Afterword:
The Design Studio of The Future

William J. Mitchell

Harvard University Graduate School of Design

Then
Things began to change in the mid-1940s, though architects hardly noticed. Scientists and engineers started to speculate that the new electronic technologies which had emerged in the wartime years would profoundly change the character of intellectual work. Vannevar Bush (1945) imagined a device called the Memex, which would function as a personal information server (figure 1). By the 1950s computers were becoming a commercial reality, and in 1956 Fortune magazine published a remarkably prescient depiction of a machine that we can now recognize as a computer-aided design workstation-complete with graphic input devices and a multi-window display showing different views of a three-dimensional object (figure 2). These wonderful machines were never built, much less put to any practical use, but they established a powerful idea.

In the early 1960s there was sufficient international interest to prompt the question, from Moscow Pravda (Sinyakov, 1962), "Can a machine create a design?" As if in reply, Steven Coons (1963) outlined the functions of a CAD system in considerable detail, and working prototypes began to appear. At MIT Ívan Sutherland (1963) built Sketchpad, and produced a memorable film sequence showing a designer (himself) manipulating an interactive graphic display with a lightpen. Architects could imagine themselves doing that, and a few became enthralled with the idea. Sketchpad-like systems with an architectural flavor were soon developed. At Cambridge University, for example, William Newman (1966) implemented one that allowed an architect to select and assemble elements from an industrialized component building system. Timothy Johnson (1963) extended the Sketchpad idea to three dimensions. By the end of the decade there were actually a few computer-aided design systems in practical use in architectural offices (Ray-Jones, 1968)
Figure 1 A 1945 artist's conception of a personal information server. (From *Life* magazine, September 1945)

Figure 2 A 1956 artist's conception of a CAD workstation. (From *Fortune* magazine, November 1956)
These first-generation CAD systems were driven by mainframe computers and had refreshed-vector displays—expensive technology by any standards. They were delicate devices that required special air-conditioned environments, represented major capital investments, needed specialized operators, and could only be cost-justified if they were used continuously and very intensively. They did not fit gracefully into design studios, and it is hard to imagine that designers could ever have been very comfortable with them.

A second generation of CAD systems arrived to mark the beginning of the 1970s. Systems of this generation were driven by 16-bit minicomputers (such as the DEC PDP-11) and usually had storage-tube displays. They were more compact, more robust, had a better cost-performance ratio, and provided the vehicle for emergence of the turnkey CAD system industry. Among the first minicomputer turnkey vendors to achieve substantial success with their products were Computervision, Applicon, Autotrol, and Calma. A few of the early products of this industry found their way into large U.S. architectural offices, but their effect on day-to-day design practice was fairly negligible. In Britain, large public-sector construction organizations, with repetitive building programs, provided a context for more effective use of this generation of CAD technology (Mitchell, 1977): among the specialized systems that were developed to serve these organizations were CEDAR (Property Services Agency), HARNESS (Department of Health and Social Security), OXSYS (Oxford Area Health Board), CARBS (Ciwyd County Council), and SSHA (Scottish Special Housing Association).

The turnkey CAD system industry began to boom when a third generation of CAD technology appeared at the beginning of the 1980s. New 32-bit supermini computers (such as the DEC Vax), running sophisticated operating systems, replaced the 16-bit minis. As memory costs dropped, high-resolution raster displays replaced storage tubes. CAD system software, in this environment, became extensive and complex—often running to hundreds of thousands of lines of code, and requiring many volumes of technical documentation. Strong men were needed to lift the huge diskpacks on which many of these systems were delivered. The systems were large, impressive-looking, and expensive—typically costing as much per seat per year as the salaries of two or three middle-level employees. This meant that they had to be used very intensively if they were to be profitable investments. Many large architecture and architecture/engineering firms installed them, and used them in multi-shift production drafting applications. Specialist operators were commonly employed, and computer operations were often kept organizationally and physically distinct from the design studios. A few schools of architecture managed to acquire third-generation systems, and pioneered their use in studio
teaching—but this was usually regarded as a peripheral and experimental activity, and there was little integration into the general curriculum.

Meanwhile, the first inexpensive personal computers appeared. Eight-bit personal computers first emerged in the mid-1970s, were popularized in the late seventies by vendors such as Apple, Radio Shack, and Commodore, and peaked in popularity in 1981/82. These machines were far too low-powered to perform any useful CAD functions, but they established the idea of personal computing and provided the foundation for growth of a large industry. They were displaced in the mid-1980s by a generation of sixteen-bit machines—most notably the IBM PC and its clones, and the Apple Macintosh—which had more speed and memory, could be equipped with hard disks, and could provide fairly acceptable graphics performance. The Macintosh also introduced to a broad public the mouse-and-windows style of interaction (which had been developed much earlier at the Xerox Palo Alto Research Center).

In everything except cost, early sixteen-bit personal computers were far inferior to 32-bit superminis as environments for implementation of CAD systems. But they were affordable by individuals and small organizations—in particular, the thousands of small architecture and engineering consulting firms that exist in the United States. Their success created, for the first time, the possibility of a mass-market for CAD systems. Entrepreneurs were quick to realize this. A fourth generation of simplified CAD systems, offering a subset of the functions of large third-generation systems, and performance limited by the capacities of sixteen-bit, single-user machines, appeared on the market priced in the one thousand to ten thousand dollar range. (By 1989 many capable CAD systems were priced at less than a thousand dollars per copy.) The Autocad system, particularly, gained wide and rapid acceptance: it was first marketed in 1982, and by 1989 more than two hundred thousand copies had been sold. Its use became commonplace in small architectural and engineering firms and in many schools of architecture: for a great many architects, engineers, drafting technicians, and students, it provided a first, relatively easy and inexpensive practical introduction to CAD.

Another effect of fourth-generation, personal computer CAD was to change the pattern of CAD software marketing, training in its use, and technical support. Vendors of earlier generations of systems had usually developed direct sales organizations, and were prepared to put a large amount of effort into discussing a particular client’s needs and tailoring a system to suit. They typically provided extensive user training, and sophisticated support to each installation. By contrast, personal computer CAD vendors sold simplified, standardized products through the mail, or through dealer networks, and relied on this simplification and standardization to minimize the need for installation, training, and support services. The software was delivered on
floppy disks in small shrink-wrapped packages, and users typically installed it themselves.

The negative effect of fourth-generation CAD was to establish a banal and simplistic conception of CAD functions and style of interaction in the minds of many architects. CAD systems were now seen, by most professionals and teachers of architecture, as the graphic equivalent of word processors—rather simple devices for input, editing, and production of graphic (rather than text) documents. The theoretical foundations of these systems (which had been established a quarter of a century before, and contained much that was problematic) remained mostly unexamined, and the wider possibilities were largely ignored.

As the flood of personal computer CAD software grew, yet another combination of technologies was emerging to provide the foundation for a fifth generation of CAD systems. Powerful graphic workstations that could be combined into networks began to appear. The first of these was the Three Rivers Perq—a sophisticated but expensive machine that came on the market in 1989, but never achieved widespread success. Apollo workstations appeared in 1980, and the first Suns in 1982. The workstation market grew very rapidly through the 80s as performance improved and costs dropped. As the end of the decade approached, the distinction between personal computers and workstations was becoming blurred as costs of some workstations dropped to personal computer levels, and as the speed and sophistication of some personal computers rose into the workstation range.

Networks provide an attractive environment for CAD because they combine the advantages of centralized timesharing systems with the advantages of the personal computer. Like a timesharing system they can provide access to centrally maintained files and software, communication between users, and the possibility of sharing use of expensive pieces of equipment such as large plotters. But, like personal computers, workstations can support a fast and fluid style of graphic interaction.

CAD system vendors from both ends of the industry reacted to the emergence of workstation network technology by adapting their products to it. Many of the large systems that had originally been developed for mainframes or minicomputers were rewritten to run on workstations, with some functions handled locally at the workstation and some at a central server. Some personal computer systems were upgraded to take advantage of the greater speed and capacity, more sophisticated operating systems, communications capabilities, and better graphics that workstations offered. And of course there were some ambitious new systems, such as IBM’s AES, designed specifically for workstation networks.
Now (at the GSD)
Since 1987 the Harvard Graduate School of Design has been engaged in incremental development of a CAD network to support teaching and research in architecture, landscape architecture, and urban design and planning. This network is designed to grow over time as CAD becomes increasingly integral to the design professions, to support a heterogeneous mix of technologies and products from different vendors, and to allow for smooth integration of new technologies as these emerge. Its organization and functions will evolve as computer, communications, and media technologies continue to develop, and as we gain experience with its use.

At this stage in the development of CAD technology, specialized CAD laboratories make as little sense as specialized pencil laboratories: it is no longer essential to confine machines to controlled environments, so we can make design tools available in the usual places for doing design work. (We should deliver the technology to designers where they need it, not force designers to go to the technology.) Accordingly, workstations are distributed throughout the GSD. Most of them are in the design studios. We expect gradual evolution of a studio environment in which a computer workstation is as normal a part of a student desk as a parallel rule or a drafting lamp.

Currently (1989) there is a Sun 386i server at the heart of the network, and connection is provided by Ethernet carried on coaxial cable. Workstations and personal computers in the network cover a broad spectrum: a Sun 4, Computervision Suns, Sun 386is, IBM RTs, PS/2s, and PCs, Macintosh ITs, and some older Macintosh Plus machines. The strategy is to provide a large number of inexpensive workstations to perform relatively simple tasks, a smaller number of more sophisticated and expensive workstations to perform more demanding tasks, and where necessary, highly specialized workstations to perform certain specific functions. There is, for example, a Macintosh II configured with a slide scanner, a flatbed scanner, and a digital film recorder, to perform image input and output functions, and there is another configured with devices for input and output of running video.

Our approach to providing application software is guided by the metaphor of the research library and the reading list rather that that of the standardized textbook. We avoid forcing everybody to approach a task in the same way, using the same constructs and techniques. Instead, we attempt to provide ranges of options, to develop in students the capacity to evaluate tools and technologies critically, and to make intelligent choices from among the available possibilities. (The ability to do this is a far more important component of professional competence than skill in application of some particular piece of software that may well be obsolete before the student
graduates.) Students may decide, for example, whether they want to invest substantial time and effort in learning Unix capabilities and commands, or whether they can accomplish what they want within the simpler Macintosh or DOS environments. If they want to manipulate some bitmapped images they may choose to work with a very simple paint system on a Macintosh Plus, intermediate-level systems like Pixelpaint or Photomac on a Macintosh II, or the sophisticated but demanding Artisan system on a Sun 386i. If they want to do some three-dimensional geometric modeling they may choose a simple sketch tool like Modelshop, a widely-used low-end system like Autocad, or a higher-end system like Computervision or AES.

Perhaps even more important than the possibility of choosing among different computer tools to perform the same task is the capability to combine tools freely in order to accomplish results that no single one could provide individually (figure 3). A designer might, for example, begin modeling a building with simple software on an inexpensive workstation, transfer the database to a more sophisticated modeler on a more powerful workstation for more detailed development, transfer again to a specialized rendering system to produce realistic shaded images, move the resulting bitmapped images to a paint and image processing system for retouching and blending with scanned images of the site, and finally send the images to a video

Figure 3 A 1989 conception of a CAD network, as implemented at the Harvard Graduate School of Design.
editing station for integration into a video presentation. Increasingly, software packages provide ranges of import and export formats and device drivers to support this, multi-format graphics translators such as are now available, and there is growing acceptance of standard exchange formats such as IGES, but we have also found that we must often devote extensive effort to writing file translators and device drivers needed to integrate a new device or piece of software into the network.

A complete functional overview of a well-developed network cannot easily be given in a single diagram: the web of interconnections between functional elements become too complex. But slices can be cut through, in various ways, to illustrate different aspects of function. Figure 3, for example, shows how traditional graphic media (drawings, slides, and photographs), bitmapped images, two-dimensional vector drawings, and three-dimensional geometric models are interrelated in the GSD network. The accompanying examples of recent student design work suggest something of what these interconnections mean. Figure 4 shows exploratory sketches made by drawing with pencil and marker on laser prints output from a drafting system. The sectional perspectives in figure 5 were produced by a high-end three-dimensional modeler. The studies of proposals in context shown in figure 6 were made by combining synthesized building images from three-dimensional modelers with scanned images of sites, then using paint and image processing operations to clean up the result. The student who produced the doorway designs shown in figure 7 first modeled basic geometry in Autocad, then performed simple cosine shading with Autoshade, next scanned slides to generate a palette of material textures, and finally used a paint system to make quick sketches of alternatives by brushing lines and textures over shaded images.
Figure 5. Sectional perspectives of projects produced using a three dimensional modeler, (Donald C. Cook, Leandro Madrazo, Daniel Tsai)
Figure 6 Studies of projects in context made by blending synthesized images with scanned images.
(Daniel Tsai, James Davidson, Nancy Cheng, Doo-Ho Sohn, Branko Kolarevic, Donald C. Cook)
Soon

Marvin Minsky (1985) has used the metaphor "society of mind" to suggest ways that complex intellectual processes can be carried out by collections of interacting intelligent agents. More narrowly, we can usefully employ "society of design" as a guiding metaphor for speculation about design environments of the immediate future. With this in view, I shall venture some predictions.

Designers and students of design will soon come to regard computer networks as indispensable utilities, and workstations will be as standard on their desks as telephones. These workstations will provide tools for accessing, manipulating, and disseminating information in a wide variety of forms: text files, numeric databases, geometric models, bitmapped images, sound bites, and video clips: after nearly half a century the Memex idea will become reality. Separations between different forms of information will disappear: sound effects might be associated with spaces in a geometric model of a building, sound notes with elements in a drawing, or video clips with technical information in a construction products database.

Modes of interaction will become richer and more various than they are now.

Only a few years ago interaction with a CAD system usually meant typing in command lines, in arcane syntax, to laboriously edit static, two-dimensional line drawings on dim storage tube displays. Now it is commonplace to manipulate three-dimensional models, which can be displayed shaded and in color, via much
more fluid and intuitive point-and-click interfaces. Real-time movement through complex shaded scenes and real-time parametric variation of complex three-dimensional objects (Danahy, 1989) will soon be standard. Stereoscopic displays and three-dimensional input devices such as the dataglove will allow escape from the picture plane where this seems desirable. (It isn’t always: simple, two-dimensional line drawings are often useful to designers precisely because they are so abstract and ambiguous.) There will be more input and output channels in simultaneous operation—speech input and output in parallel with graphics and text, for example.

Realistic rendering of details, spaces, and buildings in context will no longer be extremely expensive to generate—as they have been until very recently. Even low-end CAD systems now routinely provide cosine-shaded perspectives. Moderately priced workstations such as the Personal Iris have become sufficiently powerful to produce complex images with texture mapping and global illumination effects (Greenberg, 1989) in acceptable amounts of time for practical design purposes. This will not only enable more effective presentation of designs, but also allow designers to make extensive, routine use of realistic views in the close study of design alternatives.

The databases of architectural modelers will become more sophisticated. Where today’s systems represent buildings as collections of lines, surfaces, or closed solids organized into layers, future systems will provide for multiple simultaneous representations of components and subsystems. For example Eastman (1989) has discussed building modelers in which architectural objects have multiple descriptions or views: solid models, default views for plan and section, symbols with centerlines, and so on. As designers become more sophisticated about the organization of design information the simple idea of layering will seem increasingly inadequate: designers will want to associate properties with architectural objects in complex ways, to incorporate procedural elements in designs, to make use of default values and inheritance through use of constructs such as frames and object-oriented programming, and to select and recombine graphic information using relational databases, SQL commands, and so on. Stiny (1989) has suggested a more radical possibility: that multiple design descriptions might be constructed in multiple algebras which are combined in Cartesian products such that linked computations of different aspects of shape information (plan, elevation, section, solid), functional descriptions, assembly instructions, and so on are carried out in parallel.

Specialized network nodes will efficiently provide sophisticated services to large numbers of users. A high-speed graphics processor might, for example, do nothing but produce compute-intensive ray-traced or radiosity renderings from files of geometric data submitted through the network. Another node
(like a consulting firm) might deal with structural aspects of designs, another with mechanical aspects, and so on. The different nodes might communicate by simple file transfer, but there are some more interesting possibilities as well. Different nodes might, for example, run different knowledge-based systems that develop different aspects of a design in parallel, using a blackboard system or something similar to receive messages about the evolving work at other nodes and to post requests to other nodes for needed information (Fenves et al., 1988).

Electronic media and information servers integrated into networks will supplant traditional libraries in a broadening range of functions. It is no longer prohibitively expensive, for example, to convert architectural slide libraries to videodisc, and to attach videodisc players to workstations to provide immediate access right in the studio environment (Purcell 1989, Van Bakergem 1989). Video images have quality limitations, but these can be overcome by storing images in digital format in media such as CD ROM. As the technology for handling digital images matures (and it is doing so rapidly) it will be increasingly attractive to replace slide libraries with image servers that maintain image collections and provide access via any of the workstations in the network. (A slide carousel is, after all, just a slow, low-capacity, single-user disk storage system.) In this context there is no need to interpret the idea of an electronic 'slide' too narrowly: the collection might include not only bitmapped still images, but also video clips and sound bites.

Electronic "slides" do not just passively provide reference information: they are available for computer manipulation. For example satellite images can be processed to extract site information, videodisc movie-maps of sites can provide frames for combination with synthesized views of design proposals, and history slides can be processed to produce reconstructions (figure 8).

Collections of three-dimensional geometric models are also likely to become increasingly important. These might be disseminated on CD ROM, or from a server through a network—it doesn't much matter. Models that an architect might need include context models of urban environments, precedent models of important buildings, and extensive libraries of construction components and subsystems. A designer might scan these libraries to find interesting pieces, collect a "shopping cart" of them for current use, and instantiate them in a project model as required.

Eventually, networks will serve not only images and models, but also knowledge-bases for selecting pieces and putting them together to generate various types of designs (Eastman, 1989, Geilingh 1988, Mitchell, Liggett, and Tan 1989, Turner 1988). A designer might shop for relevant knowledge-bases—collections of facts and rules that seem to bear on the current situation—as for
Images or models, then use chosen ones as tools to develop the project. Much of the intellectual capital of a design organization may reside in its knowledge bases. Prominent designers may franchise themselves by making available knowledge bases that capture their characteristic ways of doing things. (A system for designing wooden decks has already been franchised to hardware stores, and systems for design of more complex structures cannot be far behind.)

**Conclusion**

Computer-aided design technology has passed through many metamorphoses in the three decades since it was first imagined, and it will pass through more before the century is out. Early CAD systems, inevitably, automated manual design functions in much the same quaintly literal way that a player piano automated the functions of the pianist’s fingers: sketching with a pencil on paper was replaced by sketching with a lightpen on a display surface (to be followed with careful ink drafting by a motor-driven pen). But even the earliest systems contained the seeds of far more radical ideas. Realization of these ideas will fundamentally alter the conditions of production of architecture, standards of architectural discourse and
judgement, and approaches to architectural education (Mitchell 1989)

Future CAD environments will bear as little resemblance to the first prototypes as modern music reproduction devices (like the portable CD players favored by joggers) bear to player pianos. As we discover unanticipated new needs and uses for them we will think it just as well-as thoughtful joggers must occasionally be grateful that they don't have player pianos strapped to their backs.

References


Purcell, Patrick. 1989. 'Light Table: An Interface to Visual Information Systems." This volume.


