Why Use Computers to Make Drawings?

George Whale
Loughborough University School of Art & Design (LUSAD)
Leicestershire LE11 3TU UK
+44 (0)1509 228967
g.whale@lboro.ac.uk

ABSTRACT

In the field of art and design, there are some circumstances in which the use of computers for drawing would seem to confer few tangible benefits; and in situations where computers are productively employed, usage is often tightly bound by convention. Consequently, some practioners doubt whether the technology has anything new to offer them. In this paper, a wide-ranging review of contemporary, computer-mediated drawing leads the author to conclude that such scepticism is unfounded – that computers are not only enabling artists and designers to extend the scope of drawing, but that they are also helping us to understand aspects of the drawing process itself.

Categories & Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces---interaction styles; J.5 [Arts and Humanities]: ---fine arts; K.4.3 [Computers and Society]: Organizational Impacts---computer-supported collaborative work.

General Terms: algorithms.

Keywords: drawing, computer art, digital tools, computer aided design, creative programming, collaborative art.

INTRODUCTION

> "… even if the majority accepts new technology, only a minority truly adopts new practices [p.78, 23]."

With interactive software, even artistically untrained individuals can now create complex shapes with ease, make accurate perspective drawings or render real-world scenes in a range of painterly styles. If this seems unremarkable we should remind ourselves that, not so long ago, these kinds of tasks would have required years of specialist training. Yet some artists and designers remain sceptical, citing the dubious quality of many digital drawings to support their contention that the computer has little of value to offer them. Others are more accepting of the technology, but use it in entirely conventional ways.

It is certainly true that there are many drawing situations in which pencil and paper are still the technology of choice, with good reason: in life drawing, for example, or design conceptualization, the computer may confer few practical advantages. And it seems a safe bet that digital drawing systems will never be as compact and portable as pencil and paper. So, we must ask the question: Why use computers to make drawings?

C&C'02, October 14–16, 2002, Loughborough, Leic, United Kingdom.
Copyright 2002 ACM 1-58113-465-7/02/0010…$5.00.

Computers facilitate a whole range of drawing tasks in art and design: by incorporating elements of pictorial structure into software, which can help to compensate for shortcomings in the artist's domain knowledge, production technique or motor control; and through automation, for when the computer is able to greatly reduce the timescale for task execution, then projects which would otherwise be unthinkable become thinkable. However, in the remainder of this paper it will be argued that there are more compelling reasons for using computers, arising from their ability to spawn genuinely new approaches to drawing and to contribute to our understanding of this most central of human activities.

THEMES AND VARIATIONS – DEFINING AND EXPLORING CONCEPTUAL SPACES

At the level of program source code it is relatively easy to define not only parameterized forms, but also parameterized arrangements of forms. We might say that, conceptually, the software defines an n-dimensional 'space' of graphic possibilities, where n is the number of parameters, and each set of parameter values defines a point in the space, corresponding to a particular form.

Through programming, a number of artists interested in what might loosely be termed 'systems-based' drawing have learned how to define their own 'conceptual spaces' (to borrow Margaret Boden's [7] term). Bratislavan artist Robert Urbásek, for example, has developed software which is able to generate quite complex constructions by repeatedly drawing a line or simple shape, and with each iteration incrementing selected attributes such as size and orientation by specified amounts. A simple theme, with limitless variations; and, like much so-called 'algorithmic', or 'procedural', art, minimally connected with the real world of objects, emotions and human experience.

There are many other ways of introducing (non-random) variability. Some programs, for example, employ 'lexicons' of elemental shapes, variously modifying and combining them according to sets of rules loosely analogous to the rules of morphology and syntax in natural language. Others can change an existing image by changing the underlying data values, or by reorganizing the data structure itself; the types of data structures used largely determine the kinds of changes that can be made. In 'chaotic' systems such as cellular automata, small changes may be enough to produce markedly divergent forms, since it is a defining feature of chaotic systems that they are acutely sensitive to initial conditions – the so-called 'butterfly effect'.

Given the vastness of the space of potential images that may be defined by a generative program, effective search strategies are needed. One common strategy is to start from an arbitrarily chosen point in the space – i.e. a randomly generated image – and if it looks interesting, to explore the surrounding region – i.e., 'tweak' the image data. Other strategies may employ simple, evaluative criteria for filtering out unlikely candidates, thereby significantly reducing the search space. The lack of reliable search strategies remains a significant barrier to the effective use of generative systems in art and design.

NEW TOOLS, NEW PARADIGMS

Traditionally, drawing is an activity in which tools transform movements of the hand, arm and body into marks on a surface. The correspondences between marks and actions are sufficiently constrained that sometimes we feel that we can 'read' psychological states from a drawing almost as easily as we can read them from, say, the shrug of a shoulder, or the turn of a head. Digital drawing, however, is not so constrained, since the input from a mouse or stylus is infinitely transformable by software, as Malcolm McCullough has observed:

> "Software tools introduce great power. It is the singular advantage of the software tool to give visible form and physical action to a logical operation otherwise lacking any physical correspondence, let alone any traditional counterparts [p.80, 23]."

Scott Snibbe has written a series of experimental drawing programs [29] using particle systems. Each program is intended to give "… an immediate sensation of touching an immaterial, but 'natural' world with consistent and predictable reactions, but infinite variety". *Myrmegraph* is based on the emergent group behavior of ants pursuing pheromone trails, whilst *Gravilux*, inspired by Hubble telescope images of galaxies colliding, models gravitational attraction and repulsion: as the cursor is moved through a mass of points, those in the vicinity are pulled towards it, or pushed away. Lia's *re-move* program [21] uses cursor position, speed and direction to control the generation of sequences of lines, dots or shapes which sweep and swirl across the picture surface, laying translucent clouds and veils of light on the black ground of the screen (Figure 1). Here, as in the previous examples, interaction is learnable and ultimately predictable but again, the tools themselves are quite unlike anything we might find in the traditional drawing studio.

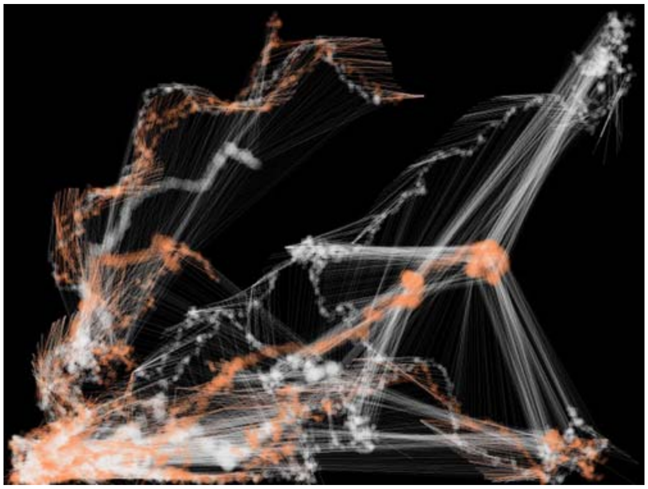

Figure 1. Image created with the interactive, online drawing program, *re-move.* Lia, 1999. <www.re-move.org>.

Most interactive drawing media, like most traditional drawing media, are surface bound, and all thoughts, feelings and responses are channelled through the single point of contact of the mouse or stylus. However, it is not a lack of suitable technology that prevents drawing developing into a truly spatial, synchronous, multi-channel activity, since all kinds of movements can be accurately captured by video cameras and by 3D motion capture devices. Moreover, programs for translating bodily movements into images, though mostly experimental, continue to be developed. The software at the heart of Annunziato and Pierucci's 'artificial life' installation, *Relazioni Emergenti* (*Emerging Relations*) [4], enables observers to interact with a large, back-projected drawing (see Figure 2) made up of "graphic filaments" which germinate, grow, interact and reproduce in response to signs of "energy and life" picked up by a nearby video camera. *Relazioni* also has a musical component: a synthesizer converts the data underlying the continuously changing shapes and patterns into sound, which is to say that both image and sound derive from the same movement data.

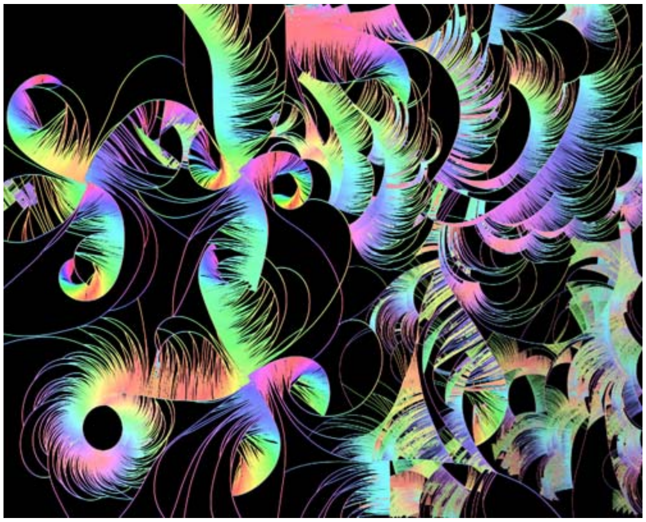

Figure 2. *Volute.* Artificial life drawing. Mauro Annunziato & Piero Pierucci, 2000. <www.plancton.com/entry.htm>.

This ease of translation between modalities – indeed, the ability to realize any kind of digitally encoded process, state or structure as a drawing – greatly extends the range of possible contexts for drawing. Benjamin Fry, of MIT's Aesthetics and Computation Group [2], has experimented with 'organic' representations of dynamic data structures. His creation, *Anemone*, gives a plantlike visual representation of the interconnected structure of a website: it continually grows and branches in response to changes in site structure and usage. In a rather different kind of visualization project [34], Jeremy Wood and Hugh Pryor used a Global Positioning System (GPS) handset to record the paths taken during various journeys by land, sea and air in the UK and abroad. They wrote a computer program, *GPS-o-graph,* to draw the paths in 3D. Two of the resulting drawings are shown in Figure 3.

Specialized programs such as these enable exploration in unfamiliar conceptual spaces, in some cases enabling users to create and manipulate a drawing indirectly, e.g., through music or text. However, as with commercial software, the space of possibilities is entirely defined by the creator(s) of the software.

The issue of software adaptability is important for artist-programmers. Finding that an existing program does something *similar* to what they want, but unwilling to compromise, most would rather modify or augment existing code than spend days or weeks writing something from scratch. Object-oriented programming facilitates the use of 'DLLs' – dynamic link libraries – by which functionality can be added to a program without changing the original code (extensions, or 'plug-ins', used with popular applications such as Adobe® Photoshop®, Adobe Illustrator® and 3D Studio Max® work in this way). This means that an artist wanting, for example, to generate and visualize new classes of 3D forms, can focus on the algorithms – the generative procedures – themselves and let the main application take care of interaction, rendering, printing and so on.

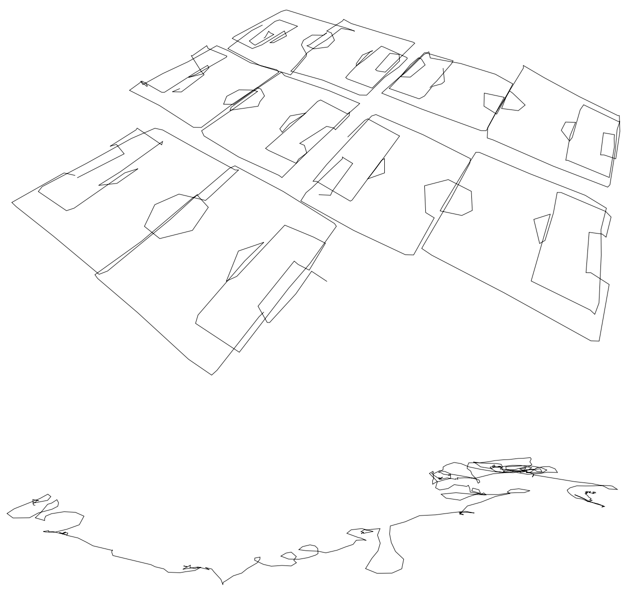

Figure 3. Top: *Hackney football.* GPS drawing of Hackney Marsh, London, UK. Jeremy Wood, 2001. Bottom: *Boris.* GPS drawing made at Summerfields School, Oxford, UK. Boris the dog, 2001. <www.gpsdrawing.com>.

Program output need not be confined to the plane. Bruce Shapiro, who has built (amongst other things) computer controlled devices for drawing onto eggs, and for etching into sand, lays down the challenge: "Now ... instead of a pen, why not an engraving bit, router, cutting torch, laser, ...? [28]." To this list we can add stereography, holography, various electronic 3D-display technologies under development, and some of the new, rapid prototyping technologies such as stereolithography, by which three-dimensional gestures can, in principle, be made fully concrete.

APPLYING IDEAS FROM OTHER DOMAINS

Given the pace of contemporary scientific research and development and the diffusion of scientific ideas into the public domain, it is not surprising that some of those ideas have been co-opted in the service of art. The natural sciences, in particular, provide inspiration for visual artists. Phenomena of self-organization, for example, which are of interest to biochemists studying the conformation of large molecules and to ethologists studying the flocking and schooling behaviours of animals, have been modelled computationally and applied in computer graphics [26]. Self-organizing systems suggest that hierarchical, 'top-down' planning may not be the only way of producing interesting or complex pictorial structure, since they rely only on simple, localized interactions of elements. The potential of such an approach was recognized by John Lansdown (one of the early pioneers of computer art in the UK) who envisaged drawings growing "… from cell-like elements whose presence or absence and position are determined by rules which depend entirely on local considerations and not on the drawing as a whole [9]."

Even if a drawing system is based on some natural system, there is no obligation to obey natural laws. As Stephen Wolfram has noted: "… nature must follow the laws of our particular universe. Yet programs can follow whatever laws we choose [33]."

Thus, for example, when evolutionary ideas are applied to the making of art, the picture data structures rarely bear any resemblance to real genes, mutation rates are unfeasibly fast, and 'aesthetic selection' replaces natural selection as the driving force. These are the principles underlying Ellie Baker and Margo Seltzer's evolutionary drawing system [6], which "… mates or mutates drawings selected by the user to create a new generation of drawings" (see Figure 4). Such drawings are not designed or preconceived, in the usual sense; only the initial data, algorithms and rules by which they are made. As Karl Sims, creator of some strikingly original evolutionary graphics software, has said: "A Darwinian process can create mysterious things without needing anyone to understand them … [14]."

Figure 4. *Evolving Line Drawings.* Average face (left) and three of its descendants. Ellie Baker & Margo Seltzer, 1994.

Some influential concepts in computer-based drawing have come from linguistics. Natural language has long provided a metaphor for talking about visual art, but it was only with the advent of computers that workable implementations of so-called 'shape grammars', comparable to natural language grammars in their ability to generate diverse, rule-governed structures, became possible. Shape grammars have been used for product design and architectural design, and for the generation of 'original' drawings in the styles of particular artists, using rules derived from detailed analyses of selected works [17, 18]. Grammars may be highly productive, which is to say that they may be capable of generating huge numbers of configurations, and a significant problem in the design of interactive systems is how best to support the user's selection decisions. Essentially, this is the conceptual space navigation problem referred to earlier.

NEW DEPICTION TECHNIQUES

Non-photorealistic rendering (NPR) has been an area of active research and development for a number of years and is concerned with creating alternatives to photographic modes of depiction, which predominate in computer art and graphics. In its least interesting manifestations, NPR enables users of image-processing or painting programs to apply pseudo-painterly styles to photographs. It becomes more interesting when it attempts to isolate and process particular features of an image, as in some of the 'pen and ink' rendering software [32] and more interesting still when lines, marks or shapes are applied directly to a 3D model, where they can be configured so as to convey information about relative depth, surface orientation and structure [19].

NPR is not only of interest to artists. Researchers in technical computer graphics are learning how to use line to improve the comprehensibility of digital representations [27], and how to isolate elements of drawing 'style' so that they can be automatically applied to any image or model [15].

Philip Rawson, in his influential and wide-ranging study of drawing [25], showed that marks and combinations of marks may serve a multiplicity of functions within a picture, including representational, expressive and decorative functions. NPR researchers, in their efforts to understand the structures and functions of drawn marks – how to create and configure them so as to convey the information they need to convey – suffer no restrictions on the kinds of marks they can use, since digital marks and their attributes are individually modifiable and endlessly permutable. To the extent that NPR programs 'know' how to apply marks in particular kinds of drawing situations, they might be considered to embody particular 'mark-making strategies'; and strategy, of course, is an important component of drawing knowledge.

REMOTE COLLABORATION

Tom Brinck described collaborative drawing as follows:

> "2 or more people conversing with the use of drawings or working together to create a drawn artifact. Conversing and creating an artifact are very different types of tasks, though there exists a continuum between them, such that it's difficult to produce a successful collaborative drawing tool that doesn't support both to some extent [8]."

Different kinds of collaborative drawing programs can be found on the Web. Few of them support conversation, and in many cases collaboration is strictly sequential, a matter of loading an existing drawing from an archive, modifying it, then saving it to the archive for other users to modify later.

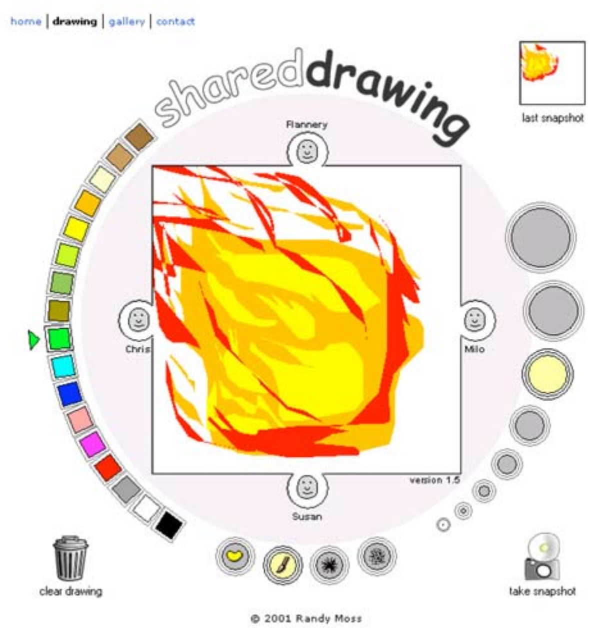

Figure 5. Interface of the *Shared Drawing* website. Randy Moss, 2001. <www.shareddrawing.com/index.html>.

By contrast, Randy Moss's *Shared Drawing* website [24] enables truly synchronous collaboration. Each partner is represented on the interface (Figure 5) by an animated iconic proxy that "draws along with you", and successive drawings, created by any number of online artists in different locations, are saved to create what Moss calls an "ever evolving hand-drawn film".

Vectorama [20], described by its originators as a "multiuser playground", allows up to ten people at a time to create a picture together, but using predefined elements (vector graphics of people, objects, backgrounds, etc. selected from libraries) which may be individually scaled, rotated and coloured. *Vectorama* users may also communicate with one another via a text messaging system.

The experience of using these kinds of programs can be quite engaging (and may produce interesting artworks) though hardly comparable to real-world situations where two or more people use a whiteboard, or a single sheet of paper, on which to record and develop ideas together. In the real world, a shared drawing can promote a rich dialogue in which the activities of drawing and annotating are augmented by speech, facial expression and body language. It is a way of working common in design conceptualization, but until recently has required participants to be in the same room.

Clearboard, developed by Kobayashi and Ishii [30], comprises two networked terminals and a video connection supporting concurrent, "face-to-face interaction" between two collaborators. The drawing surfaces are large, half-silvered displays, onto each of which real-time pictures transmitted from the other location are back projected. The effect is of a large pane of glass through which the other person can be seen drawing on the reverse side. The system supports "gaze awareness" – each person can see which part of the drawing the other is attending to, and 'collisions' are thereby avoided. In addition, a videoconferencing function enables the collaborators to speak to each other.

As *Clearboard* and similar systems become more widely available, new drawing practices could emerge. The results are difficult to predict:

> "… collaboration and communication produce surprising results. … If a fundamental reciprocity can be maintained – a balance of getting and giving – what emerges is undoubtedly unforeseen and potentially exciting [11]."

UNDERSTANDING REAL-WORLD DRAWING

Harold Cohen's now legendary program (or rather, family of programs), called AARON has grown during the last three decades into probably the most knowledgeable of knowledge-based, fine art drawing programs. It operates according to a complex hierarchy of rules which enable it to decide what actions to take in a given drawing situation. At the highest level, overall organizational decisions are made; at the lowest level, decisions about the forms of individual lines and shapes. The first versions of AARON drew abstract shapes, later versions drew figures: all were capable of generating innumerable new compositions [22].

AARON possesses two main types of stored knowledge, which interact with each other: specific knowledge about how to draw certain classes of objects (rocks, plants, human figures, etc.); and general knowledge of representation, e.g., how to depict solidity or occlusion. This knowledge derives from Cohen's long experience of drawing practice, and it is no surprise that the program possesses a distinctive style. An unusual feature of the program's operation is feedback, which is to say that in making decisions it takes account of the current 'state' of the drawing. This, of course, is a crucial aspect of most human drawing activity.

One of Cohen's motivations was to arrive at a better understanding of drawing processes by teaching a computer how to draw, and to the extent that AARON models thinking processes as well as motor processes, we might say that it embodies a cognitive model of drawing, albeit one intimately bound up with Cohen's own ideas and restricted to drawing from memory (since there is no 'visual' input).

Another notable success in computational modelling is Ed Burton's ROSE program, described as "… a tool to both illustrate process-oriented theories of children's drawing and guide new ideas for their further development". The program's outputs are derived from 3D representations stored in memory, and are uncannily reminiscent of young children's topological drawings of human and animal figures. Despite the success of his model, Burton recognizes its limitations, and identifies drawing *development* as perhaps the most difficult problem to contend with:

> "ROSE's … drawing behavior is fundamentally different from that of a real child because ROSE's drawings do not develop. … ROSE's ability never changes without the intervention of a computer programmer [10]."

Computers and people have very different strengths and weaknesses. Unlike us, computers can perform masses of complicated symbolic manipulations quickly and flawlessly. Computer vision systems can pick up (if not understand) every detail of a scene in an instant, whereas human vision appears to be highly selective: we scan the scene, fixating on salient parts and extracting from the retinal stimulus only the minimum information needed for the task at hand. In memory retrieval, we outperform computers, being able to retrieve salient information almost instantaneously from a long-term memory (LTM) of practically unlimited capacity, which accounts in part for our astonishing powers of recognition, far superior to those of any computer. We are comparitively bad at making *absolute* judgements of shape, size, colour and so on, but good at making *relative* judgments. And although we take time to learn new motor skills, we are far better than any robot at adapting those skills to new situations.

Given these, and other, fundamental differences, it seems unlikely that algorithms designed for computer implementation are anything like our own 'algorithms' for doing the same things: far more likely that human drawing expertise develops in ways which exploit our inherent strengths. Any model of drawing, if it is to be plausible, must be consistent with current (scientific) understanding of human vision, cognition and action. Whilst we should not underestimate the difficulty of the task, we can take encouragement from recent advances in the modelling of human performance. Byrne and Anderson's implementation of their *ACT-R/PM* theory [3], for example, demonstrates the feasibility of modelling real-world perceptual-motor tasks. It uses production systems, which model the execution of hierarchical plans by means of rules, or 'productions', each of which triggers a particular action when a particular set of conditions is met. The implementors regard a production as "a basic step of cognition". Given a suitable set of them, complex sequences of actions can be performed.

One way of obtaining insights into the mental processes associated with drawing is by introspection, but it is not at all clear how reliable introspection is as a method of knowledge elicitation. Indeed, cognitive scientists agree that there are some aspects of cognition which are forever impenetrable to introspection. Future research in this field might well heed the experience of knowledge engineers: those whose job it is to elicit specialist knowledge from experts for incorporation into (mainly scientific or commercial) 'expert systems'.

ACCUMULATING KNOWLEDGE

If we consider graphical computing in terms of data and processes, then it is clear that much graphical interaction is concerned with selecting subsets of the image data and selecting processes to be applied to that data. For example, in a photo-manipulation program, any region of interest can be modified by using a shape selection tool to isolate it, then searching the menu hierarchy for a processing option, e.g., contrast enhancement or halftoning. Similarly, in a 3D modelling program, objects or object parts may be selected and operations applied to them: for example, Boolean operations such as union and intersection can be applied to pairs of objects. Data (hence objects and images) may be transformed, restructured, combined or abstracted in many other ways, and the kinds of operations possible will depend to some extent on the kinds of data structures used (lists, arrays, trees, graphs, etc.).

First, there must be some data to work with, and in most cases this will be derived from bodily movements, usually via a mouse, stylus or similar input device. Streams of coordinates constituting movement paths are turned into patterns of pixel-mapped marks, geometric shapes or virtual solids by some (pre-selected) process. Data can also be acquired from other sources, such as image files or other kinds of files (e.g., text or music, though the data may need to be restructured to make visual sense of it) or from some kind of generator within the program itself, e.g. a (parameterizable) procedure for producing spirals, or a shape grammar (whose lexicon and syntax may be user-definable).

In summary, the points at which the artist/user can exert control and apply his/her knowledge are: in the selection of data and the selection of processes to act on the data; in the creation of data and (to some extent) in the configuration of processes. The computer, in turn, runs the processes to produce new, or changed, images.

The term 'knowledge' is difficult to define, and even in computer science circles the precise relation between data, information and knowledge remains a source of confusion and disagreement, as Aamodt and Nygård [1] have noted. However, if we concur with them that knowledge is something "incorporated in an agent's reasoning resources", something which "is brought to bear *within the decision process* itself", then such attributes as expertise, experience and intuition all fall within its scope. In computer-mediated drawing, the user's knowledge (which must include some knowledge of the software and its functions as well as knowledge of depiction) is brought to bear during interaction. But it is clear that *algorithms* can also embody expert knowledge of drawing, including: knowledge of the various 'drawing systems' [12] such as perspective and orthogonal projection; knowledge of what might be termed 'mark-making systems', ways of configuring marks so as to convey representational, or other, information; and, as we saw in AARON, knowledge of 'strategy', by which appropriate and timely decisions are made concerning the form and disposition of elements at each stage in the drawing's development. (As yet few, if any, drawing programs are capable of gathering or generating *their own* knowledge.)

The crucial question in relation to software-embedded knowledge is: Where does it come from? For it could be argued that a drawing is an artist's own only to the extent that he/she has applied his/her own knowledge in the making of it, whether through interaction or programming. James Elkins has argued that computer graphics tools incorporate a number of pictorial conventions:

> "Computer graphics is inextricably linked to the history of Western picture-making. The expressive meanings, artistic strategies and conventions of that genre continue to underwrite developments in computer graphics, especially when they are not acknowledged [p. 335, 13]."

To be fair, they often are acknowledged: in NPR research papers, for example, we can find numerous references to Impressionist painting, Renaissance and Baroque engraving and the like. Nevertheless Elkins' point is a valid one, and these influences can be distracting. Clearly, the artist-programmer (or artist *plus* programmer) enjoys an advantage over the artist-user in being able to *create* the data structures and algorithms, to define and explore conceptual spaces of his/her own. But an equally valuable consequence of this way of working is that aspects of drawing knowledge are necessarily made explicit, as artist Roman Verostko has found:

> "When you start working out a procedure for generating a new form you are forced to think in elementary ways about the very nature of the drawing process. … you learn a lot *about how to draw* by writing *drawing code* and you learn a lot about how to write *drawing code* by *drawing* [31]."

In making the work, the artist also describes his/her method, and the description is in a form that can be studied and analyzed, adapted or developed by others.

THE FUTURE

Why use computers to make drawings? In addressing this question it has not been my intention to persuade anybody that computers *should* be used to make drawings because, in the right hands, traditional drawing media remain as flexible and as eloquent as ever. But in showing some of the ways that computers *can* be used, both to extend the scope of drawing and to help us understand some aspects of drawing processes a little better, I hope to have dispelled some doubts.

The computer is useful only to the extent that it supports the drawing enterprise, and it must be acknowledged that there are situations in which its adoption may confer little, if any, practical benefit (though it may still be an invaluable research tool). One such situation is the life studio. However, in design fields at least, it seems likely that computer-mediated drawing will become increasingly important. Today's passive 'drawing environments' could eventually be displaced by what have been called 'active drawing support systems' able to provide users with valuable, context-sensitive information; to indicate appropriate strategies, media and techniques; to identify errors and suggest alternatives; or perhaps even to evaluate drawings according to functional or aesthetic criteria. The key to such developments will be the embodiment of (generic and domain specific) knowledge in software, and we may one day see systems able not only to apply knowledge but also to learn, so that knowledge gleaned from previous drawing situations can be adapted and applied in new situations. Thus, the promotion of the computer from tool, to assistant, to fully-fledged collaborator would be complete.

At present, relatively few artists design or write programs, but as more intuitive, visually-oriented programming languages become available this could change, enabling more of them to contribute to the growing body of explicit drawing knowledge. This knowledge, in turn, could facilitate the development of more 'intelligent' support systems and, more importantly, might bring us closer to an understanding of what it is that makes one artist different from another.

Inevitably, as new technologies expand our notions of what drawing is, or could be, traditional practices, as exemplified in the life studio, are called into question. The 'language of drawing', to use Edward Hill's term [16], looks more like a babel of languages, of which observation-based drawing is but one. A pre-eminent one, perhaps, if we believe that it alone can sharpen powers of visual discrimation, but of diminishing relevance according to some, such as Roy Ascott:

> "The practice of life drawing, sometimes called 'objective drawing', is defended as offering a complex structure against which hand-to-eye coordination can be perfected. But is it not hand-to-mind coordination we should seek? [5]"

It seems possible that drawing will come to be associated with quite different sets of practices than those we are accustomed to, perhaps more attuned to developing the skills of "hand-to-mind coordination" which Ascott talks about. Whether or not this turns out to be the case we can be certain of one thing, which is that true originality in drawing – the 'X' factor that separates the very best artists from the rest of us – will remain the province of the human imagination, at least for the time being.

ACKNOWLEDGMENTS

I would like to thank all of the artists and researchers whose stimulating images and ideas provided the impetus for this paper.

REFERENCES

1. Aamodt, A. and Nygård, M. 1995. Different roles and mutual dependencies of data, information, and knowledge: an AI perspective on their integration. *Data and Knowledge Engineering* 16, 191-222.
2. ACG (the Aesthetics and Computation Group). URL: <http://acg.media.mit.edu>. (Accessed 22 January 2002).
3. Anderson, J. R. and Lebiere, C. 1998. *The Atomic Components of Thought.* Mahwah, NJ: Lawrence Erlbaum Associates, 167-200.
4. Annunziato, M. and Pierucci, P. *Relazioni Emergenti.* URL: <www.plancton.com/entry.htm>. (Accessed 31 January 2002).
5. Ascott, Roy 1988. Art and Education in the Telematic Culture. *Leonardo* (Electronic Art Supplemental Issue), 7-11.
6. Baker, E. and Seltzer, M. 1994. Evolving Line Drawings. *Proceedings of Graphics Interface '94* (Banff, Canada, May 1994), 91-100.
7. Boden, M. A. 1990. *The Creative Mind: Myths and Mechanisms.* London: Weidenfeld and Nicholson.
8. Brinck, T. *Usability Glossary: collaborative drawing.* URL: <www.usabilityfirst.com/glossary/term_261.txl>. (Accessed 22 January 2002).

9. Bruton, D. 1999. Images within Images: computational generative imagery using a contingent sense of grammar. In *Proceedings of First Iteration* (Monash University, Australia). (CD-ROM).

10. Burton, E. 1997. Artificial Innocence: Interactions between the Study of Children's Drawing and Artificial Intelligence. *Leonardo,* 30, 301-309.

11. Deck, A. *Icontext.* URL: <www.artcontext.com/icontext/html/about.html>. (Accessed 22 January 2002).

12. Dubery, F. and Willats, J. 1983. *Perspective and Other Drawing Systems.* London: The Herbert Press.

13. Elkins, J. 1994. Art History and the Criticism of Computer-Generated Images. *Leonardo,* 27, 4, 335-342.

14. Frauenfelder, M. 1998. Do-It-Yourself Darwin. *Wired*, 6, 10. Available at URL: <www.wired.com/wired/archive/6.10/sims_pr.html>. (Accessed 31 January 2002).

15. Hamel, J. and Strothotte, T. 1999. Capturing and Re-Using Rendition Styles for Non-Photorealistic Rendering. *Proceedings of EuroGraphics '99* (Oxford, UK), 173-182.

16. Hill, E. 1966. *The Language of Drawing.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

17. Kirsch, J. L. and Kirsch, R. A. 1988. The Anatomy of Painting Style: Description with Computer Rules. *Leonardo*, 21, 4, 437-444.

18. Lauzzana, R. G. and Pocock-Williams, L. 1988. A Rule System for Analysis in the Visual Arts. *Leonardo*, 21, 4, 445-452.

19. Lee, P. 1997. Linear colour contouring for fine art printmaking. *Visual Proceedings of Siggraph* (Los Angeles, CA), 185.

20. Lehni, J., Lehni, U. and Koch, R. *Vectorama.org: A Multiuser Playgroun*d. URL: <www.vectorama.org>. (Accessed 22 January 2002).

21. Lia. *re-move.* URL: <www.re-move.org>. (Accessed 22 January 2002).

22. McCorduck, P. 1991. *AARON's Code: Meta-Art, Artificial Intelligence and the Work of Harold Cohen.* New York: W. H. Freeman and Company.

23. McCullough, M. 1996. *Abstracting Craft: The Practiced Digital Hand.* Cambridge, MA: MIT Press.

24. Moss, R. *Shared Drawing.* URL: <www.shareddrawing.com/index.html>. (Accessed 22 January 2002).

25. Rawson, P. 1969. *Drawing: The Appreciation of the arts.* London, New York: Oxford University Press.

26. Reynolds, C. *Boids: Background and Update.* URL: <www.red3d.com/cwr/boids>. (Accessed 31 January 2002).

27. Saito, T. and Takahashi, T. 1990. Comprehensible Rendering of 3-D Shapes. *Computer Graphics* (Siggraph Proceedings), 24, 4, 197-206.

28. Shapiro, B. *The Art of Motion Control: Process overview.* URL: <http://www.iaxs.net/~bshapiro/home.htm>. (Accessed 29 July 2002).

29. Snibbe, S. *Dynamic Systems Series.* URL: <www.snibbe.com/scott>. (Accessed 22 January 2002).

30. Stocker, G. and Schöpf, C. (eds.). 2001. *Ars Electronica 2001. Takeover: who's doing the art of tomorrow.* Vienna, New York: Springer, 253.

31. Verostko, R. 1994. *Algorithms and the Artist.* Available at URL: <www.penplot.com/alg-isea94.html>. (Accessed 22 January 2002).

32. Winkenbach, G. and Salesin, D. H. 1994. Computer-Generated Pen-and-Ink Illustration. *Computer Graphics* (Siggraph Proceedings), 28, 4, 91-108.

33. Wolfram, S. *Graphica Defined.* URL: <www.graphica.com/defined>. (Accessed 22 January 2002).

34. Wood, J. and Pryor, H. *GPS Drawing: the Global Positioning System drawing project.* URL: <www.gpsdrawing.com>. (Accessed 22 January 2002).