Driving the Creative Machine.

Orcas Center, Crossroads Lecture Series
September 2010.

A few weeks ago I found myself in my habitual pre-writing condition, wondering, that is, how long I could put off making a start on this talk, when – a gift from heaven! -- there was Thomas Friedman, in his Op-Ed column in the New York Times, writing about creativity – more precisely, about a Newsweek article about creativity. New York Times? Newsweek? Wow, I thought, is everyone talking about creativity these days?

"To be creative requires divergent thinking (generating many unique ideas) and then convergent thinking (combining those ideas into the best result)."

So says Newsweek. But if that's all there is to it, how come we don't see creative behavior all around us? Ah, well, of course there's the divergent thinking bit. Where does divergent thinking come from? Friedman is clear about that, though:

"It comes from being exposed to divergent ideas and cultures and people and intellectual disciplines. As Marc Tucker, the president of the National Center on Education and the Economy, once put it to me:

"One thing we know about creativity is that it typically occurs when people who have mastered two or more quite different fields use the framework in one to think afresh about the other. Intuitively, you know this is true. Leonardo da Vinci was a great artist, scientist and inventor, and each specialty nourished the other. He was a great lateral thinker. But if you spend your whole life in one silo, you will never have either the knowledge or mental agility to do the synthesis, connect the dots, which is usually where the next great breakthrough is found." "

Sounds very authoritative, and you might suppose that I'd agree, given that I've spent half my life trying to get a computer program to do what only rather talented human beings can do. But, in fact, I don't know, intuitively or otherwise, that what he's saying is true.

Citing Leonardo to make a point is a bit like yelling a slogan at a political rally; you can't argue with Leonardo. Actually, Leonardo was a single example of the Renaissance ideal of the Universal Man – the educated individual who could turn his head and his hand to anything. So were Giotto and Piero dell Francesca and Raphael and Donatello and Brunelleschi and Michelangelo and Giorgioni and whoever else you care to name. That isn't what Tucker's addressing, though; he's talking about the 20th century Specialist; specifically, the atypical one who crosses the boundaries of his own specialty. Two specialisms do not a broad education make, and even with their broad education you'd have a hard time finding evidence that Piero's design of

public events nourished his painting; or that Giotto's painting influenced his design of the campanile in Florence; or that Leonardo's expertise in painting led to his invention of a non-functional helicopter. Where, exactly, was the next great breakthrough in these cases? Did Leonardo invent a helicopter that could paint landscapes?

As it happens, most of the artists I've admired and most of those I've known personally, have lived and worked precisely in their own single silos; few of them achieved mastery any place else. (That includes me, by the way; I could never claim to have achieved mastery in computing to match that of any average computer science masters student.) Cezanne makes it clear that his painting is about painting; he wanted to “do Poussin over again from nature.” The Impressionists didn't need to be masters of chemistry to take advantage of the new colors presented them by the industrial revolution; Picasso didn't need a PhD. in African art history to recognise the significance of African sculpture to his own painting; he merely needed to see it.

My biggest problem with Tucker's account, though, is not that he seems to be extrapolating from – yet still not identifyng -- an extremely small high-tech sample, but that he's talking about the *mechanics* of creativity, without giving so much as a nod to what I suspect to be the most important issue of all; namely, the engine that *drives* creativity, no matter what mechanisms are employed in its exercise.

I happen to believe, for example, that typically creativity arises when the individual starts to question the unquestioned assumptions of his field and to act out the scenarios that present themselves as a result. What if an image didn't need to reflect the appearance of the world from a single viewpoint? (Cubism?) What if we didn't have to use only the colors we find in nature? (Matisse?) What if one could discover the rules we use in making art and then have a machine act out the rules? (Cohen?)

What if? That essential question is, I'm sure, a central mechanism of creativity; but what sets the enquiry in motion in the first place? What drives the various mechanisms? Some intellectual disquiet about the current state of the field? Possibly; but why, then, would one individual feel disquiet about the state of his field when all the other experts in the field are perfectly comfortable?

These 'whys' aren't the sort of thing creative people usually talk about; for one thing, creative people don't necessarily know about their own personal 'whys'. But today I want to talk about the whys, because my topic is not just creativity but machine creativity, and only when I know why something has happened in one place can I say with any confidence that it may or may not happen someplace else.

Machines have been involved in the exercize of my own creativity for slightly more than half my life, and now I'm trying to say whether those machines could be, or have been, equally creative. But why the question? isn't it clear that my program,

AARON, is creative? With no further input from me, it can generate unlimited numbers of images, it's a much better colorist than I ever was myself, and it typically does it all while I'm tucked up in bed.

What, in fact, is my own claim to creativity when I've literally had no hand in actually making the work?

Well, of course, I wrote the program. It isn't quite right to say that the program simply follows the rules I gave it. The program IS the rules, and regardless of where they came from, it is those rules – that is, the program – that generates material I could never have imagined or generated myself.

So it would seem at first glance that I get credit for writing the program and AARON gets credit for doing the work. I'm going to offer a rather different formulation, however, and I want to put credit assignment on hold for the moment and talk about how – and why -- the program came to be written in the first place and how and why it has developed more or less continuously over the past forty years.

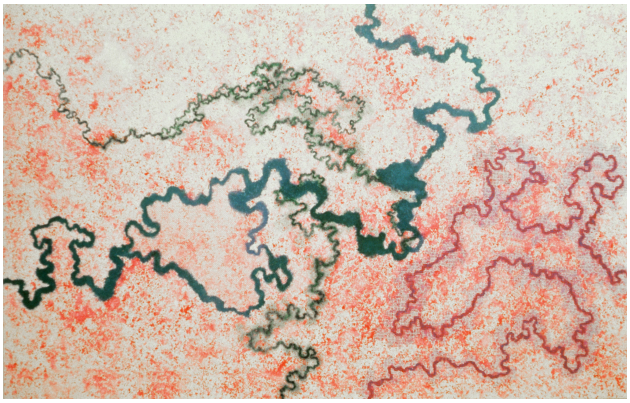
I propose to examine some key events along the way – what Tucker would call “breakthrough” events -- with a view to clarifying what part was played by me and what part by the program.

The first of them, clearly, has to be my initiation into the mysteries of computing in 1968, when I came from London to the newly established Visual Arts department at UC San Diego. Just days after I arrived, I was introduced to a graduate student in the Music department who offered to teach me programming.

It was a random event, in the sense that I didn't know the student, I didn't initiate the conversation, and I wasn't looking for anything. He simply had an itch about teaching programming to anyone who would listen, as long as they weren't computer scientists. And, in retrospect, my decision to take him up on the offer might seem more than a little bizarre. Recall that in 1968 computing meant IBM cards chugging slowly through computers literally millions of times less powerful than today's laptops, and it could cost you hours, sometimes days, to determine that there was a semi-colon missing on one of your cards. Why would any sane individual put aside a flourishing career – because I'd been at the center of the London art world for most of the decade – in order to play with something as unpromising as a gigantic toy locked away in an air-conditioned hall behind some operator's desk?

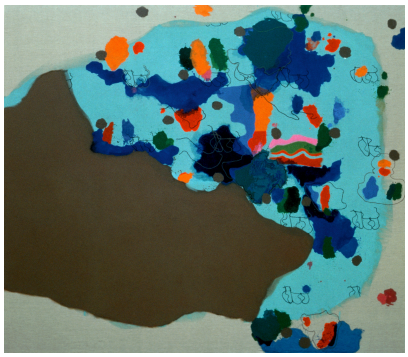
The engine driving the decision was not random, however, though at the time it never occurred to me that there was more to it than the intellectual excitement offered by this great new game called programming. Looking back, however, the fact is that all of the paintings with which I'd represented the UK in the Venice Biennale in 1966 (figure 1) had their origins in the notion that making art didn't have to require ongoing, minute-by-minute decision making – the unquestioned,

“normal” mode – and that it should be possible to devise a set of rules and then, almost without thinking, make the painting by following the rules.



(figure 1: “Vigil Completed” 1966. collection: Tate Gallery)

I had no idea in 1966 that two years later I would be presented with a language in which those rules could be expressed, clearly and unambiguously; much less that it would be backed up by a machine that could actually execute the rules.



(figure 2: “Pastoral” 1965) (figure 2:

(figure 3: “Before the Event” 1965)

So I was set up for my move to computing by an existing preoccupation with the idea of a rule-based way of making art. But we have to go a bit further back to see *why* I was driven to abandon the normal “continuous decision-making” paradigm of art-making in favor of this new rule-based paradigm.

My paintings in the early 'sixties (figures 2, 3) rested upon my ability to invent material, which would then be presented as if it were material in the real, physical world. The very phrase -- as if -- expresses something fundamental to any image-making, and the work went well. Yet I found myself gripped by an increasing uneasiness about the invention part of the process. Could one actually go on inventing for another fifty years?

Well, yes, of course one could, and I have. My move to computing didn't make invention redundant, though that was its aim, it simply shifted it to a higher level. But my uneasiness was no less compelling for being ill-founded, and it was enough to change my very sense of what art was, or could be, and set me off on the path I've been following ever since.

In summary, then; this first “creative breakthrough” did involve some cross-fertilisation of technologies, though heaven knows I was no expert in computing; and it did involve a questioning of unquestioned assumptions. But it also involved a root cause, buried deep within my own psyche – much deeper, of course, than this superficial account suggests. And that root cause – the engine that drives the process – would seem to indicate that, whatever the creative act does for science or for art, it is accomplished primarily to serve one's own needs.

Where, then, does that leave the idea of machine creativity? The day may or may not come when machines have a sense of self. If it doesn't, it means that machines will never be creative in the same sense that humans are creative. But even if that day never comes, it doesn't mean that machines have no part to play with respect to creativity. Had I not met my first computer in 1968, had I not found this particular way of expressing my own needs with respect to art-making, my life and work surely would have followed a very different vector for the past forty years.

My second case history begins some fifteen years after the first, and it comes in two parts, both involving color. As a painter, color had always been central to my work, but of course there were no color devices in the early seventies, AARON began while I was a guest scholar for a couple of years at the AI Lab at Stanford, and for it's first decade or so it was limited to drawing. I colored some of those drawings by hand (figure 4), some were projected onto canvas and became paintings; some of them even became tapestries and murals (figure 5, 6). But I could never shake off a growing sense of absurdity; if the program was so damned smart, why couldn't it color its own drawings? Was it working for me or was I working for it?

(figure 4: hand colored machine drawing, 1983)

(figure 5: Two murals for Digital Equipment Corporation, 1986)

(figure 6: San Francisco Museum of Modern Art, 1969.)

Yet for all my experience at Stanford, the birthplace of expert sytems, I was unable to see how I might capture my own expertise as a colorist into a program. And when the breakthrough came, finally, in the mid 'eighties, it wasn't specifically about color. IBM's Big Blue had just beaten Kasparov, one of the greatest chess players in history. It beat him at his own game, I thought, but surely it wasn't playing it the way Kasparov played it. How could a program choose which half-dozen moves to consider without assessing all the other enormous number of legal possibilities, the way a grand master does?

The critical insight was that we do what we do with the resources we happen to have, and we're both enabled and limited by those resources. That has to be true both for man and machine, and while I had no idea what resources would be called into play in chess, I began to see that man and machine have very different resources to bring to bear on the use of color.

On the enabling side, for example, human colorists have a highly developed color vision system. On the limiting side, though, we have virtually no color imagination – I mean that we can't build an internal representation of a complex color scheme. Actually, we don't even have an adequate vocabulary for describing simple color relationships, much less complex color schemes. The combination of the two – enabling and limiting -- has led to the near-universal mode for human colorists; a feedback-controlled mode in which images get colored bit by bit and with constant adjustment of what's already there, until the colorist somehow knows he has it right. Not even the greatest colorists could write down the instructions for coloring a complex image and give them to someone else to execute.

The computer, on the other hand – mine, at least – had no visual system at all. But on the enabling side, it's able to build and maintain schemes of many kinds, at any level of complexity. Obviously the machine couldn't do coloring the way I did it myself. But I saw that it could do it, if only I could develop an adequate formalism in which to describe color relationships within something as complex as an image.

That was the first indication I had that programming might involve trying to think in the computer's terms, as it were, rather than trying to get it to think in human terms. That, truly, was where the real breakthrough was for me. But that was only the first part of the story of AARON as colorist, and the subsequent part is a whole lot more interesting.

(figure 7: screen shot, 1992)

(figure 8: Oill on canvas, 1992)

(figure 9: Painting Machine, 3rd version
collection: Museum of Computer History

I was quickly able to get AARON coloring; there were color monitors by the mid-eighties, so I was able to see output on the screen, but there were no physical output devices. For some time I photographed the screen, then enlarged the images up onto canvas and painted them by hand (figure 7, 8).

History and bought myself one of the wide format printers that were just then coming onto the market. I've been using it and its more recent successor ever since

(figure 10: Painting Machine output, 1995)

(figure 10: Painting Machine output, 1995)

(figure 10: Painting Machine output, 1995)

(figure 11: Wide Format Printer in studio)

(figure 11: Wide Format Printer in studio)

(figure 11: Wide Format Printer in studio)

But I thought I shouldn't need to be in the path from program to output, and I spent altogether too much time building a series of painting machines (figure 9) so that the program could handle the physical side of image-making for itself. (slide 10) In

practice, my “direct” path from program to output proved to be a detour lasting several years, and finally I gave the last version to the Museum of Computing

Even after the move to wide-format printing, however, the software supporting all this work had remained in the orthodox rule-based form to which I'd become accustomed – that is, the coloring part of the program was a long list of situations that could arise as an image developed, each invoking what to do about it: if such-and-such is the case, do the following. And if not this such-and-such, go on down the list until you find which such-and-such is the case, and do what it says to do.

The problem was that such-and-such's are very hard to determine in relation to something as fluid as color relationships, particularly when the imagery to which they apply is itself changing over time. The list was getting longer and longer and increasingly difficult to manage; a change in one place could easily screw up things someplace else, and frequently did. Finally, more from exasperation than from any clear plan, I decided on a complete overhaul, and set about rewriting the entire system.

Before I got very far, however, I awoke one morning with a small voice in my head saying “why don't you try this?” What it was proposing was so manifestly absurd that it should have been easily discounted as the remnants of a silly dream. But the voice persisted. And since it would clearly take very little effort to write the code to try it, I tried it. And half an hour later I watched in astonishment as AARON generated image after image every bit as good as, and a good deal more varied than, anything my rule-based system had ever done (figure 12, 13).

The details of this shockingly simple program are not really the issue here – and in any case they've been described in several of my papers available on the web. What's to the point is that the program had, in a single step, become an expert colorist in its own right. Obviously I was the source of the knowledge behind AARON's new expertise – it had no other source -- but the form of that expertise in the program was quite alien to me. I couldn't see why it worked as well as it did, and in the following years I found I was unable even to describe it without going back and reviewing the code I'd written.

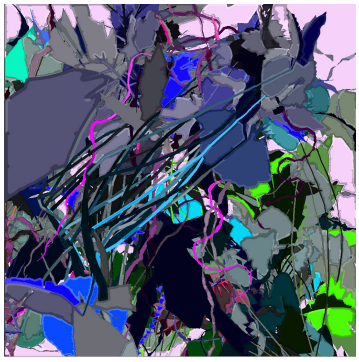
But most significantly, I could never have used the program's strategy to do coloring myself. At the risk of being unduly fanciful, it was almost as if, after years of requiring the program to do things my way – if such-and-such, do this -- the program had finally said, just tell me what you want done and I'll do it my way.

Let me see if I can summarise the circumstances surrounding what was certainly the biggest single breakthrough in twenty years of working on the program.

First, the story began with my mounting dissatisfaction about having to do the program's coloring for it. That was the engine driving the process. As to the

mechanisms, technology had its part to play; through the availability of color monitors, obviously, and – much more importantly – through a change of programming language from C to Lisp.

(figure 12: Untitled, September 2006)



(figure 13: “Rincon #8” October 2006)

The major element, however, rested upon an insight of a general nature which then raised a question about the nature of expertise -- color expertise particularly. Was it true that one could weazle knowledge out of an expert's head and install it in a computer program? There were dazzling examples – I'd been in direct contact with some of them when I was at Stanford -- but I began to suspect that success depended in some degree upon the distance between the expert's modalities and those adopted by the computer programmer; the so-called knowledge engineer. Mixing colors is very unlike writing computer code.

As to the second part of the story; it began with me recognising myself – and the program – to be in a dead-end situation and thus being forced to step back and to reconsider the entire enterprise. Why the answer came in the bizarre way that it did I cannot say, and the experience has never been repeated.

But there was one other feature of the circumstances that had been hiding in plain sight for a very long time, and I was beginning to suspect that it was, in fact, the most pertinent of features. It was that by this time I'd been working on and with the program pretty much every day for more than twenty years, making small shifts in response to the program's output; large changes like this one in response to an over-arching review of where we were at, AARON and I.

And as the significance of this long term relationship came into focus for me, I began to reconsider the whole issue of credit assignment for creativity in a new way. It was wrong, I began to suspect, to divvy up the credit for creativity, giving some to me and some to the program. Creativity – this particular example of creativity -- lay in neither the programmer alone nor in the program alone, but in the dialog between program and programmer; a dialog resting upon the special and peculiarly intimate relationship that had grown up between us over the years.

That view of the issue was reinforced, in an upside down sort of way, by my next “breakthrough” case study. By early last year AARON needed no input from me with respect to color – nor indeed was I able to supply any -- and I'd shifted my attention to drawing and composition, in the broad sense of generating material and deciding where to put it. For several years AARON's material had been essentially, though not explicitly, botanical; I mean that my goal was not to generate identifiable and “correct” images of particular plants, but rather to generate material that could stand-in for those plants in evoking a natural environment. (figure 14, 15.)

(figure 14: Untitled Print, #060852 August 2006)

(figure 15: Untitled Print (diptych), #080123.27 January 2008.)

From a programming point of view, the code emulated the structure of leaves and trees, not their appearances, and each element would be grown from scratch, so to speak, not trotted out as variations of stored prototypes. It never had any prototypes other than the generating code itself, which would produce family-related but never identical forms.

The generating code had to be written, however, and writing it required a good deal of specificity with respect to the individual families. And more recently, AARON's

(figure 16: Untitled Print #080913.20 September 2008)

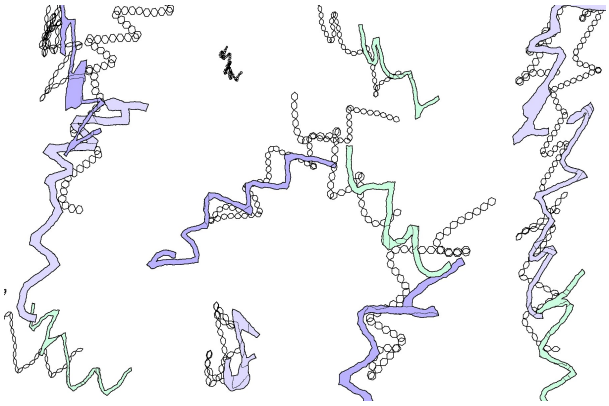
(figure 17: Untitled Print #090924.5, September 2009)

environment had moved from forest and meadow to what I thought of as under-water reef (figure 16, 17). The shift required having to write new code for each new family I might want to add, of course, and I became increasingly frustrated by what I saw as a simple-minded accumulation of independent chunks of computer code. Which of course led to the inevitable question. Why couldn't the program generate its own forms?

Well, it could. In the new version of the program, everything is built from an abundant supply of a single element; a small, hexagonal cell which attaches itself to the corners of other cells. Once a colony of cells has reached a certain size or complexity, AARON draws its enclosing boundary, using very much the same code it used in its very earliest versions forty years ago, and places this newly generated formal element into the developing image (figure 18).

These formal elements come in families, in the sense that a family is defined by the particular local rules it has for how and where a new cell may attach itself to a developing colony. It's easy enough to spot some of them – the zigzagging forms are clearly related, for example – but in general the rules give rise to a wide range of forms, and I ended up with something resembling a continuum rather than discrete, identifiable families (figure 19).

One of the strangest, and most significant, things to emerge from this new way of generating material was that it produced usable results only when I concentrated exclusively on devising local rules for attaching cells, without any regard to the appearance of the outcome. When I fell into the trap of trying to devise rule-sets to produce something I thought I might like to have – an octopus, say, or a fish – the results were uniformly hopeless. The program was in command here; once I had told it how to proceed I had little or no control over the physical appearance of what emerged. I either liked it and used it, or I didn't like it and abandoned it.



(figure 18: cell colonies with (offset) outlined results)

(figure 19: Untitled Print #090921.1, September 2009)

The very success of the program in fact led to the biggest personal crisis for me in many years. I'd spent all those years trying to increase the autonomy of the program; it could already do all its coloring without my intervention, now it could do all its drawing too. The problem wasn't that there was nothing left for me to do – there was still much for me to do; trying to figure out how AARON could assess its own work, for example. But you can't change the subject in the middle of a dialog, and I felt that my dialog with the program, the very root of our creativity, had been abruptly terminated.

How to re-establish that all-important dialog? Well, I did find a way, though I could never in my wildest dreams have imagined a year ago what it would be. And that leads me to what is, in effect, a report from the trenches; not a case history but a case present.

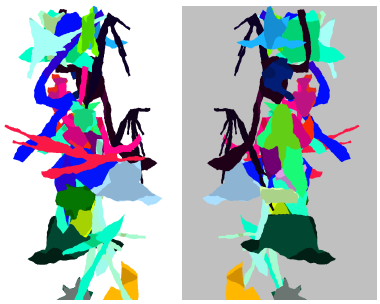
Let me start by describing the circumstances leading up to what I think may be the most important single step for me since I met my first computer forty-two years ago.

In the wake of my excommunication, and with exhibitions coming up, I had little choice but to stop work on the program and to concentrate on physically producing some of AARON's output. For several years now that has meant printing its images on heavy rag paper using archival pigment inks, then permanently bonding the result to lightweight rigid panels and spraying them with a tough protective coating. That way they could be displayed in very much the same way as paintings, without the need for framing behind glass.

Then, early this year, the work was interrupted. Someone put my name forward for a limited competition public art commission. There were to be five such commissions, all of them for some sort of hanging sculpture for the new county administration complex in San Diego. Now, I'd done several public pieces in the past by disassembling AARON's work into separate cutout figures, like this screen wall for the Buhl Science Center in Pittsburgh (figure 20). But I'd never done a fully three dimensional work that could be viewed from all around, and I spent several weeks working on the geometry of what was to have been a fifteen foot high piece. At the end of May the seven remaining contestants submitted our proposals. The five commissions were reduced to three, and, for better or worse, I didn't get one of them.

I was left with a large model, and several small panels I'd printed to show the committee how different variations of the basic plan would look from the two ends of the hall in which it would have been displayed (figure 21). Not much for three months' work. That was the least of my problems, though, because by this time I was so out of touch with AARON that I could barely remember what I'd been doing before. Was there anything to be salvaged here?

(figure 20: Screen Wall,
Buhl Science Center, Pittsburgh, 1984)



(figure 21: demonstration panel)

I quite liked the little panels, but I'd printed them using completely neutral gray backgrounds that I didn't like at all. And one day, for want of anything better to do, I thought I might liven them up a bit by painting over the backgrounds with oil paint (figure 22). I could hardly believe the difference that this simple change to the background made to the reading – the presence – of the image as a whole.

(figure 22: panel with modified background.)

What had I done, exactly?

Things can always sound rational and purposeful after the event, and I want to emphasize that it was well after the event that I began to see that what I'd done on a whim actually made a great deal of sense. Let me see if I can explain.

It sounds too obvious to be worth saying that images have to be made. What isn't quite so obvious is that making has always required the manipulation of physical material, whether that meant scratching through the desert varnish on a rock, or guiding gooey colored stuff around on a canvas with a brush to paint the Mona Lisa. Those materials and those manipulations always leave their identifying character on the resultant images, so that a pencil drawing doesn't look like a pen drawing; an oil painting doesn't look like a fresco painting or a watercolor.

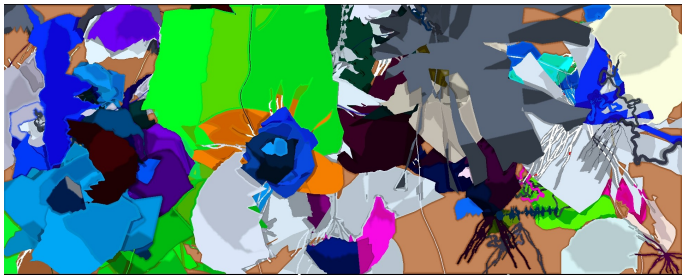
That was always true, at least, until the invention of photography, although even here there was considerable material-based variation until Kodak standardised everything a hundred years ago. The difference is that photographic images don't get *made* in the sense that a drawing is made. The photographer chooses what to point the camera at, and exercises control over the various post-processes, but the image itself is made, with neither thought nor intention, by the light coming from the subject.

Electroni*c* images don't get made in the sense that a drawing is made, either; they just sort of happen in the phosphors on a display screen. They can be printed, of course, as AARON's images have been printed, and the printer can exercise a great deal of control. But the actual printing is a mindless, automatic process in which each pixel in the image is given exactly the same emphasis as every other pixel. No human-generated image in history was ever made that way.

Now, the viewer will discover the intention of an image – what he might think of as its meaning -- only if he believes the image to be intentional. We may see an elephant in a cloud, but we don't think that “elephant” is the meaning of the cloud; that would require us to assume that the weather system, or some other power, was intentionally using the changing cloud formations to communicate elephant-ness.

So, marvellous though it may be to make images without the troublesome material side of image-making, part of the problem with electronic imagery is precisely its untouched-by-hand look; if it wasn't touched by hand, if it shows no evidence of the manipulation of material, then it becomes that much harder to believe in its intentionality. There's the most important reason for Photoshop's long list of fake textures; not so much to pursuade the viewer that he's looking at a drawing or a watercolor, but rather to enhance the image-maker's apparent intentionality.

Interestingly enough, as long as AARON's images were pretty complex there didn't seem to be much of a problem with the untouched-by-hand look of its prints, just as there doesn't seem to be a problem with photographs; the intentionality gets transferred to what the image represents. But in the final months of last year I had been making a conscious effort to simplify the imagery, with the result that the individual elements were getting larger and, consequently, flatter (figure 23).



(figure 23: Untitled, #081027.2, October 2008)

(figure 24: “At First Light” August 2010)

Whether I knew it or not – and I didn't – that seems to have been the reason for painting over the background of one of AARON's little panels. I was opening the door to the assumption of intentionality in the reading of the image. Once I did know it, there was only one way to go, and In the past four months all of AARON's images – not just their backgrounds – have been worked over (figure 24 – 27).

Where that way will lead I have no way of knowing. For the present, I have to regard this as the most radical move I've made since I met my first computer forty-two years ago. It has not merely re-opened my dialog with the program, it has redefined the relationship upon which that dialog has been based.

(fig 25: Untitled. (ref 00018) August 2010)

(figure 26: untitled (ref #00008) August 2010)

In both the previous cases -- color and form – the way forward had rested upon strategies that I could not execute myself, even though I wrote them. Now I was contributing something the program had been unable to do, and in the process relinquishing my exclusive role as rule-giver and becoming collaborator.

Needless to say, I have no reason to suppose that my dialog with AARON has been characteristic of machine-involved creative enterprises generally; if, indeed, they exist. I can't call to mind another example of a program that has continued to evolve over forty years under the control of a single programmer.

(figure 27: untitled (ref 00032) September 2010)

But my goal today hasn't rested on AARON's uniqueness in this regard. It has been to establish what I think is a rather obvious principle; namely, that the best-designed set of mechanisms won't accomplish much without an engine to drive them, and that the engine is provided for a creative human agent by a lifetime of experience of the world and of self in that world. It is rooted in needs of a highly individual and often idiosyncratic nature; and I have argued that the creative act serves primarily to satisfy those needs, not to advance the state of one field or another, though frequently it does that, too.

If I look again at my case histories for evidence of the engine driving them, I find that they could all be described without too much distortion by a simple text that would run something like this:

“He spent years following what had seemed initially to be a potentially open road to his goal, only to find eventually, and to his frustration, that the road petered out in open country. To move on, he had to ask, not “how do I backtrack to where I lost the path?” but “how do I get there from here?” The need to get 'there' was the driving force that kept him moving, though he never knew where 'there' was, or what he would find – what he would be -- if he ever got there.”

Is it possible to come from that position and still believe in the possibility of machine creativity? Certainly! I believe that my dialog with AARON *is* an example of machine creativity, albeit a small one. But considering the question in a realistic light does require a shift away from the anthropomorphic paradigm proposed by Rosum's Universal Robots a hundred years ago and updated in our own age by Commander Data of the Starship Enterprise, a whole genre of movies and a robotics industry that seems intent on giving robots a beautiful skin and a friendly smile. Until Commander Data can have a self rather than a history, machine creativity will

not mean a machine sitting alone in its studio pondering how to re-establish a path to an ill-defined but persistently beckoning goal.