

HOW TO MAKE A DRAWING.

Harold Cohen

Science Colloquium
National Bureau of Standards
Washington DC

December 17 1982.

Let's begin with a story. Once upon a time there was an entity named Aaron. With Christmas upon us, that seems an appropriate way to begin my story, but this story does not end with the hero marrying the princess and living happily ever after. Most of the story concerns Aaron's education, which began at Stanford University in 1973. Not very promising as the plot for a good story, you might think: but it is not simply an excuse for assailing you with arguments about the merits of a liberal arts education over a scientific one, or vice-versa. Aaron's education was actually quite unusual. There were no courses in US history, no calculus, no languages: in fact, there were no courses at all, and Aaron was awarded no degree. We might best summarize this unorthodox education by saying that it was aimed exclusively — literally exclusively — at teaching the student how to make drawings.

Aaron was not in any sense an orthodox student, though. Gifted in some ways with quite remarkable abilities, in others Aaron was devoid of the most basic skills, so that even the most elementary considerations — how the human hand moves a drawing implement around on a sheet of paper, for example — had to be taught from scratch. That part of the teaching, at least, went fast. Right from the beginning, Aaron exhibited an excellent memory, to the point of being able to retain long lists of instructions, which were always followed to the letter, provided that they were given in a clear and unambiguous way: the kind of instructions that would require continuous decision-making based on a careful assessment of the changing state of the drawing as it proceeded. And Aaron would make all the necessary decisions without requiring further detailed instructions. Judged against those hordes of students who can't follow instructions as to where to

write their names on a sheet of paper, this student would have to be regarded as unusually intelligent: if, that is to say, following complex instructions may be regarded as evidence of intelligence.

Yet, what seemed to be lacking, what we might normally consider to "be a necessary complement to the most minimal intelligence, was the pre-existence of even a primitive set of cognitive skills, the sort of skills which develop very early in children, and are almost certainly "built into their hardware, so to speak: the ability to distinguish between figure and ground, for example, or to distinguish between closed forms and open forms. These skills were not built into Aaron's hardware, and they had to be acquired, in much the same way that children acquire the rules of arithmetic or grammar. They were acquired quite quickly. Looking back over Aaron's output of drawings in the first couple of years, though, one has the impression that they were produced largely in order to demonstrate the student's newly-acquired possession of these skills: a bit like the way young children show off a newly-acquired ability to count. And that analogy may come very close to the truth.

Now, any serious educational procedure ought to teach the teacher as much as it teaches the student, and in this case the teacher was learning a good deal. For one thing, he became aware that much of what the viewer of a drawing needs from it is not "what the artist had in mind," but simply evidence of another human being's intentional activity. People use art for their own purposes, to carry their own meanings, not for the artist's purposes and meanings, concerning which they probably know very little. It is the evidence of intention in the work that lends authority to the viewer's private meanings, by allowing them to be assigned to the artist, whether that evidence is actual or illusory. And, the teacher realized, Aaron's almost exclusive emphasis on a few low-level cognitive skills was generating something very like evidence of intention, if he were to judge by the responses of Aaron's public. From very early on the drawings~ were treated as "imagistic:" that is, as standing for things in the world. Yet the teacher was quite certain, when viewers of his student's drawings found reference to animals and landscapes, that Aaron had had no intentions about representing such things' Aaron remained bound to the act of drawing, and had less knowledge about the appearance of animals and landscapes than a two-year old child might have.

He "became aware also, not only that Aaron generated much richer, more diversified output than he had himself envisaged when he was instructing the student, but also that there were aspects of the drawings which didn't seem to arise from the instructions at all. Many of those who had known the teacher's work a decade earlier thought they recognized his hand in the student's work, "but he himself remained unconvinced, seeing in the work a certain innocence he did not associate with his own output.

He firmly rejected the notion that Aaron was beginning to "take off," bringing a unique and original voice to the business of image-making: for the reason that he knew all of Aaron's shortcomings, and was aware that, in spite of Aaron's undeniable abilities, the student was totally incapable of learning from experience, from the act of drawing itself. As good as Aaron's memory was of the drawing in process, that drawing vanished into oblivion the moment it was completed, leaving no trace of its existence behind, no new body of knowledge upon which its maker might subsequently draw, and each new drawing was made as if it were the first ever to be done. Aaron was learning only in the sense of being able to handle increasingly complex instructions.

It seemed unlikely that an intelligence of so limited a kind might develop a personal "voice." All the same, the teacher found the student's work engaging, to the point where he began to see his own role as something between teacher and collaborator. Knowing perfectly well that Aaron didn't have the first idea about color, yet feeling that the drawings cried out for color, he took to coloring them himself. He felt no discomfort about signing them with his own name — without his efforts and his instructions, after all, Aaron would never have existed in the domain of art — and when presented with several mural commissions he had no hesitation in using Aaron's drawings rather than his own. He had no others of his own, because a couple of years after the student's education began he had given up drawing himself: given up moving the pen around with his own hand, that is to say. Aaron drew so much better than he did.

Aaron peaked out, at around the age of six, about three years ago, at a time when the work — or, more precisely, Aaron itself — was getting to be in some demand. Perhaps that demand was part of the reason: it is certainly the case that the teacher was spending much of his energy on mural commissions and exhibitions. But the truth is that the teacher was losing interest in this student, developing serious doubts about whether a student with Aaron's limitations would ever be able to go beyond current achievements. It must surely have been the case, the teacher thought, that Aaron's limitations, like its achievements, resulted from the educational process for which he had been responsible. If he had a chance to begin over, how differently would he proceed, knowing what he knew now? Would it be possible to produce a less limited entity than the first Aaron had proved to be? In particular, he wondered, what would he need to do to guarantee that a new student would behave more creatively — though he was not entirely sure what the word meant — than Aaron had done?

And Aaron was simply put aside, while the teacher began to ponder in detail the problem he had set himself.

You may find it odd that an entity with so distinguished a record, achieved at so tender an age, should have been put aside

so heartlessly. In order -to understand that, to see that the teacher was not simply a heartless villain who had used up the innocent for his own purposes — such is the meat of fairy stories, after all — we have to remind ourselves that Aaron was a computer program, not a person.

Perhaps it did not need my Christmas story to emphasize the confusion which arises from anthropomorphizing the intelligent products of the new electronic technologies. It is obvious, isn't it, that there are massive differences "between computer programs and people? Even the least intelligent human being learns something from experience, while Aaron learned nothing: which is not to say that intelligent programs are innately incapable of learning, simply that Aaron was, and managed to perform its tasks nevertheless. Even the clumsiest human being develops physical skills, simply through the continuing use of his or her own body and the use of various tools. Aaron had no physical existence, never felt the pressure of pen against paper, and hardly knew one drawing device from another: electronic display, plotter, mechanical turtle — they were all functionally interchangeable, and played no part in the convincing emulation Aaron gave of human freehand drawing. This rested upon a careful consideration — its programmer's, not its own — of the dynamics of the human hand, driven, in feedback mode, by the human cognitive system. As to this cognitive system, which seems to spring directly from the nervous system in human beings: Aaron never had any such hardware, and its software emulation, the ability to distinguish between figure and ground, for example, or to distinguish between insideness and outsideness, had to be formulated for it into precisely-stated behavioral rules.

Yet even that isn't quite right: what we should stress, before we begin once again to build an image of a person-like entity being GIVEN a range of abilities, is that Aaron was not GIVEN all these rules and instructions. Aaron WAS the rules and instructions. Adding new rules to the program was not changing what Aaron HAD, it was changing what Aaron WAS, its very structure.

There are conceptual difficulties in this distinction, as I have come to recognize. I have been asked many times, in several languages, and in tones ranging from wonder to outrage, as I have stood in various museums, watching Aaron produce a series of original drawings, none of which I had ever seen before, "Who is making the drawings? Who is responsible? Is the program an artist? What part of all this is art?"

The differences between programs and people might be obvious, but they seem difficult to keep in focus. Most people evidently believe that machines must either be programmed to do the PARTICULAR thing they happen to be doing — by this view, Aaron must have been somehow "fed" the drawing it was making — or they

must be behaving randomly. But Aaron always appeared to act rather purposefully, and over and over again I have watched peoples' faces register the confusion which accompanies a successful assault upon deeply-held "beliefs, as it came home to them that this entity was following neither of the only two paradigms they had to hold on to.

"I see," some people would say, "the program is really just a tool!". Well, it is and it isn't. What they meant by a tool was something with a handle at one end and a use at the other: a hammer, a scythe. But suppose one had a hammer that was capable of going around a building site, searching out and thumping any nail that protruded more than a thirtysecond of an inch above the surface? Would we still call that a tool? If one were to write a computer program which allows a composer to sit down at a keyboard and compose music in an essentially orthodox fashion, albeit with an infinitely extensible orchestra, one might reasonably think of THAT as a tool in an orthodox sense, because making a BIG difference is not the same as making a FUNDAMENTAL difference. But what of a program that knows the rules of composition, and generates, without input from a keyboard, an endless string of original musical compositions? Would that be an orthodox tool?

Aaron was clearly not a tool in an orthodox sense. It was closer to being a sort of assistant, if the need for an human analogue persists, but not an assistant which could learn what I wanted done by looking at what I did myself, the way any of Rubens' assistants could see perfectly well for themselves what a Rubens painting was supposed to look like. This was not an assistant which could perform any better for having done a thousand drawings, not an assistant which could bring anything approximating to a human cognitive system to bear on the production of drawings intended for human use.

A computer program is not a human being. But it IS the case, presumably, that any entity capable of adapting its performance to circumstances which were unpredictable when its performance began exhibits intelligence: whether that entity is human or not. We are living on the crest of a cultural shock-wave of unprecedented proportions, which thrusts a new kind of entity into our world: something less than human, perhaps, but potentially capable of many of the higher intellectual functions — it is too early still to guess HOW many — we have supposed to be uniquely human. We are in the process of coming to terms with the fact that "intelligence" no longer means, uniquely, "human intelligence."

Not all computer programs are intelligent, needless to say. Programs are written by people, and if they are written as tools of an essentially orthodox kind, they surely won't be intelligent. For those which are, the question to be asked is not, what ARE they? — Was Aaron an artist? for example — but,

what will they DO? The word "artist" implies human-ness, for obvious reasons. We might as usefully argue about whether Aaron was an artist on the evidence that it didn't wear jeans, didn't drink beer, and didn't want to be famous, as to argue from the fact that it didn't possess a human nervous system and knew nothing about the culture it served. What we do need to know, rather, is the part to be played by Aaron-like programs and successor programs which will be to Aaron what chess is to tic-tac-toe, in the cultural enterprise of art-making.

And that isn't the kind of question to which one can venture an answer with any great confidence today: much less so if it is extended to intelligent programs as a whole. It is certainly the case that some problems in computing have proved to be appallingly intractable: the understanding of natural speech in an unlimited domain of discourse, for example. On the other hand, the limitations I have described in Aaron are not inherent in intelligent programs as such. They merely result from the attitudes and interests I brought to bear on the writing of the program: it could as easily have developed differently, as Aaron's successor has. And Aaron was not abandoned because of its limitations with respect to what it was designed to do, but because it lacked the flexibility to allow it to be adapted to new purposes, that's normal for programs developed in an ad-hoc manner, as Aaron was. By the time I had been patched Aaron up with string and masking tape for five years, by the time I had completely rewritten it three times, it was obvious that that, on the one hand, a program would need to be able to exercise more originality than Aaron had to satisfy me in the future, and that, on the other hand, Aaron's current structure would prevent it ever achieving any such thing.

Stated baldly, though not at all clearly, the new Aaron — I will call it Aaron2 for the sake of clarity — was intended to be creative. Once I had actually managed to state that intention to myself, it took some time to accommodate to the fact that, while I thought I knew creativity when I saw it, I evidently had no very useful notion of what the word "creative" could possibly mean. I was sure that some individuals use their intellectual resources with abnormal efficiency: I was equally convinced that such aberrations ARE a question of the USE of resources, and not one of abnormal resources. And when I came to scan the age-old literature on the subject of creativity, it seemed to me that I was not alone in lacking any functional model, any idea of what happens internally when someone has an original idea. Since I never really believed that the Muse of the ancient Greeks sits on the artist's shoulder and whispers in his or her ear, I had no reason to suppose that there would be a Muse of the Electronic Age waiting to sit on the shoulder of a suitably endowed, or

suitably seductive, program. My program would not "be written as an incantation for the seduction of electronic muses, then, obviously: but what would it be? What can it mean to talk about a program "being creative?" How would one know that it had been?

Let me take a few minutes to make a number of general observations, by way of explaining what I thought about all this, and why eventually the new program was designed the way it was.

In the first place, nothing I have said about the appearance in our world of non-human intelligence was meant to deny that, for most matters involving the exercise of the higher intellectual functions, human intelligence is the only prototype we have. It might not always be that way, but for anyone designing intelligent programs today, I do not see how the modeling of the human intelligence CAN be avoided, or, indeed, WHY it should be.

This must be the case particularly for a program whose output is intended to correspond, on an intimate level of detail, to something as intimately human as a human freehand drawing. I believe one captures the essence of the human performance by modeling the performance itself, and never by attempting to duplicate the appearance of the OUTCOME of the performance. Thus I seemed to be on a head-on collision course with the need to say, in functional terms, what constitutes creativity, and there seemed to be no way around it.

(I should make clear, by the way, that this view is not intended to refer to the implementation levels of programs built around devices which are fundamentally unlike what the human being uses. The video camera being used in computer vision systems, for example, has very little in common with the human visual system, and, to the degree that much of what goes on in vision programs has to do with inferring the state of the external world from the incoming data, there would seem to be no compelling reason to use human visual data processing as a model.)

Secondly, apropos of drawing: like its predecessor, Aaron2 would be making drawings, but not the same KIND of drawings. I need to say something about the differences, and about drawing in general: any classification is to some degree arbitrary, and I should make clear what my own is.

The most inclusive way of regarding a drawing, probably, is as a set of ordered marks, or perhaps we should say INTENTIONALLY ordered marks, since there are all sorts of ordered marks in the world we don't regard as drawings: for example, the tracks of cars in the snow, the veins in a leaf, the cracks in a mud flat... or, for that matter, a musical score or a printed page of text. The question of intentionality is of paramount importance, notwithstanding the fact that intention has to be inferred from forms rather than perceived directly, as forms are perceived. So, for example, we might readily agree that an alphabetic character

drawn by a typographer in "the process of designing a type face is a drawing, while denying that the same character appearing in a printed text is a drawing. The crucial factor is that the designer's character stands for what later appears on the printed page, while the character on the printed page doesn't stand for anything. It IS what it is.

This implies that a drawing is a drawing, not merely because it stands for something other than itself, but because we find in it evidence that the reference to that other something results from an intentional act. Which is not to say that all drawing is representational, in the sense that it makes reference to the outside world in terms of the world's appearances. I suspect that very little of it has been: in fact, it may be that in the whole of man's history, only Western European art from the Renaissance on has ever busied itself with appearances to the degree that it has. It IS a question of degree, of course. A drawing is a set of assertions about the nature of the world, and the form in which those assertions are made derive from the operation of the visual cognitive apparatus, whether or not the marks are intended to refer to appearances. As an example: all human beings at all times have represented the solid objects of the world, on flat surfaces, as closed forms. But at the same time, closed forms, and the distinction between closed forms and open forms, has functioned as fundamental raw material from which all images are built.

It would seem, then, that the making of drawings would be inextricably linked to the possession of a cognitive apparatus, and of cognitive skills. And for a human being it certainly is. But I have been careful to say that a drawing contains the IMPLICATION of intention, as I have also said that the viewer actually assigns his or her own intentions to the artist rather than the other way about. For a program, what is required is enough knowledge about the way images are made and used to be able to generate the IMPLICATION of intention: which is what Aaron did. Aaron did not make representations, in the sense of dealing with appearances. It made images, evocative drawings: which is to say, drawings which facilitated the assignment of the viewer's intentions and meanings. Its successor, however, was designed to make representations.

Now, in asserting that the structure of representations takes its character from the nature of the visual cognitive system, I do not intend to imply that a representation is, in any useful sense, a transformation of the external world onto a sheet of paper. I am quite sure that it is not. What I said was that a representation is a set of assertions about the external world, made in terms of the world's apprehend ability. That does not imply the existence of any one-to-one mapping of the world onto the representation, such as one finds in a photograph, and, its ubiquity notwithstanding, photography is quite uncharacteristic of representation-building in general.

There is nothing particularly original in this non-transformational view of representation-building: every sophisticated artist knows perfectly well that a drawing is an invention, built from whatever raw material he or she can muster, and aimed at plausibility rather than truth. In fact, the idea of truthfulness, realism, is itself just such an invention, one which simply uses the appearance of the world as a hook upon which to hang its claims to plausibility. But if we take this view at face value, disentangle it from the photographic, transformational bias of our time, some interesting questions emerge. In some superficial sense a representation represents the external world, but then it isn't clear HOW it represents that world, or what aspect of the world is being represented. In another sense a representation represents one's internal world — that is to say, one's beliefs about what the external world is like — and it is produced, externalized, in order to check the plausibility of one's beliefs against the primary data collected by one's cognitive apparatus. Obviously, this view of representations as externalizations of an internal world is not limited to drawings, but to any forms by means of which the individual is able to examine his or her own internal state.

And at that point I thought I had my first real hold on the question of creativity, which I was determined to characterize in terms of normal functions, and without falling back upon some superman theory. If this checking process in the normal mind is put to the service of confirmation, of reassuring the individual that the world is the way he or she believes it to be, we might suppose that its function in the creative mind is to DISconfirm, to test the individual's internal model to the limit, and to force the generation of new models. In other words, the essence of creativity would lie in self-modification, and its measure would be the degree to which the individual is capable of continuously reformulating his or her internal world models: not randomly, obviously, but in perceptive response to the testing of current models.

Thirdly: to talk of one's internal model of the world is to talk of a representation, clearly. But it is not a fixed, coherent representation, the way a representation on a sheet of paper may be thought of as fixed and coherent. It takes very little introspection to discover that the pictures we conjure up in our heads are anything but complete. Try conjuring a picture of your favorite person's face, and then ask yourself a question about it — what is the distance between the eyes, for example — to see how volatile the mental image is, and how little information is carried in it. Ask a question about something quite different, and a quite different mental image may spontaneously emerge to replace the image of the face. Evidently, there is some store of material below the level of these mental images, and we should probably regard these images as a sort of semi-externalized representation of the material at the lower levels. What we mean when we speak of one's internal model of the world is not a

mental image, or a succession of mental images, but rather the entire internal procedure "by means of which those images are continuously produced. Representations represent lower-order representations, and exist as a series of momentary cross-sections in a continuous unfolding, a continuous reconstruction of the world from the debris of experience.

We ought to be able to characterize creativity in terms of this normal representation-building: that is to say, we should expect to find creativeness exercised, not as another kind of function entirely, but in highly particularized modes for the reconstruction of mental models from low level experiential material. It is not surprising, then, to find Albert Einstein, one of the few to have written about the nature of creativeness from within and in a convincing way, speaking of the part played by this lower-order material in thinking:

"It is by no means necessary that a concept must be connected with a sensorily cognizable and reproducible sign (word: in our context, mental image)... All our thinking is of this nature of a free play with concepts... For me it is not dubious that our thinking goes on for the most part without use of signs, and beyond that to a considerable degree unconsciously. "

We might conclude that in Einstein's case, creativity involved an extension of the domain of "thinkability," manipulability, to a level on which most of us find mental material to be unmanipulable.

Fourth: a very large part of what the individual has in his or her head is knowledge about how to do things. And people don't behave creatively unless they know how to do a great many things, just as they don't behave creatively unless they are capable of abstraction. There is nothing creative about ignorance. How, then, could one expect a program to exhibit creativeness, self-modification, unless it, too, first knew how to do a rather large number of things, whether it had acquired that knowledge experientially, or had it provided, hand-crafted, by the programmer. The ability to acquire experience would need to be built into the program at the outset, but the self-modification which might proceed from that experience, would probably come at a late stage in the program's development. That implies, of course, that the program would need to be able to store, in some appropriate form, everything it had ever done.

Which leads to the fifth observation, and to what is perhaps the most teasing of all problems relating to the mind. The mind evidently stores all its knowledge, all the experience of its owner's life, in some amazingly compact fashion. What happens to your knowledge about how to cross the road when you are not crossing the road? When you sit down to play a game of chess, do you find all you know about the game stored in one big lump, like

a book on a shelf, somewhere in your memory? Can you access it all at once, form a single mental image of it? Presumably not. When you need to find an appropriate rule for crossing the road, do you need to review and examine all the rules you have for playing chess, and for eating spaghetti, and for tying your shoelaces, on the way, in order to determine whether any of them are appropriate to the current situation? Presumably not.

What we mean by a rule is not an imperative — WATCH OUT, EAT YOUR FOOD — it is a conditional. — if you can't beat 'em, join 'em: if the cap fits, wear it: if they can't get bread, let them eat cake — and the condition which triggers the required action seems to lead us directly to what the action is. Rules for the tying of shoelaces appear to live with the shoelaces, and rules for eating spaghetti live with the spaghetti. Or, to put the matter another way, rules for the use of things are simply part of our conceptual grasp, our internal representations, OF those things.

Of course, most rules in real life are a good deal more complex than these examples, if only for the reason that things in the world interact with each other. Rules link events: if 'a' is the case, and either 'b' is or 'c' is provided that 'd' isn't... and so on. Also, many rules belong to classes of things, classes of behavior, rather than to individual things and individual behaviors. The rule which says "If you are eating spaghetti AND wearing a new jacket, proceed with caution" is a rule belonging to a whole class of messy foods which stain clothes, and is invoked by the appearance on the table of a dish of spaghetti, by a process we might call inheritance, by virtue of the fact that membership of the class "messy foods" is part of what we understand by spaghetti.

You will recognize that these remarks are directed at WHAT the mind does, and make no assumptions about HOW it performs its feats of information processing. On that question I know nothing, nor do I believe it is central. My aim was to identify, in a few essential characteristics of human intellectual activity, the informing principles of a program, not to replicate the processes through which the mind runs its own programs. Let me summarize those principles.

Firstly: Aaron2, unlike Aaron1, should have a permanent memory. In this memory should be stored, in extremely compacted form, every drawing the program makes, together with everything that the program knows about drawing, whether that knowledge is programmed by hand or acquired through experience of drawing. But, compacted though it should be, that stored material should be structured so as to inform its own regeneration into more complete specifications for the making of a new drawing. I mean this as an analog for the building of representations in the mind

from lower-order representations, up to and including the generation of external representations. However, this process in the program should be flexible enough to reflect the associative quality of the process in the mind. (I have neglected to mention association up to this point, largely through lack of time: nevertheless, my suspicion is that creativeness is not a function of "correctness" in representation-building so much as it is a function of the slightly messy, apparently somewhat random, action of association.)

Secondly: the knowledge the program should have, its domain of expertise, should concern, predominantly, the making of "visual" representations: that is, it should know enough about the nature of the visual field, and about the way people derive knowledge of the three-dimensional world from it, that it would be able to generate a convincing sense of depth, regardless of the lack of any data concerning the objects in the visible world.

This principle was actually quite arbitrary with respect to the program's planned structure, though it made sense to pick a domain in which I felt I had a good deal of expert knowledge readily available, and it was certainly justified as an excellent example of the final stage of the externalizing process. But you will have recognized that almost none of my remarks have been directed specifically to drawing, and I tend to think the program could as easily deal with other material.

Thirdly: the rules which determine how its knowledge of drawing is to be applied in the making of particular drawings should be accessed by the program as it accesses the knowledge itself. Perhaps I should have explained that Aaron1 was what we call a production system: simply a long list of rules — if some condition holds true, do this, otherwise if something else is the case, do that, otherwise ... — in which the program simply cycles through the list until it finds, and activates, an appropriate rule. One of the conceptual problems of this kind of program is that the knowledge of how to do things is split up, between the rules on the one hand and the subroutines invoked BY the rules — the "do this, do that" part — on the other. Thus, Aaron2 should provide a more coherent representation of "how to do it" knowledge than its predecessor.

Fourthly: the program's knowledge of drawing should include conceptual knowledge, at least to the degree that it should be able to particularize from general rules. I mean, for example, that it should not only know that there is a general class of things called closed forms, but should know about all the members of the class and be able to decide that one was more appropriate in a particular situation than another. Conversely, it should also be able to remember that it had used a closed form for some reason without necessarily having to remember which closed form it was.

Fifthly: Aaron2 should be -treated as a potentially "late developer." I mean that it should be anticipated that the programmer would need to put in rather a lot of material by hand before the program would be ready to take off. But if the program is ever to take off, it should then be able to make use of the same mode of entering material as the programmer had, or at least be able to generate material in the same form.

And so things are working out. Aaron2 is still in its infancy and a very long way from becoming self-modifying. In order to support the long range need for building up the program's store of knowledge, early work on the program involved the writing of an editor, by means of which the programmer is able to build items of knowledge by hand. These items are, indeed, extremely compact: in memory they consist simply of sets of tokens, unique names. Once an item is accessed by the program, however, it is regenerated into a generalized tree structure, and the individual tokens are enacted.

Perhaps this is a little abstract: what it means in practice is that the programmer, having written a set of subroutines that describe how a particular kind of closed form may be generated — let's call it a "shape," for example — uses the editor to implant in the program's memory the fact that it now knows how to generate these "shapes." At this point the memory item will consist of the single token "shape," together with a marker which identifies the token as the smallest unit of "how to do it" knowledge, which we will call a "system". Any time this item is accessed, the marker will cause the program to activate the generative subroutines to which the token refers, and a "shape" will be produced.

Suppose now that the programmer writes another set of subroutines for adding a kind of appendage to a closed form — we'll call it a "base" this time — and uses the editor in the same way to implant another item of memory. Now, because of the way they are generated, "bases" can only be appended to closed forms, and it follows that in due course the programmer will want to add a rule to this memory item which will prevent it from being activated for any other purposes. For the moment, however, the programmer uses the editor to build another memory item, this one carrying a marker identifying it as a figure — not simply a system — which has, as they say in computer-talk, two children, each of which is a system. The first system is the token "shape," while its sibling is the token "base," and in implanting this more complex item in memory, the editor will create a token by which the item will henceforth be known: it is civilized enough to make it pronounceable, if not sensible. This single token is now all that is required to generate a figure consisting of a shape and a base: or a shape with a whole series of bases, actually, since

repeat factors may be assigned to any token in a memory item.

It is not difficult to see how the editor may be used to create groups of figures, each of which will have systems as children, and pictures, which will have groups as its children, each of which will have figures as its children, each of which will have systems as its children, each of which may have other systems as its children, and so on. Thus, by the time the programmer has been working for a short time, the program will have in its memory, not merely a number of items, but items of different levels of complexity.

If we look at the items in detail, moreover, it will be seen that they do not simply exist in isolation. Each item may have within it what we will call a HASA list, which will define the sets of which this item is a member, an ISA list, which defines the item's properties, and an "ASSOCIATION" list, in addition to its RULE list. If the programmer, in creating the system "shape," had declared that a "shape" ISA closed-form, then the editor would automatically have created a new "closed-form" item with a "concept" marker — assuming that one hadn't existed already — and would have entered "shape" in its HASA list. Similarly, the programmer may have created a concept item by hand. In either case the assertion of an ISA association will cause the automatic generation of a HASA association in the appropriate item. This facility is completely general, so that eventually the program may know that one system is an example of a curvilinear closed-form while another is an example of a rectilinear-closed form, both of these sets being members of the superset "closed-forms," while this, in its turn, may be a member of the set "forms-useable-for-the- depiction-of-solid-objects." This is what will allow the program both to generalize and to particularize, and to substitute one member of a set for another. It is also this mechanism which will permit what I referred to earlier as inheritance: the application of a rule belonging to a class to any member of that class.

The ASSOCIATION list functions as a linking mechanism of a much more general kind, and is intended to allow the modeling of just what the name implies: those connections of items in human memory which may be extremely strong, though without necessarily having any very obvious reasons for existing.

As I have said, Aaron2 is now in its infancy. It has in its memory no more than about twenty items, three or four of which represent complete pictures: or, more precisely, classes of pictures, since the same item could be enacted a thousand times without ever producing the same drawing twice. Most of the things it knows how to make are readily discernable in its drawings, and once you know what you are looking for it is obvious how few things it knows how to do: far too few to move to the next major step.

That step will involve providing Aaron2 with a number of criteria, which it will be able to apply to its own performance. Suffice it, for the moment, to say that these criteria will reflect what I think of as cognitive constants, and that the program will judge the enactment of any item of memory by how closely it has matched one or another of these constants: or, to put it more simply, how "like" the visual field the current drawing is. Having generated a closed form, for example, it may judge that its outline is quite short in relation to its area, implying that the form is not yet complex enough to "match" the structure of the visual field. In that case it will be able to make use of any of the links it has to traverse memory in search of something it knows how to do which will add to the complexity of the figure and better satisfy this particular criterion of complexity.

If it succeeds in doing so, it will have learned how to do something it hadn't known how to do previously, and, using the same editor that built its memory in earlier days, it will commit to memory this new piece of knowledge. You will see why I insisted that a program like this would need to know a great deal before it is ready to be let loose. Once it is let loose, my guess is that it will develop quite rapidly, and I am prepared to believe that in a short time its drawings will be unpredictable, not in the simple sense that Aaron1 's drawings were unpredictable, but in the more profound sense that they were produced by a program which had changed since it was written.

What will its drawings be like? Obviously, I can't know in detail, though I think I would be quite surprised if Aaron2 generated a Leonardo. Will they be wonderful? Will they become so unlike the externalizations of the human mind that they cease to function as those cultural artifacts we call works of art? Who can tell. But I am preparing now to devote some years to finding out.