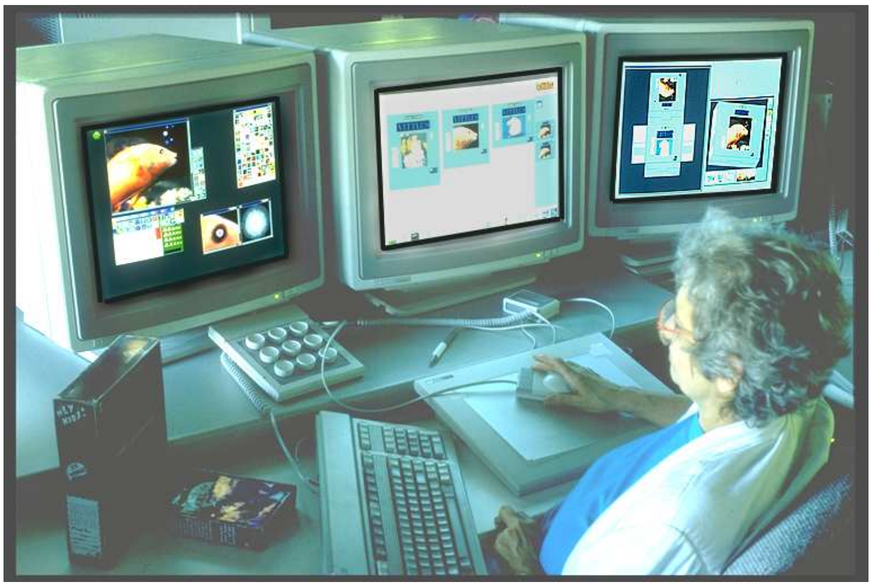
Intelligent Graphics:
Artificial Intelligence at the Visible Language Workshop

DRAFT

Henry Lieberman



For many graphic designers and artists of the last generation, the idea of computers entering the design process was viewed as threatening. The exploration of visual forms and sense of communication in visual design was viewed as essentially and uniquely human, and the incursion of the computer into the process was viewed by many as demeaning the role of the creative professional. Of all the branches of computer science that proposed relevance to visual design and communication, artificial intelligence was perhaps the most threatening of them all, since it proposed to model the model aesthetic judgment, which, in a pessimistic view, might leave little role for a human designer.

A small handful of visual designers, however, saw it very differently. A lot had to do with a designer's personal view of the design process itself. The prevailing culture often viewed design as magic, designers as magicians. But some viewed the design process as a constant search for the relation between form and meaning, a continuing exploration of possibilities for expression. They welcomed tools that could assist a designer in relating form and meaning, in exploring and evaluating alternatives. They saw the possibility for the computer to become an intelligent partner in the design process, to extend the reach and grasp of the human designer. From that point of view, the concerns of artificial intelligence in representing meaning, inference, and learning are not foreign to the design process. They are central.

And it was not always the most technologically oriented among designers that had the greatest enthusiasm for the potentials of the new technology. It was the vision of new, dynamic, intelligent, image-making tools that drove a few designers, educated in a culture that was artistic, not technological, to put up with learning the often frustrating details of computers, then as now.

Muriel Cooper was one of those rare pioneers, who had a vision of computer-based tools that could actively participate in the process of dynamic visual expression. She saw no inherent boundary between the visual design process and the software design process, only one imposed on us by the imperfections of our current technology. She was often iconoclastic, challenging the unstated assumptions of both visual designers and computer scientists.

Generating design alternatives

In Muriel's view, the computer is a new design medium, a new kind of brush, a new kind of canvas, a new kind of palette. There is no more reason for a designer to fear computers than for any artist to fear their tools. You learn them and work creatively within their strengths and limitations. The particular strength of the computer as a design tool is its ability to respond dynamically to the user, to remove the burden of tedious, precise or repetitive operations, to become personalized to its users' needs and style of working. Muriel was attracted by Nicholas Negroponte's original vision of the Architecture Machine as an intelligent assistant for architectural design, and envisioned a similar role for an intelligent assistant for visual design tasks.

The Visible Language Workshop

Muriel's concern for supporting the design process with intelligent tools led her to make artificial intelligence one of the major themes of her research group, the Visible Language Workshop. Certainly, the AI community paid little attention to visual design problems during the 70s and 80s. It preferred to concentrate its attention on applications in such areas as engineering problem solving and medicine, which were felt to be "more well understood" and have a more immediate real-world payoff. The computer graphics community looked mainly toward involvement of the computer in image production aspects and not toward the computer as a tool for conceiving new designs.

Muriel made the Visible Language Workshop a meeting ground for people from diverse backgrounds who shared a fascination for the computer as a place where technology and design could converge. Her collaborator in starting the VLW was Ron MacNeil, who came from a photography and printing background, and who also became a strong proponent of the AI approach to design. Designers who had little familiarity with the technology encountered computer scientists who hadn't yet given serious thought to problems of visual design and expression, and both sides learned enormously from one another. The VLW embodied the ideal of interdisciplinary spirit in which the Media Laboratory was founded.

A personal note

It was this spirit of interdisciplinary collaboration that brought me to the Visible Language Workshop. I had started out at the MIT Artificial Intelligence Laboratory with Seymour Papert's Logo group, which had the then-radical notion of using ideas and computer systems developed for artificial intelligence work to teach children about problem solving. The most popular activity for the kids was the *turtle*, first a robot, then an imaginary agent on a graphics screen that could be commanded to draw pictures. My job was to program the computer graphics, and I had written the first Logo system for then-new raster graphics screens, and the first color Logo system. I was struck by how engaging the computer graphics domain was for the kids and by the power of combining both visual and abstract problem solving.

The kids programmed math and physical simulations, but art was also a major activity. I had exhibited some computer art in the Siggraph conference, and some of my students won prizes in Byte magazine's first computer art contest. My first official VLW event was when I was invited by Muriel to participate in a panel on computer art, along with Russell Kirsch, who constructed AI simulations of famous artists' styles. Russell, in his sweet and gentle manner, tore me to shreds for my naivite about artistic traditions and concerns. There was something exciting about the interplay between artistic and computational concerns that I hadn't encountered elsewhere. The panel was part of Muriel's annual summer course, The New Graphics, which introduced people from a variety of backgrounds to computer image making. The activities in this course represented some of Muriel's pioneering effort which established the field of what is now called electronic publishing. The course used the sometimes cranky, sometimes miraculous VLW image editing system, called Sys. The encounter between people of design and computer backgrounds was, for me, exhilarating.

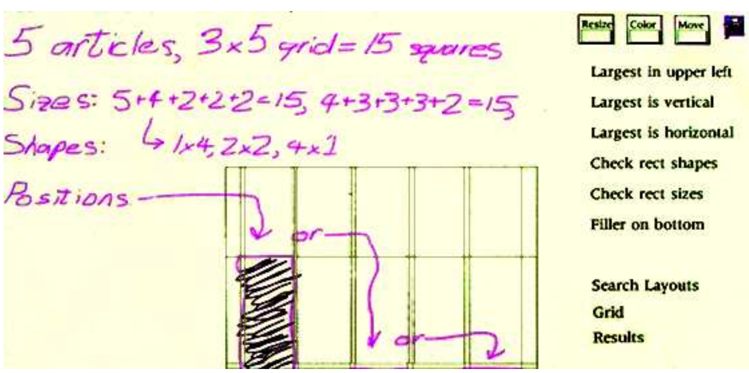
Continuing contact with Muriel and Ron through the incorporation of the VLW of the Media Lab, I started to teach in the introductory VLW course, and participate in thesis committees for VLW students. The course bore the rather nondescript title Computer Graphics Workshop, but it was much more than that. Taken mostly by students immediately after joining the VLW, it was "boot camp" in the VLW philosophy. While much of the course was conducted in the manner of a traditional design course, with design exercises, projects and critiques, Muriel also wanted the students to gain some background and experience in AI issues, and I supplied that perspective, along with Ron and Patrick Purcell, who had done seminal work in studies of design process in architecture. I was part-time between the Media Lab and the AI Lab for a while; then, feeling increasingly that the exciting directions for the future lay in the convergence of artificial intelligence with interactive interface design issues, I joined the VLW full time in 1987.

Threads of VLW research

There were several recurring themes in VLW research projects in which AI techniques played a central role. We'll explore some of these threads in the remainder of this paper, surveying some student and staff projects that address these themes. Among the themes are automatic layout, intelligent assistance for the construction of visual designs, learning by example, visual representation of knowledge, and other projects that don't fit squarely into these categories.

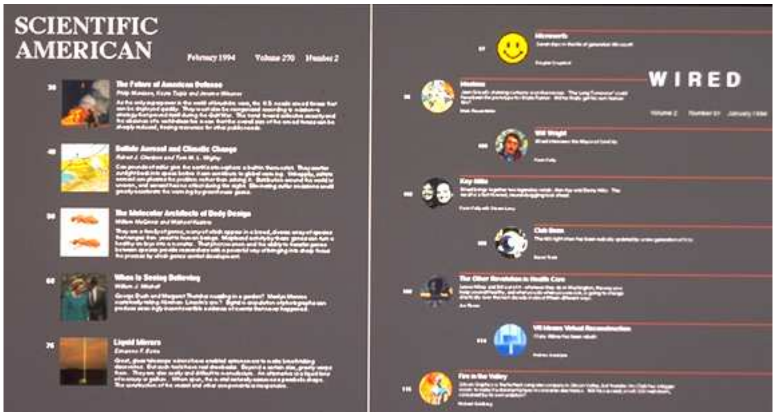
Automatic layout

One of these has always been *automatic layout*. It was recognized quite early that, no matter how good electronic tools for image editing, image composition, and screen design for interactive interfaces get, it is unreasonable to expect that a human designer's participation would be required for each and every screen that the user sees, or each and every page of hardcopy. Digital data simply comes too fast, and is too dynamic to have every visual presentation designed by hand. An electronic news service cannot have a human hand-design each screen when stories are coming in minute-by-minute. Some sort of automatic layout is required.



Layout from sketches

Projects in automatic layout took place in many domains and with several different approaches, explored by many VLW students. Students Timothy Shea, Tom Amari and Alka Badshah were early explorers of rule-based systems for design, with business graphics and packaging being domains of application. Russell Greenlee used sketches of principal elements of layout as a guide, and his system generated sequences of possible layouts for the user to choose from that were consistent with expressed constraints. I had designed an automatic layout system that used best-first search to deal with the common problem of overconstrained space allocation. More recently, Grace Colby applied the techniques of case-based reasoning and constraints to the automatic layout problem. B. C. Krishna looked at the problem of automatically detecting layout constraints from scanned images. Louis Weitzman pursues a linguistic grammar-based approach, with the aid of a parser that can enforce geometric and temporal constraints during automatic layout.



Automatic conversion of layouts

Visual representation of knowledge

Applying concepts from AI programming to visual design problems such as layout was not the only concern of the VLW. Equally important was going in the other direction: applying visual design ideas to the process of constructing AI programs. Muriel expressed constant frustration at the lack of visual imagery in the programming process itself, even when the subjects of the programs themselves were computer graphic images, or when representing knowledge about visual design. Conventional textual programming languages and environments put up a barrier for people who thought of themselves as visual thinkers against expressing themselves in the computational medium. So the VLW also pursued some work in visual programming and visual representation of knowledge in AI.

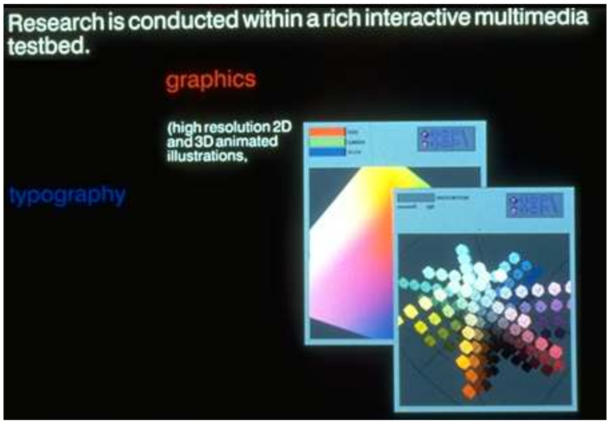
A 3D program representation

I led a project in investigating alternative visual representations, including 3D programming languages, and use of color and translucency. Ron MacNeil's early Tyro system used a representation of visual design elements connected by "spider webs" representing constraints. Projects by students Dorothy Shamonsky, Ming Chen, and Michelle Fineblum also explored issues of visual representation in programming. Storyboards containing visual examples of states of a program appear in my Mondrian system and in Louis Weitzman's VIA. Visualization of rules, constraints and graphic relationships continues to play an important role in several VLW projects.

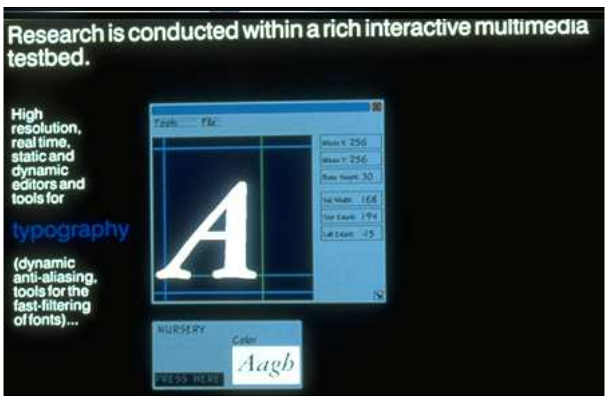
AI tools for interactive design

AI techniques can be used to provide intelligent assistance for visual designers in the process of constructing both static images and interactive media. Designers who use interactive image editors and multimedia editors can benefit from having tools which explicitly support the design problem solving process. This involves building some representation and understanding of the design process into the computational tools.

Muriel provided the insight of a top design professional into the process of design. A task for me and for her VLW collaborators was to "knowledge engineer" some of Muriel's design expertise, and the design expertise she brought to VLW in the form of visitors from the design world and reference materials. The knowledge engineering was trying to come to understand enough about the design process to understand what aspects of could be embodied in intelligent interactive tools. We still have not come very far, relative to the creative powers of an expert designer, but Muriel aided greatly in taking some innovative steps towards intelligent design tools.



Two views of an adaptive presentation

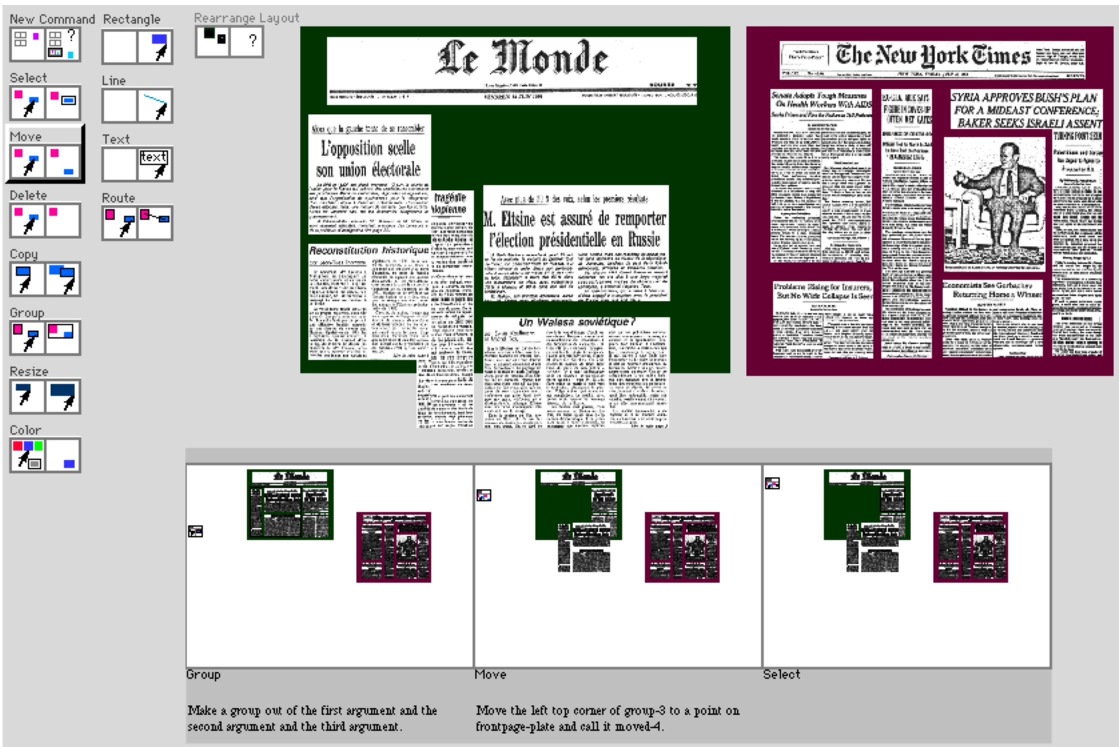


Debra Adams used the observation that type designers often base their designs on a few prototypical letterforms, and she constructed a system that would automatically design a complete font after having been shown designs for the prototype letters. Craig Kanarick built knowledge about the design of charts and spatial data into a case-based presentation tool. Laura Robin, at my suggestion, built a best-first search engine into a hypermedia browser, so that the time and level of detail of a presentation could be automatically adjusted to produce the best possible presentation in a given amount of time and subject to expressed interest in a subset of the topics. Michelle Fineblum also explored adaptive hypermedia presentations.

Learning from examples

A particular interest of mine in both artificial intelligence work and in studying the design process has been *learning from examples*. As a teacher, I have always been firmly convinced that the best way to learn is by example, and as a learner, my personal style has always been to concentrate on learning from examples. One of the most important lessons about visual design and communication that I learned from Muriel was the extent to which design is based on the generation and critique of concrete visual examples.

Open up any book on graphic design, either one intended for teaching beginning design or for communication between seasoned professionals. Unlike books in science and engineering, few principles will be explicitly stated in the text; there will be no equations, no axioms. So how can design students use these books to effectively solve problems? What the books will provide is an abundance of examples: examples of exemplary designs, examples of design variants [or "near misses" in AI terminology]. Good design students are experts in generalizing examples and making analogies from examples to new problems they encounter.

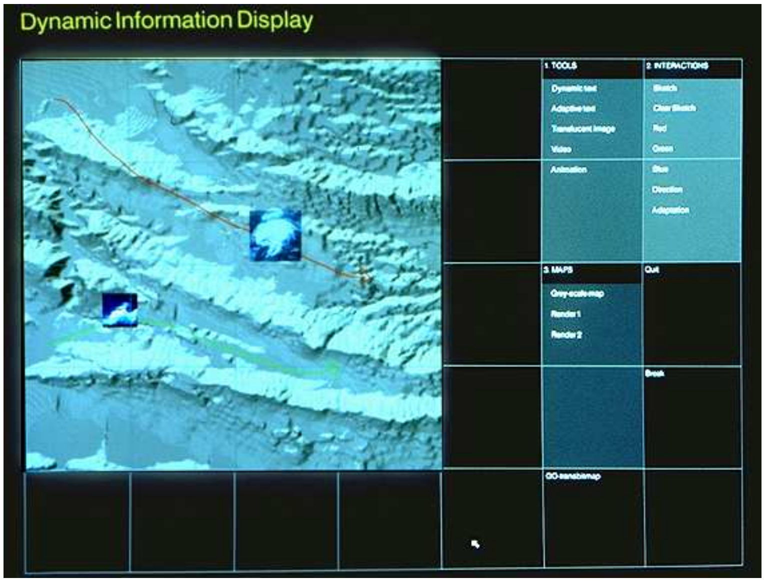


Learning from examples

But why can't we use concrete visual examples in teaching our computers how to do designs, instead of incomprehensible programming languages and rule languages? It is this question that led me to work on the idea of *programming by example.* In this approach, a designer can use an interactive interface, and a software agent records the actions performed by the user, and can generalize them so that they can be applied to analogous problems in the future.

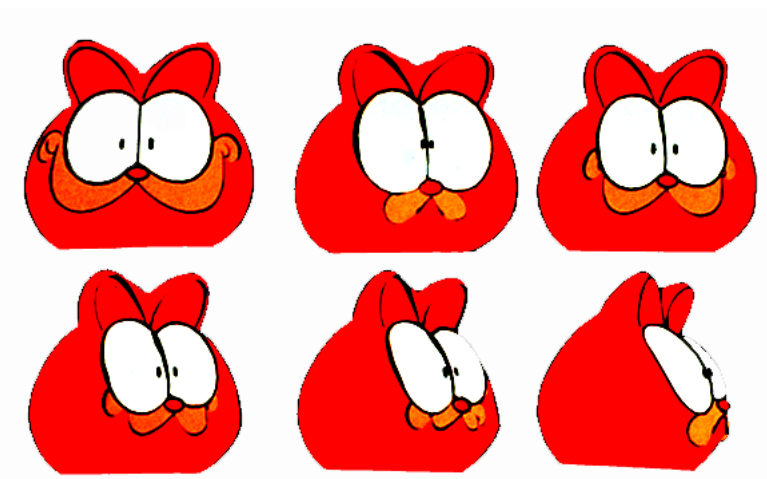
I [and others in VLW] built several systems that incorporated this approach. Mondrian is an object-oriented graphical editor that can learn new graphical procedures by example. Its base is a MacDraw-like graphical editor, a familiar tool for design professionals. Its learning agent could also be applied to many other sorts of image editing and media editing tools. Mondrian incorporates a learning agent that records and generalizes procedures presented as sequences of interface commands. The user selects objects to represent the examples, and demonstrates a procedure which depends on those objects. The user interface is extended with a new operation that can be applied to different objects in the future.

Mondrian uses a consistent visual language for communicating with the user. Operations are represented with dominoes, before-and-after pictures representing a visual example of the operation, and storyboards, a comic-strip like sequence of frames. It automatically produces natural language descriptions, both written and spoken, and accepts advice from the user via speech recognition. I also built on my previous work with a system called Tinker, which allowed example-based programming with multiple examples, conditionals and recursion.



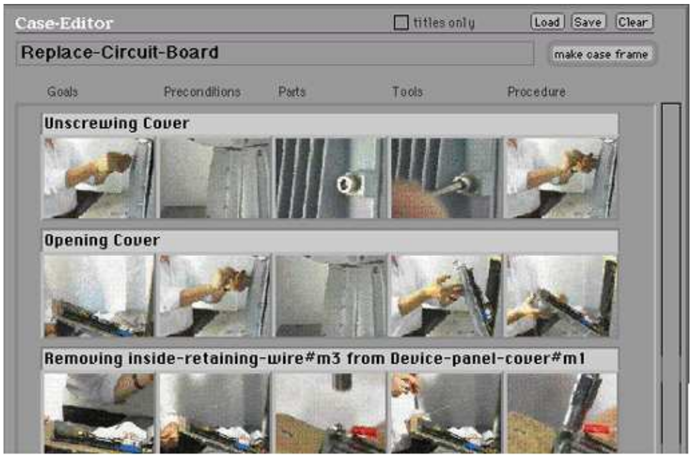
Learning by examples from maps

Suguru Ishizaki did a thesis on a programming by example system for geographic information systems. Alan Turransky connected his example-based system to produce rules accepted by Louis Weitzman's interactive automatic layout system.



Intelligent interpolation

Sketched examples played an important role in Russell Greenlee's layout-from-sketch system, Steve Librande's intelligent image interpolation system, and Karen Donoghue's gestural animation system. Case-based reasoning, another example oriented technique, was central to Colby's and Kanarick's work mentioned earlier, and Ron MacNeil's Tyro, which constructed new multimedia presentations based on previously presented examples.



Case-based reasoning for presentations

Muriel's legacy: AI and Design

This quick survey only scratches the surface of the breadth and scope of projects in the VLW that applied AI techniques to visual design problems. Part of Muriel Cooper's enduring legacy will be her view of visual design as a creative problem solving process, and the hope that we can model it and provide intelligent, dynamic tools to help designers in their work. Future work in AI and design will be much indebted to Muriel and to the work of her group.