Bathers* series, colored pencil over laser print, 1985. (© Harold Cohen)

be than by what he or she wants the work as a whole to *become*. Becoming implies change and the introduction of new material. But it does not imply arbitrary change. While the ability to generate new material is, unquestionably, a necessary condition for creative behavior, it falls far short of being a sufficient condition. Nobody has ever considered the 12 poor exploited monkeys to be creative because they type material we have never seen before. That's about all they ever will type: we usually question whether they will ever generate something we *have* seen before; specifically, the complete works of Shakespeare or even a single sonnet. Plenty of new material, no significant new material; no way of having the monkeys observe the rules that would constrain the search space to the space of all legal sentences, for example, and certainly no way to have the monkeys dream up the rules for the production of world-class poetry.

How does the human being dream up the rules for the production of world-class poetry? Step by step. It seems to me that creativity involves moving through a series of intermediate states that are significant to the degree that they are increasingly closer approximations to some weakly defined but strongly sensed further-off goal. The creative individual does not find those approximations, because they are not there to be found; they don't exist until he or she constructs them.

In this view, what we characterize as creativity is likely to be most in evidence in the way the individual goes about constructing the successive approximations and in how the individual's internal model of the world is reformulated in the process. We aren't going to find a separate "creativity function." Indeed, it may never be entirely clear that what we find in the individual or what we build as a program is specific to creative behavior as opposed to normally intelligent behavior. It is only after the individual is seen to have gone part way towards that further-off goal that the accolade "creative" is granted.

But if trying to write a program's "creativity function" would be a waste of time, and I am sure it would be, what, exactly, should we expect to write?

Machine intelligence is not the same sort of thing as human intelligence. That has to be kept clearly in focus. Yet we now have ample evidence that machine intelligence can do many of the things we have conventionally assigned only to human intelligence. For all the protestations of those who want to reserve intelligence as a function of the wetware—the human nervous system—the fact is that Big Blue beat the greatest living chess player at his own game and, if it was not using intelligence in doing it, it was certainly using something pretty potent. Instead of asserting that a machine can't do this or that because it doesn't have the complex nervous system that enables its human owner to do this or that, we have to see, rather, whether we can devise strategies for doing this or that in terms of the resources the machine *does* have.

Over the past several years I've been able to endow my program with expertise in an area that might have been regarded as a paradigm of what computers can't do. Quite without the visual system that enables human artists to manipulate color, AARON has been performing rather well as a colorist (Color Plate A No. 2). What resources has it been using if not vision?

AARON's expertise is embodied in a set of rules robust enough to generate good results, as any expert system should, under a wide range of subject matter and compositional considerations. But since AARON can't see what it is doing, the rules themselves wouldn't be of much use, were the program not able to build and refer to a reliable internal representation of a complex and evolving image involving complex color relationships. Given the human colorist's reliance upon vision, it might seem understandable for the critic of computer programs to point to AARON's lack of vision as an insurmountable obstacle. But if I were a computer program with equivalent preconceptions, I might well conclude that the human colorist is hopelessly hampered by his miserable inability to maintain a stable internal representation. Visual feedback from what is happening on the canvas is a poor substitute for *knowing* what's happening, and it leads to the need for lengthy and continuous adjustment of color relationships; first the brightness of this patch here, then the hue of that patch there; no wonder the poor human artist almost never gets it right the first time.

In a more even-handed summary, we should say simply that the human artist has successive-approximation rules that require visual feedback from the external world, which the program doesn't have, while the program has dead-reckoning rules that require an internal representation that is beyond the capacity of the human being.

Now, obviously, AARON would never have acquired its expertise without my own experience as a colorist and had I not been able to reformulate the fruits of that experience in program-specific terms. But how AARON came by its expertise is a little irrelevant, because I would be no more capable of functioning as a colorist with the strategies I designed for AARON's resources than the program would be capable of functioning with the vision-resourced strategies I use myself.

While I am prepared to claim that AARON is a capable colorist I do not claim it to be a creative colorist and nothing would support such a claim, in my view, without the recognition that the program had, of its own volition, moved color-use up a notch from what I had supplied. But to get some idea of what must be a pre-condition for creativity, consider the following brief account of the steps involved in giving AARON its expertise:

First, formulation of a theory of color-use that sought to explicate, in general terms, what human artists use color *for.* (The decision to focus on "*for*" rather than "*how*" was critical to everything that followed.) Second, consideration of how human beings develop strategies based on their own resources to satisfy their theoretical requirements. Third, speculation about how a visionless program could satisfy similar requirements. Fourth, knowledge about color perception derived from my own experience as a painter. Fifth, and deriving from the same source, considerable knowledge of how different coloring materials behave. Sixth, and to supplement this general knowledge, extensive data concerning the particular properties of the colored dyes the program would be using—what is seen on the screen is in fact a simulation of a large-scale painting machine AARON has used in museum exhibitions—especially with respect to the effect of dilution on different dye mixtures. And finally, seventh, the construction of a robust rule-set for generating color palettes for individual images and for assigning individual colors to the different components of the image.

Evidently the program is heavily dependent upon the knowledge I was able to assemble; and I am quite sure that knowledge, and lots of it, is essential to creativity. Presumably the rules express the knowledge in some way; how else could the program perform at an expert level? Yet the fact remains that the program itself is just a set of rules. What can it mean to say that the expert knowledge is implicit in the rules?

Production rules, as we call them, are statements of the form: "If such and such is the case, do the following"—a predicate is followed by a consequent. There must be at least two different kinds of knowledge, then, that are available to the rules but are, nevertheless, external to the rules themselves. There would be no point in defining a consequent action unless the program knew how to perform it, and to define the predicate is to presume that the program can discover enough about the state of the world to decide whether various criteria are satisfied.

It would seem, then, that knowledge on the predicate side is expressed through the criteria that need to be satisfied; but what exactly are criteria?

A movie critic writes, "So-and-so's new movie simply doesn't sustain one's interest for two hours," or "This director's work indicates a singular lack of moral awareness." Criteria are not just standards to be reached; they are standards to be reached *with respect to particular issues.* The critic's comments imply that a movie *should* retain the viewer's interest, that the director *should* be aware of the moral issues raised by his or her own work. But the critic rarely bothers to state explicitly *why* that should be.

Criteria—standards to be reached with respect to particular issues—are almost invariably implicit in rule-based systems, whether those systems are operated by people or by programs, and it is rare for more than the most immediate level to be in evidence: the rest of the hierarchy, upon which the criterion immediately to be satisfied rests, is hidden. "If the sauce starts to separate, do the following . . . " is a rule, but the state of the sauce is the subject of the predicate, not the criterion. The criterion, which the cookbook author never bothers to state explicitly, is that sauces should have a smooth consistency and should remain smooth throughout their production. And nobody but a child would dream of asking *why* sauces should have a smooth consistency.

Inevitably, the writers of rules make certain assumptions about the issues in relation to which their rules attempt to define standards. We are all assumed to know what the issues are and we are all assumed to agree that they are important, to the degree that they never need to be discussed. That may be fine for the critic, but is a good deal less so for the creative individual, who moves forward into territory that is, by definition, unconstructed, unpredictable. And for anyone trying to write a creative program, the distinction between a rule and what implicitly informs the rule, between a predicate and a criterion, is critical. One can write explicit rules; there is no special reason why a program should not modify them, other than the fact that what would *require* the rules to be rewritten would have to be a shift in the program's criteria, which are not explicitly stated in a rule-based program any more than they are explicitly stated by the critic or, for that matter, by the human artist.

Which means, briefly, that the determining factor in whether a program can be creative is not its ability to modify its own production rules, but its ability to modify the criteria that implicitly inform those rules and thus provide the grounds for their modification.

One might say much the same thing regarding the human artist. To a large degree such criteria are culturally determined; we do agree on most issues, and it is rare for the work of a single individual to enforce a major shift of criteria for the entire culture. At the same time, creative behavior must involve an increasing differentiation, in whatever degree, of the individual's criteria from those of the culture at large.

The process of acculturation begins almost as soon as the child leaves the womb, and my guess is that individuation begins not much later, by the time the child is able to abstract from experience enough to construct something very like a rule-based system for itself; in the transition, that is, from the observation "When I cry Mummy picks me up" to the rule "If I cry, Mummy will pick me up." The connection between behavior and result—predicate and consequent—gets less simple as the child gets older, but anyone who has raised a young child will surely have concluded that the criteria that drive its rule-based systems are related to maximizing the child's influence over its environment to its own advantage; to increasingly differentiate itself, that is, from the environment.

My own young child, Zana *[aged 4 at the time of this writing—Ed.],* acquired drawing skills rather early and for several months elicited praise (read: maximizing influence over environment) by spending almost as much time showing us her latest drawings as she did in making them. Eventually goals change: now she draws for long periods without showing the drawings to anybody and discards them as soon as they are done, using drawing both to enlarge her range of capability and to record and to review her own experience.

The development of drawing skills has some affinity with the development of language, in that the child begins by using set verbal forms and only later is able to use language as a general tool of potentially infinite expressivity. Similarly, as a child's technical resources and skills improve it will be her *conceptual* grasp of the external world, finally, that limits what she can do.

To paraphrase Wittgenstein, we only know what is in our heads by the images we make; the image serves as a reality check, allowing us to compare what we believe against the outside world and thus enabling appropriate modification of those beliefs.

It is clear, of course, that knowledge does not mean a collection of facts, but a complex internal model that one builds to represent the external world. And it is equally clear that what can be externally represented depends upon a complex interplay between the patterns of knowledge constituting the internal representation and what representational skills the individual can call into play. That is a general statement about the way the mind works and it does not tell us anything about creative activity in particular. But it does describe a condition that has to be met before creativity is possible, no less for a computer program than for a human being, regardless of the resource-determined differences of implementation we know must exist between the two.

Can a computer program satisfy this condition? I want to show, without implying anything beyond what can be demonstrated, that AARON does satisfy it; that the images it makes of an external world do reflect its internal representations of that world and its representational skills. And that, in fact, the representational skills themselves—AARON's drawing ability—result directly from the nature of the internal representation.

In the early days, I tried to base AARON on what I had seen my own children doing when they were very young; scribbling, specifically at the stage at which a round-and-round scribble migrates outwards and becomes an enclosing form. Eventually, the closed form takes on an independent existence for the child—no longer requiring the original scribble—that can then serve as a basic element in representing objects (Fig. 1). However, AARON still today has to draw the scribble, or what later replaced the scribble, "in imagination" as it were, in order to generate the closed form.

At some point, then, AARON's development diverged from that of its human archetype, and I was obliged to devise strategies for an increasingly alien entity that could generate results that would not be alien to the human viewer. That caused problems. For example, as AARON moved on from its early "primitive" phase to explicitly representational drawings of people and plants, its knowledge about what it was drawing continued to be carried exclusively by the core figure from which outlines are generated—no longer scribbles now, but internal structures articulated in response to posture. There was nothing else for it to relate to, of course; vision-less as AARON is, surfaces do not exist for it as they do for us, as the physical boundaries of real-world objects reflecting light to our eyes. And without any concept of surface, the program also has no conception of occlusion and no way of dealing with occlusion using any of those fundamentally photographic simulations—hidden-surface removal, ray-tracing, whatever—that support computer graphics. I was obliged to develop a quite non-standard method for handling occlusion. AARON always draws only from front to back, it never uses an eraser and it infers what parts are closest in a figure from what it knows about the figure's posture. All of this follows directly from the single fact that AARON has no visual system.

The relationship between what the program knows and how it goes about drawing it becomes even clearer if we consider what happens if the program decides to draw two figures. What does it need to know about where it had placed the first figure when it then draws the second one?

That isn't as simple as it sounds. To begin with, in any representation of the physical world dealing with where things are, the word "where" has, simultaneously, two distinct meanings. It refers to the location of the representation of the figure within the frame of the picture and also to the location in the real world of the real figure that gives rise to its two-dimensional location on the picture plane. However, the two are not necessarily compatible, in that a perfectly reasonable distribution of objects in the real world does not necessarily result in a satisfactory distribution on the picture plane. We tend to think of perspective—and by extension photography—as a model of vision, but in fact there is no direct way of adequately predicting what the picture will look like from what the world looks like. That is why Polaroid is able to sell instant film to professional photographers; they need to know what the picture will look like while they are still in the process of arranging their models and their lights and their camera tripods.

I suppose I understood all that well enough when AARON became overtly representational, but without having then developed explicitly compositional strategies I told myself that, since anything we see through a window looks perfectly reasonable, a simple perspective rendering of that frame should be equally reasonable. It did not take long for the resultant drawings to show me how wrong I was. At that stage, AARON had only a sort of two-and-a-half-dimensional model of the figure, which it would construct from the head down, just as any human artist would do. Individual figures looked plausible until I attempted to place them within a physical context, whereupon the program's lack of control over where the feet finished up became quite evident. With embarrassing regularity I would find people treading on each others' feet, or a foot halfway up a tree that was supposed to be behind the figure (Fig. 2).

I responded to that problem in part by providing AARON with a fully 3D model of the figure and in part by having AARON construct its figures from the feet up, ending with the head: something a human artist would not do. In planting the feet firmly on the 3D ground-plane, I solved the misplaced-feet problem, but found that I had introduced a new one in its place: namely, the placement of the head was now largely a function of posture and thus independent of compositional considerations. The result was that two figures that had been side by side in the real world would frequently end up with one head partially hiding the other.

By this time it was becoming clear that the problem of placing representational elements in the picture could not be solved by placing the real-world elements in the real world and then rendering them in perspective. And, of course, a little reflection revealed that it never had been solved that way by human artists. Images utilizing fully developed perspective have always been preceded by sketches, the primary purpose of which has been to consider and manipulate the distribution of elements *on the picture plane* independently of where the elements might have been in the real world. That was true even during the High Renaissance, when "correct" perspective was mandatory and rectangular solids—buildings, tessellated floors, furniture and so on—were habitually constructed by rule rather than by observation. AARON, conversely, was not considering composition at all; the elements were rendered in perspective and the possibility of a fine-grain compositional relationship of those elements to each other was cancelled out by the vagaries of posture.

The next phase of AARON's development thus required a significant increase in its knowledge of the picture plane and how to place things in relation to it. That may sound like a non-issue—after all, what is there to say about a picture plane? In fact—and as I've tried to indicate—there is a big difference between an empty frame that is held up to the world to receive whatever happens to be out there, with the bits falling where they may, and a rectangular space whose very rectangularity plays a dynamic role in the artist's determinations; because if the artist has control over the way elements are placed in relation to the frame, he or she also has a good deal of control over the way the image is read and understood by the viewer. I do not mean composition here to imply the various geometrical strategies that have been used by a few artists from time to time—the oft-cited but rarely used Golden Section, for example—for placing distinctly non-geometrical entities like heads and bodies within the frame. AARON's problem was not that it didn't know anything about the Golden Section, but simply that it did not know how to find space for what the viewer was supposed to see.

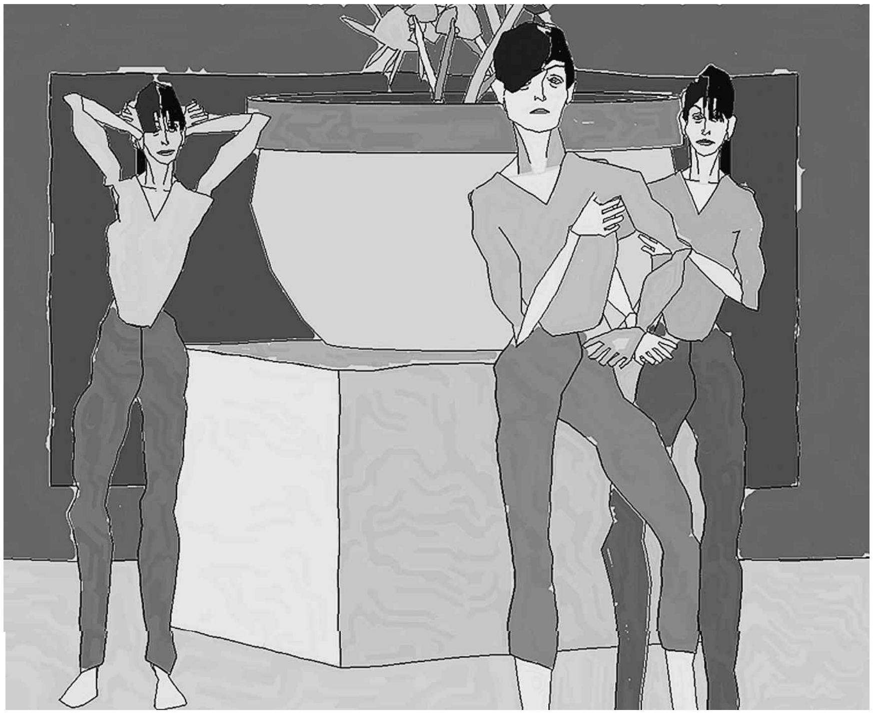

Fig. 3. AARON, screen image, 2000. (© Harold Cohen) This is a recent image demonstrating the program's improved control since this paper was written in 1999.

In its current version, AARON comes much closer to the human way of doing things. It still constructs its figures as 3D objects viewed from a position within their own space, as in an orthodox perspective; plausibility requires that much. But it also has the freedom to manipulate the placement of the figures in relation to the frame. Not simply by shifting the elements around, because the sizes the figures are to have in the picture are determined by where the program wants them to be and how much of the frame it wants each one to occupy, while their *relative* size, as determined by their distance from the viewer in 3D space, still needs to be at least approximately maintained if plausibility is not to be sacrificed.

In practice the core figure is posed, moved into the 3D space in relation to the viewer, rendered in perspective and then scaled to fit whatever space on the picture plane the program wants to assign. The outlines are generated only after the figure is in its final position. The difference between a head-and-shoulders portrait and a full-length figure is then principally a difference in scaling: in each case, the bridge of the nose may be placed roughly three-quarters of the way up the frame, say, while either the top of the sternum, for the portrait, or the feet, in the case of the full-length figure, is assigned to somewhere close to the bottom of the frame.

When the program comes to draw a second figure, which is further away in the real world and thus smaller in the original perspective, it uses a slightly different strategy. Instead of deriving the scaling factor from two points in the figure and where it wants them to be on the picture plane, it uses only one point and modifies the scaling that was used for the first figure according to how far away the second figure is supposed to be. In this way, it is able to keep the size relationship consistent with that of the perspective rendering, even though neither figure has the size or placement of the original perspective (Fig. 3).

For the vertical component of the composition, then, the program needs only to know the scaling and the placement of a couple of key points in the first figure in order to place the second figure plausibly. But the vertical component is only half the story. The other half—the horizontal placement—is less simple, and the current implementation leaves something to be desired. The problem is that the closer the two figures get, the more the program is limited through lack of knowledge. It certainly could not decide, for example, to place the second figure with its right ear just to the right of the first figure's left ear, for the perfectly trivial reason that it does not know where the first figure's left ear is. A strange reason! Obviously it knew where it was when it was drawing it. Why doesn't it know now? Quite simply, the program cannot—or rather, does not—remember. There are now almost a thousand control points in AARON's prototype figure, and each point is represented by a data structure requiring about 15 words of memory. That's in addition to almost three megabytes allocated to the matrices on which the program records the changing state of the drawing as a whole. AARON dates from a time before memory was the cheapest part of a computer, and these 15,000 words have always been recycled after each figure is completed to make room for the representation of the next figure.

Computer memory is cheap now, and there's no longer any reason, beyond my own disinclination to go through a megabyte and a half of Lisp code to make all the necessary changes, why the program's internal representation should not retain all the information it accumulates and a good deal more. That's a perfectly trivial reason, since apparently it involves nothing much more than bookkeeping. Yet it provides a very graphic illustration of the way the program's limitations can be traced back to the most fundamental character of its internal representation and of its representational mode. Its design sprang from the scribbling behavior of young children and the resultant need to construct an internal core in order to generate a closed form. Since AARON could generate closed forms this way, the need for visual perception—impossible in any case—was bypassed and with it the more obvious, surface-dependent methods of handling occlusion. The program was then obliged to infer where occlusions would occur from what it knew about the figure, and the volume of what it knew grew steadily to satisfy the needs of this inferential method. But this method figured as an alternative to perspective, and the program reached a limit at the point where it attempted to go beyond perspective, only to find that it no longer had what it needed, having discarded it along the way.

Actually, I am not unreasonably lazy about making changes—I have re-written the program from the ground up, in different languages and on different platforms, a dozen times. If I haven't made the changes to provide AARON with better memory capabilities, it is because I think the time has come to provide it with an entirely different *level* of memory. Currently, the program's knowledge—as opposed to the accumulated knowledge in my head—is limited to the drawing it is making. It cannot make any determinations on the basis of its own past history—another way of saying it cannot learn from its own experience—unless it has access to that history, and it has no archival memory to access. But what is the program supposed to remember? Obviously not the simple verbal description—two women with potted plant, or whatever—and not the matrix representation of the final image. There is nothing to be gained by committing to memory knowledge of past performance if AARON cannot subsequently use that knowledge. It is clear that remembering and recalling have to remain tightly coupled. But that's about all that is clear to me about how to proceed.

I remarked at the outset that, unless AARON can pick up from where I leave off, it will go on generating images that are original only in a very limited sense, and that no claim could then be made for the program's creativity. Evidently, there is a long way to go before AARON can go on to eternity producing new, original images generated out of autonomously directed, creative behavior. I do not think it makes any sense to talk about creative behavior that is *not* autonomously directed and, increasingly, I find myself thinking about the program's autonomy rather than its creativity. But now I wonder how much autonomy is enough. I used to define what I meant by saying that I wanted the program to be able to do drawings in August that it could not have done when I stopped working on it the previous January. That test no longer satisfies me; it correctly captures the need for self-modification, but not the need for *directed* self-modification; and I suspect that this idea of a test grew from a notion of absolute autonomy that has lost whatever appeal it may once have had for me. I think of autonomy now in terms of that weakly defined but strongly felt future state that manifests itself in the criteria that direct creative behavior. And if I take some satisfaction in being able to formulate it in those terms, I also recognize that the criteria and the creativity in question are mine, not the program's.

I began this essay with a joke about being the first artist in history to have a posthumous exhibition of new work. Maybe. I have to confess to feeling less like a potential immortal than like Moses looking out over the promised land he will never enter. The biblical Aaron didn't either, if I remember correctly. I don't mean that I think the final stages of AARON's development are out of reach, though they certainly may be out of *my* reach. I mean that if AARON ever does achieve the kind of autonomy I want it to have, it will go on to eternity producing original AARONs, not original Harold Cohens. Apparently the joke is on me.

Professor Emeritus and Senior Research Professor Harold Cohen came to the United States from London in 1968 with an international reputation as a painter to take up a 1-year visiting professorship in the Visual Arts Department at the University of California, San Diego (UCSD). He met his second wife and his first computer and never returned to the U.K. In the early 1970s he spent 2 years as a guest at Stanford University's AI Lab, where AARON was first conceived. Almost all of his exhibitions since that time have been "live," with AARON generating art in real time, with output devices ranging from simple drawing machines to complex painting machines.

Cohen was responsible for introducing computing to the art curriculum at UCSD and subsequently became the founding director of the Center for Research in Computing and the Arts (CRCA) at UCSD. Retired now, he spends his time working on the AARON program, painting, writing and admiring his six-year old daughter, at his home in Encinitas, California.

The AARON program has been the subject of many television programs and is now available as a download from the site of Kurzweil CyberArts Technologies.