DRAWING, SEEING, AND REASONING: THE ADDED VALUE OF COMPUTER AIDED ARCHITECTURAL DESIGN

AUTHORS
Mark D Gross
Sundance Lab for Computing in Design and Planning
University of Colorado
Boulder, Colorado 80309-0314 USA

ABSTRACT
Viewing computer aided design in the context of the history of tools and media in architectural design reminds us that tools have consequences on the way of working. Design involves seeing, reasoning, and drawing in an iterative and interactive process. These three activities provide a framework for addressing the question of ‘added value’ in architectural design. A number of recent projects at the Sundance Lab illustrate how computer aided design can support these various activities.
DRAWING, SEEING, AND REASONING: THE ADDED VALUE OF COMPUTER AIDED ARCHITECTURAL DESIGN

Design Tools and Media
As we contemplate the added value of computer aided architectural design it may be useful to consider the history of other tools and media in design. Certainly, computers and information technologies appear to be reshaping the discipline of architectural design. The practice of architectural design has changed over the centuries and a brief look at the history of design technologies and media may help us understand the significance of the changes that we are witnessing today.

Architects have employed various methods to organize the design of buildings. For example, throughout the history of architecture architects have used arithmetic, geometry, and grids to control the positions and dimensions of material and space elements of buildings. Each method constrains the possible arrangements of form. Arithmetic was the main design tool for classical architecture, in which the plans and elevations of buildings depended on simple ratios (e.g., 1:1, 1:2, 1:3) of dimensions (Hersey 1976). Geometry was the predominant method in architectural design for the designers of Gothic cathedrals (Gimpel 1980). Geometric design allowed buildings to have dimensions that were not simply additive: the intersection of two arcs, for example, locates a point that cannot be fixed arithmetically. Architects have used grids to control the positions and dimensions of built elements. The Romans used the grid to lay out their military camps; and many modern architects have used systems of grids to organize their designs.

Various physical media have also been used to support designing throughout the history of architectural design. For example, tracing paper, so common in architectural offices today, is a relatively recent addition to the toolbox. Drawing tools too—the pencil and the ink pen—have evolved over the centuries (Petroski 1992); for example, the often-used felt tip colored marking pen is a late twentieth century invention. The techniques used to make drawings—projective geometry, the construction of accurate two-dimensional representations of three-dimensional space, in isometric, axonometric, and perspective projection—are also relatively recent developments. Perspective drawing, developed in the Italian Renaissance as a painter’s tool by Brunelleschi and Alberti, proved to have profound effects on architectural design and more generally on scientific thinking (Ivins 1973). Perspective representations enabled architects to construct visually accurate drawings of a building, and it has been argued that the adoption of perspective as a design tool led to changes in the buildings that were designed.

Thus, computer aided architectural design is a development in the practice of making buildings and places that we can view in the context of other historical development. Like any tool, it implies ways of working, or what was called in the 1960’s and 1970’s ‘design methods’. The impact of computers and communication technologies on architectural has been so sudden and apparently great, that it seems at first difficult to grasp the question of “added
value”. Perhaps an examination of the activities in designing can offer a framework addressing this question.

The Seeing-Reasoning-Drawing Triad

It has been proposed that designing is an iterative process of seeing and acting, or as the late design researcher Donald Schön put it, a ‘see-move-see’ cycle (Schön 1992). That is, the designer articulates a proposition by making an external representation such as a drawing, studies the drawing, for example evaluating it with respect to design criteria or goals, and then makes another proposition as a result of the study or seeing. In other words, in this view three processes make up the designing: seeing, reasoning, and drawing. In computational terms, we might call these processes: input, processing, and output.

Conventional modelling and drafting software, for example, supports primarily the ‘Drawing’ process. Architects use this software to produce external representations of buildings for evaluation and presentation. Knowledge based tools such as expert systems, simulations of heating, lighting, and structural support, and building product models and object ontologies support primarily the ‘Reasoning’ process. So far, little software has been written that supports the ‘Seeing’ process, but that may well change as machine vision technologies improve.

The following sections describe projects carried out over the past few years at the Sundance Lab for Computing in Design and Planning, locating the projects with respect to Seeing, Reasoning, and Designing. Of course, some projects fit better than others into the scheme and some projects could fall into more than one category.

Seeing

Seeing imagery produced by computer has been limited in various ways by the technologies available. Early computer graphics were produced by teletype and line printer, with gradations of tone produced by overstruck characters. Drum and flatbed plotters produced line drawings in a limited number of colors, without variation in line weight or tone. Only recently, with the development of low cost inkjet printers, has acceptable hardcopy output become available. Screen imagery has likewise been limited. In the early days, CRT displays produced monochrome line drawings; raster displays have gradually increased in spatial and color resolution to where the image quality is now acceptable, though still limited in size. Immersive imaging (head mounted or CAVE displays) offers ways to go beyond the fixed screen size, although the application of these technologies in architectural design are still not mature. A remarkably effective low-cost alternative to immersive hardware has been panoramic imagery within a window, as offered for example by Quicktime VR, and similar freeware alternatives, which enable the construction of interactive 360 degree panorama pictures that can be viewed in a window. Virtual Reality Modeling Language (VRML) and its successors are a more computationally intensive, but effective way of viewing buildings and places.
Two recent projects, Ceren Virtual Archaeology project (Lewin 1997) and the Hagia Sophia Web Resource, as well as efforts described in B.J. Novitski’s book, “Rendering Real and Imagined Buildings” (Novitski 1998) illustrate the possibilities of enhancing the “seeing” process. In these projects, panoramic photographs or VRML models enable architects to explore an on-line representation of a place, real, historic, or imagined. The computer screen is a window onto a larger three-dimensional virtual world, and the visual representation of the world is linked with additional information. Jen Lewin and Mark Ehrhardt’s Ceren Virtual Archaeology site comprises a collection of multimedia information, including renderings of computer models of the excavated buildings. Ceren also offers field specimen data about the buildings and artifacts through a database that is linked to items shown in the panoramic renderings. Mark Ehrhardt’s Hagia Sophia Web Resource project (figure 1) links close-up photographs and interpretive text about the building’s construction and history with the panoramic imagery. In both these examples, the computer presents images of architecture that are enhanced with additional layers of information beyond the visual appearance of the place.

![Figure 1: The Hagia Sophia Web Resource links historical information to panoramic pictures. Purely visual imagery is enhanced with overlays of integrated and linked information.](image)

**Reasoning**

Many efforts have been made in computer science and artificial intelligence research to construct computer programs that reason. Some of these, such as the Deep Blue chess program, now perform as well or better than people, within a limited and well-defined domain. A large array of software techniques is available to researchers who are trying to automate or augment human performance in design reasoning tasks. The challenge remains for design researchers to articulate architectural design reasoning with sufficient clarity that it can be coded.
My own CoDraw program and subsequently Construction Kit Builder project both explored the idea of design as a process of making and following a system of rules (Gross 1992, Gross 1996). In both these programs, the designer specifies constraints, or rules, that govern the arrangement of design elements. The CAD program is then responsible for seeing that the design obeys these rules, as the designer adds to and modifies the design. Both CoDraw and CKB support both rules about relative positions of elements, as well as rules that govern the placement of elements with respect to a grid.

Ellen Do’s IsoVist program is an example of support for a different kind of design reasoning, the evaluation or analysis of a design with respect to performance criteria (Do and Gross 1997). IsoVist offers an analysis of the visual field of a floor plan. The designer locates a standpoint in the floorplan, and the IsoVist program calculates the area visible from that location and displays that area as a shaded region in the plan.

Ellen Do’s IsoVist program is an example of support for a different kind of design reasoning, the evaluation or analysis of a design with respect to performance criteria (Do and Gross 1997). IsoVist offers an analysis of the visual field of a floor plan. The designer locates a standpoint in the floorplan, and the IsoVist program calculates the area visible from that location and displays that area as a shaded region in the plan.

Figure 2: The IsoVist program calculates the viewshed from a standpoint in a floorplan; simulation programs can help designers evaluate the performance of building designs.

Drawing

Until recently, the production of drawings was done by hand, using pencil or pen on paper; hand-work is now augmented by computer aided drafting and modeling. Many of the drawing and drafting programs are quite limited in their expressive ability; they allow only hard-line drawing using geometric primitives; and they depend on a structured menu human-computer interface. They do not substitute for freehand exploratory drawing.

Our ‘Back of an Envelope / Electronic Cocktail Napkin’ project is based on the observation that designers draw diagrams to explore concepts during early design (Gross 1996). Therefore, the computer should be able to recognize and interpret designers’ drawings. Once the computer can participate in the graphic dialogue, it may be able to offer advice, simulation, or other activities that support design reasoning. At the core of the project is a recognizer for symbols drawn freehand and a parser for configurations made up of these
symbols. Other layers of the project are built around that core. We’ve built “sketchy” interfaces to visual databases, simulation programs, and other applications. For example, we’ve built a sketch interface to the IsoVist program described above, and a scheme for indexing URLs in the Web using diagrams. Ellen Do's “Right Tool at the Right Time” manager attempts to guess what the designer is doing by watching the drawing, and based on its guess, it offers what may be an appropriate knowledge tool. For example, if the designer is drawing sun rays, it offers a case library of daylighting solutions (Do 1998).

![Image of a drawing with a sketch interface]

Figure 3: The Back of an Envelope / Electronic Cocktail Napkin project accesses knowledge-based design tools by freehand drawing.

Design is team work, and often several designers, engineers, and other concerned parties work together reviewing and making annotations on a drawing. In architectural teaching, too, the desk critique is a common way that teachers interact with students, drawing on the student's design work to make suggestions. Dongqiu Qian’s NetDraw is a Java based collaborative drawing program that supports collaborative drawing over the Internet. It enables several designers to share a drawing workspace, drawing on top of an overlay and making graphical gestures to point out what they are talking about in a linked text chat application. It goes beyond conventional whiteboard and conferencing applications by providing support for synchronization and concurrency control, grouping and simple graphical constraints on drawing objects, and ephemeral ‘gesturing’ that disappears within 30 seconds after it is made. NetDraw also communicates with AutoCAD and our Back of the Envelope program. We’ve designed it to run on mobile platforms, such as palmtops and Personal Digital Assistants.
Finally, Thomas Jung’s Immersive Redlining project (figure 5) is designed, like NetDraw, to support collaborative design among several designers, or to support critiquing and comment among a design team, including participation in the design process by clients. Our Redliner tool allows several participants to browse a stored VRML model and leave colored annotation markers in the model, linked to text comments. Viewers who browse the design model later can see and these annotations. The designer can also prepare alternatives to certain parts of the design, allowing the viewer to modify the position, color, or material of a building element using a simple palette of controls.

Discussion
The three-part framework, seeing, reasoning, and drawing, helps organize the methods and tools of computer aided architectural design. To be sure, it is not immediately obvious where to place in this simple framework other important design activities such as communicating with colleagues, or looking up standard dimensions, or creative imagination. These questions are arguable.
Nevertheless, the framework offers a way to consider where in the process of design various computer aided architectural design tools add value.

Acknowledgements
The Back of the Envelope and NetDraw research was supported in part by the National Science Foundation under Grant No. IIS-98-19856. The views and findings contained in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional support has been provided by the University of Colorado President’s Teaching with Technology initiative (the Ceren project) and the Chancellor of the University of Colorado at Denver (Hagia Sophia). Thanks to Ellen Yi-Luen Do, Jimmy Davidson, Handan Demirkaya, Mark Ehrhardt, Thomas Jung, Jennifer Lewin, Sinan Ozgen, Eric Schweikardt, Payson Sheets, and others who collaborated on these projects.

References