FORTY-FIVE YEARS LATER…

By Harold Cohen

I was in my thirties when I started to fret about the long-term prospects for my painting. I thought it rested too heavily upon my ability to invent formal material, and I was beginning to doubt whether that inventiveness could be sustained indefinitely. Shouldn't it be possible, I wondered, to formulate a set of rules by which paintings could virtually "paint themselves?"

Indeed, it was possible. A group of paintings I showed in the Venice Biennale in 1966 were made by spattering paint onto the canvas to create a kind of "terrain," then applying rules that determined how a line would move through that terrain. I thought of it as analogous to the way the rule of gravity controls the way water finds a path through an area of rough ground.

A good beginning. But not an end.

Two years later I left the UK for a one-year visiting professorship at the University of California, San Diego – I finally retired from teaching there ten years ago – and there, in the early days of a revolution that would change the shape of society, I met my first computer. It's nearly impossible today, in a culture awash with Apples and PC's, to give any sense of what computing was like forty years ago. Computer power has been doubling every eighteen months, roughly, so looking back across twenty-six doublings, you have to try to imagine the unimaginable; a University's main computer having something like one ten-millionth of the power of the PC on which I am typing this text, at hundreds of times the cost. There were no canned programs sitting on the shelves of computer stores – there were no computer stores – and if you needed a program you wrote one, punched it into IBM cards, and came back the next day to see if it had worked.

Strange as it may seem, considering the difficulties, I found programming to be a thoroughly absorbing discipline; which was just as well, because I certainly wasn't smart enough to see the connection immediately between this curious, alien device and my goal of having the painting "paint itself." Even so – and there are many ways to think about what a computer program is – it was hardly surprising that I should think of it as a set of rules for having the computer do something.

Do something? Like what?

Well… Income tax? Chess? Painting?…

Painting??? (Oh, he's the loonie from the art department…)

Eventually computers started to spread through the culture, and opinions about computing spread along with them. During that phase I'd frequently hear people saying things like “they can only do what you tell them to do.” You can see why I’d feel challenged; not by what they were saying, which may very well have been correct. But the implications behind this mantra of human superiority were clear and unmistakable: income tax, certainly; chess, possibly. But painting? Surely not! Art is an expression of the soul, the human spirit. Machines don’t have souls and so they can’t make art.

I’m prepared to agree, as a working premise, that computers don’t have souls. (Nobody has ever found evidence that they do, but that doesn’t prove that they don’t.) But it’s hardly relevant for me, since I’ve never regarded art as an expression of anyone’s soul. Indeed, I’ve never found anything in human-made art that would require a soul to have made it.

I’m also prepared to agree – for the moment – that computers can only do what you tell them to do. But there’s a fine point hidden in that bland statement; because if they can only do what you tell them to do, then the issue is not what they can do but what you are able to tell them. Is there a limit on what you can tell them? If there is I doubt whether we’ve identified it. What is undeniable is that, over the past forty years, one *can’t-do-that* after another has fallen as people have found ways to tell computers to do them.

One of the *can’t-do-that* items that sat for years on my personal list was for my program, AARON, to do its own coloring. It’s not hard to see why it languished there. For people, expertise in coloring rests upon refined color perception, and my program had no visual system. Still doesn’t. Then, around the time an IBM computer was beating the world’s greatest chess player – removing a very large *can’t-do-that* from the public’s perceptions – I figured out that I’d been focusing on the wrong problem. The issue was whether a program could handle color, not whether it could do it in the same way that humans do. I began to see that we’re all hardware dependant to some degree – humans and computer programs alike – and if my program didn’t have physical resources equivalent to those that underpin human expertise, it couldn’t do it in the same way.

That didn’t mean AARON would never be able to handle color, however. Like any computer-based program, my program had resources of its own. Once I’d understood that AARON’s expertise would need to be built in terms of those resources, rather than by trying to emulate resources it didn’t have, coloring came off my *can’t-do-that* list and the way was open to move forward. Not that everything fell into place all at once; it has been almost twenty years since that change of status and much of that time has been spent refining and revising AARON’s color expertise.

I’ve also been refining – no, updating -- my understanding of the goals I’d set myself so long ago. I started with the notion of defining rules so that all I’d have to do was to follow the rules and the painting would “paint itself.” I didn’t think I was changing anything very important, when I met my first computer, to have the computer execute the rules in my place. After all, the truly important part of the whole enterprise lay in defining the rules, not in executing them. But the initial work on color brought with it a subtle conceptual shift; the rules weren't merely executed by the program, they *were* the program. Increasingly, I found myself focusing upon the program's autonomy: what, and how much, could it do for itself?

Has that really changed anything fundamental for me? I think not. It must have been some time in the late 'seventies that I made a remark about being the first artist in history to have a posthumous exhibition of new work. It was intended as a joke. But looking back now I can see that it served as a manifesto, a definition of purpose, not simply setting the stage for what was to come, but identifying what was already well under way.

Where is AARON now? What can it do for itself?

I can start my program running before I go to bed at night and I'll have a hundred original images to look at the next morning. They're all good, some of them much better than good. The color is better – more exciting, more effective -- than I was ever able to do myself.

That's a whole lot more autonomy than anyone would have considered possible ten years ago.

It would be nice if AARON could tell me which of them it thinks I should print, but it can't. It would be nice if it could figure out the implications of what it does so well and so reliably, and move on to new definitions, new art. But it can't. Do those things indicate that AARON has reached an absolute limit on what computers can do? I doubt it. They are things on my *can't-do-that* list, and I'm aware I may never find a way of getting them off. But how can I know what insights tomorrow may bring?

Harold Cohen
Artist
May 2007