CAD as a Social Process

William J Mitchell

School of Architecture & Planning
MIT

We can now look back over more than three decades of CAD research, software development, and deployment in practice. In that time, several guiding paradigms have emerged - then have eventually been superseded(or, at least, extended and augmented) by new ones. In this paper I briefly glance backward at successive paradigms, consider what each has contributed to our understanding of the subject, then go on to consider the latest - CAD as a network-supported, social process.

Keywords: geographically-distributed computing, virtual design studios

1 Convergence of computation and telecommunication

Back in the 1960s, it seemed particularly useful to think of designing as a form of problem solving. In their now-classic pioneering work, Newell and Simon had demonstrated that computers could solve a variety of intellectual problems (chess puzzles, theorem proving tasks, and the like) by searching spaces of candidate solutions to find ones that satisfied pre-specified requirements. This view fitted nicely with the batch processing computer environments that were commonplace at the time; you would input some set of requirements, let some algorithm crank away for a while, and expect the answer to the problem eventually to pop out. And it was also consistent with unreconstructed modernist ideology; formulate a sufficiently Now and detailed set of requirements to satisfy, and the architectural solution should follow more-or-less automatically, by application of some appropriate design procedure. So there were, for example, lots of attempts to write floor plan layout programs that would take lists of area and adjacency requirements as input and produce plans satisfying these requirements as output. Some clever algorithms were developed, and some of them worked quite well, but it soon became obvious that this approach would have limited payoff.

Then interactive computer graphics became increasingly feasible - a long technological development that passed through the eras of refreshed vector displays, storage tube terminals driven by minicomputers, and eventually inexpensive personal computers with sophisticated raster graphics. Ivan Sutherland's Sketchpad system early and dramatically introduced the idea that designing might be treated as a process of graphically editing a database. The initiative in the process (perhaps the "creativity") would remain safely in the hands of the human user, but the computer could play a useful role by applying a variety of algorithms to the database representing the design. You might assign to the machine tasks of formatting and plotting drawings, generating shaded perspective views and walk through animations, taking off quantities and calculating costs, performing structural and thermal analyses, producing facilities management reports, and so on. Research focused on creating building description databases that would support efficient editing and application of wide ranges of useful algorithms, and on developing algorithms for performing useful document production, visualization, and analysis tasks. This approach became a huge commercial success when AutoCad met the personal computer in the 1980s. It has been very useful, but it has always seemed focused mostly on the mechanical and mundane aspects of design - never really getting to the creative heart of the design process.

Knowledge-based design systems, which flourished most vigorously in the 1980s and early 1990s, attempted to overcome the limitations of both the problem-solving approach and the interactive graphics approach by capturing and applying something of what designers have in their heads. Inspired by early successes of knowledge-based systems in domains such as medical diagnosis, knowledge-based design systems used various formalisms (productions, arcane logical formulae, repertoires of cases, shape grammar rules, and the like) to encode facts and rules pertaining to the solution of particular sorts of design problems. The idea was a plausible one; successful designers do seem to draw on extensive stores of accumulated knowledge. It was encouragingly easy to produce demonstration systems that worked well in very narrow and limited domains. But creating knowledge bases to deal effectively with consequential, practical problems usually seemed to turn into a huge, expensive, and dauntingly difficult task.

What now? It seems to me that the most interesting new directions are suggested by the growing convergence of computation and telecommunication. This allows us to treat designing not just as a technical process (as CAD systems based on the problem-solving, interactive graphics, and knowledge-based paradigms have basically done)but also as a social process.

2 The virtual design studio

Why should this be important? Well, walk into any design office and look around. You will see drafting boards, computer workstations, telephones, and conference rooms. What are all the designers doing? Chances are they're mostly talking on the telephones, holding discussions over the work on the boards and on the screens, and having meetings with consultants and clients in the conference rooms. They are spending a fairly small part of their collective time and energy making marks on paper or editing displays, and a larger part of it in discussing, arguing, negotiating, forming consensus, trying out ideas and getting reactions, identifying and resolving conflicts, and reaching shared understandings an agreements. It's a person-to-person, social process -not just one of solving technical problems and producing documentation. That s what practical design is mostly about. And traditional CAD hardly helps with the all-important social aspect of it at all.

It's the same in design teaching. We ask students to produce design proposals in response to problem statements. But they don't do this in isolation. The discussions that take place in desk crits, intermediate reviews, and final juries are an absolutely crucial part of the process. It's in these discussions of student-generated propositions that most of the learning takes place.

Now consider this scenario. An architect in Singapore wants to discuss a construction detail with a consultant in Boston. At her workstation, she opens a shared CAD window; both she and the distant consultant can view and manipulate the design displayed there. She opens a video conferencing window as well, so that she can see and hear the consultant. Re has a corresponding video window at his end. They discuss and resolve the problem, much as they might by meeting face-to-face over a drawing-pencils in hand. This is a case of synchronous telecommunication; appropriate combination of CAD with digital video and networking has created a simple but useful virtual design studio.

The virtual desk crit is a variant on this. Prominent visiting design critics often cannot spend a lot of time with students in the studio, since they need to be back in their offices running their projects. Now, we can begin to augment this limited face-to-face contact through digital telecommunications. The idea is to place inexpensive video conferencing workstations in the studio and in a visiting critics distant office. Students develop their work in a CAD environment, and can discuss it with the critic by remotely sharing access to a CAD window and opening a video link.

Here's another one. Instead of sending a roll of drawings from Boston to Singapore to be marked up and returned with changes, an architect mounts them on a WWW site. The Singapore colleague surfs in and adds the necessary comments and suggestions. Next morning, the Boston architect accesses the site again, finds the new material, and goes on from there. Here, asynchronous multimedia telecommunication has effectively been used.

And here's something a bit more advanced (and not yet quite possible).An architect working on a 3D modeling system wants to select and specify some ceiling tiles for a project he's designing. He indicates the position of the tiles in the model, and invokes a software agent that crawls out on the Web. This agent visits servers, maintained by vendors, with product information about ceiling tiles. It finds a product to match the context and performance and price requirements, inserts the corresponding product information into the model, and generates an updated rendering showing how they will appear. If the architect indicates approval, the agent automatically places an order. Here, some of the negotiations and transactions involved in design and construction are performed by an agent that operates in an open-ended network environment. It's a logical extension of the basic ideas inherent in the World Wide Web.

Why is this sort of thing potentially important? Why not just co-locate all the members of a design and construction team, so that fast and convenient communication can take place among them? First, it's rarely feasible in practice; practical design projects of any magnitude are almost invariably carried out by multidisciplinary, geographically distributed teams. The architects may be located in one city, the client in another, the consultants someplace else, and the construction site at some distant location. And the construction team will typically involve fabricators and subcontractors from widely scattered locations.

Secondly, many design firms increasingly need to compete in international rather than local markets, so they need effective ways to export their services. If an American firm wants to do work in China, for example, telecommunications is immediately a major issue. In addition, competitiveness increasingly depends on the ability to aggregate needed expertise in the most flexible and efficient way. It's an advantage to pull into a design team the most expert consultants, the most skilled fabricators, the most capable managers, and so on - and you have a better chance of putting together the best team if you can draw on the resources of the whole world, not just those available locally. Finally, a geographically distributed operation can sometimes provide ways to reduce costs by tapping into the most advantageous labor markets. The Bangalore software industry is striking testimony to. this.

Since the growing needs are matched by emergence of technologies increasingly capable of serving them, development of geographically distributed virtual design studio environments is likely to be speedy.

The worldwide digital telecommunications infrastructure is rapidly developing into a system with the ubiquity and switching capabilities of the telephone system, the bandwidth and multimedia possibilities of television, and the virtually limitless storage capacity and processing power produced by the silicon chip. As CAD technology is integrated into this increasingly powerful new environment, we will find that we have not just as a collection of computer tools for performin~ certain design tasks more efficiently, an opportunity to reinvent design organizations in the most fundamental and sweeping ways.