Conceptual Design as HyperSketching

Theory and Java Prototype

Raymond McCall, Ekaterini Vlahos and Joshua Zabel

Sundance Group for Computing in Design and Planning

College of Architecture and Planning, University of Colorado

Key words: sketching, hypertext, hypermedia, conceptual design

Abstract: Hand-done design drawing still has a several advantages over current, CAD-based approaches to generating form, especially in the early stages of design. One advantage is the indeterminacy of hand drawing--i.e. its abstractness and ambiguity. Another is a non-destructive drawing process, where new drawings are created without modifying old ones. A third is designers' creation of large collections of inter-related drawings--i.e. graphical hyperdocuments. A fourth is the unobtrusive character of conventional drawing tools. These advantages might be taken as reasons for continuing to do early design on paper, but they also suggest ways in which CAD might be improved. We have created software prototypes that incorporate these features into a new type of CAD based on sketching with electronic pens on LCD tablets. One prototype, which we call HyperSketch II, simulates tracing paper in the sense that it enables the user to trace over previous drawings and to build stacks of traced over drawings. It also enables the user to create a hypermedia network in which the nodes are sketches and the links represent various relationships between sketches.

1. PROBLEM

CAD is increasingly used by architects for design, though it generally finds much greater use in the later stages of projects, especially in *design development*. In the conceptual and early schematic stages of design, however, we still see many architects preferring to design by sketching with pencil and pen on paper.

Some people view the persistence of use of hand-done drawing in early design as merely a "generational thing"; they believe that when today's computer-literate children grow up they will design exclusively using tools like current CAD. We, on the other hand, believe this persistence reveals certain fundamental advantages of hand-done design drawing over current CAD. We do not, however, see this as an either-or choice between the computer and hand-done drawing. Instead, we argue that the advantageous features of traditional design drawing can and should be integrated into CAD software. In this article we describe software prototypes that move toward this goal by supporting sketching in design. Unlike work by most other researchers in this area, our emphasis is on providing support for the production, management and retrieval of large numbers of design sketches.

1.1 Differences that make a difference

Even a cursory examination of hand-done design drawings shows that they are strikingly different from the "drawings" or models created with CAD. In this paper, we take as our starting point the possibility that these obvious differences might represent important advantages of hand drawing for early design. In the following we describe a number of such differences and give our interpretation of their significance for CAD.

Difference 1: Hand-done design drawing is indeterminate in varying degrees, but current CAD models are highly determinate. By this we mean that hand-done design drawing is typically approximate, abstract, vague or ambiguous to some degree. It uses thick, wobbly or multiple lines; shapes are often rounded, lines only approximately parallel. CAD models, by contrast, use hard-line drawing and are typically based on precise dimensions and angles. Figure 1 shows an example of a hand-done design drawing.

It seems clear that the placement and shape of the lines in this example is only approximate and that this lack of precision is intentional and not the result of carelessness or incompetence. Conceptual sketches by the most famous architects reinforce this interpretation. As we see it, conceptual drawings such as the one shown here are meant to convey only approximate shape and location. Issues of precise dimensions and details are deliberately avoided and the "sketchy" appearance of such drawings is meant to convey that fact.

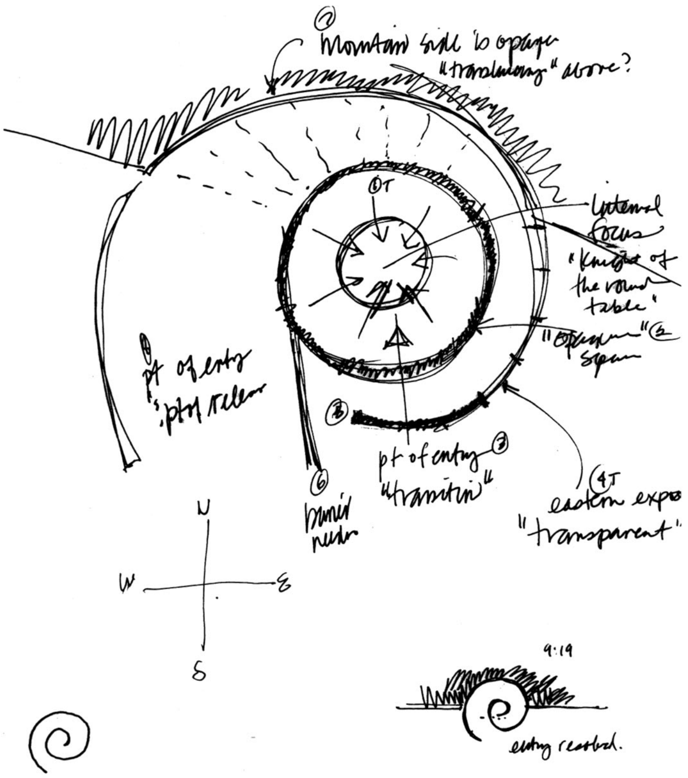

Figure 1. A conceptual sketch of a building with two "satellite" sketches in the lower corners.

That the indeterminacy of hand-done drawing is deliberate is further suggested by the fact that its degree varies systematically during design, generally becoming less indeterminate as design progresses. By being indeterminate, hand-done design focuses on larger issues of design while temporarily ignoring the many detailed issues that would arise in determinate drawing. Indeterminate drawing thus enables designers to use a *divide-and-conquer strategy* for attacking the complexity of architectural design. In other words, the sketchiness of the drawing is a means for decomposing the problem, for ignoring some issues to concentrate on others. In particular, issues dealing with precise location and shape are not yet being dealt with by the designer; they are a distraction at this point in the design process. They are likely to be tackled only after issues of general form and layout have been dealt with.

The implication here is that CAD—at least in its present form—does not provide as much opportunity as sketching for ignoring detailed issues. The hard-line character of CAD is a constant invitation to the designer to become distracted with such detailed issues as wall thickness, window dimensions and exact proportions of rooms. It also gives the viewer an unintended impression of precision.

Difference 2: Hand-done design is based on non-destructive drawing, but all current CAD is based on editing, an inherently destructive process. Once the initial version of a design solution—e.g., a model of a building—has been created with current CAD, creating the subsequent versions is accomplished by editing one version of this model to create the next version. This is a destructive process in that old states of the solution are destroyed—i.e. modified—to create new ones. Hand-done design, however, seldom involves modifying old drawings to create new ones. Instead, designers create new drawings from scratch or by tracing over previous ones. Destructive drawing, i.e., by erasure, generally has little or no significant role in early design.

Typically, the hand-done design drawings in a given project are highly heterogeneous. Different drawings represent many different levels of detail and different aspects of a building—including function, structure, appearance, circulation, lighting, user interactions, views, relation to site and surrounding context. Drawings show and compare alternatives at many different levels of aggregation and abstraction. As a consequence, most are not recognizable as being—even conceptually—edited versions of other drawings. Paper and pencil are thus much more than a "poor man's graphical editor," as is often supposed.

Especially important is the fact that use of non-destructive drawing preserves an *episodic history* of the project—i.e., a history in which graphical actions, such as the drawing of individual lines, are grouped into *episodes* corresponding to individual drawings and groups of drawings. Often a large sheet of paper is used as a way of grouping related drawings into still larger episodes—as shown in Figure 2. The episodic history of a project facilitates backtracking. It also enables evaluation of the design solution by examining its version history.

An editing process creates no episodic history. There is a history kept by the system to support *undo* functionality, but this is too fine-grained—at the level of individual operations on lines, shapes and groups of shapes—to be useful. Useful, high-level, cognitive groupings of such actions—such as drawings—are nowhere in evidence. CAD users often keep their own episodic history by saving intermediate stages in the drawing process in files and folders with names that they think up, but desktop CAD systems provide no explicit support for this.

Difference 3: Hand-done design creates a large collection of interrelated drawings but CAD creates only a single model. A large collections of drawings. Non-destructive drawing, by definition, produces multiple drawings, but this by itself does tell us how many drawings. Our studies suggest that a large number of drawings is typical. We observed one architect who created more than a thousand sketches in a two-week period. Many drawings by famous architects—such as the one shown in Figure 2—suggest the rapid production of many sketches during the conceptual design. Figure 3 shows one study of a professional designer who created 58 drawings in the first eight hours of a project. And in more that a hundred videotaped observations of students given a one-hour "sketch problem," we see them typically produce 6 to 25 drawings in that hour.

Current CAD, of course, is designed to support repeated editing of a single model. As mentioned above, designers can save multiple versions of their models in special files and folders that they create and name. But this approach is impractical when the number of drawings/versions goes into the hundreds—much less the thousands. Nor do current desktop CAD systems have either the database or version-control capabilities needed to manage drawings in such quantities. Furthermore, few CAD systems support creation of building representations in a minute or two, a regular occurrence in our studies of student design.

Interrelated drawings. In hand-done design, there are important relationships between drawings. Tracing is one indicator of such relationships. Some drawings represent alternatives to designs in other drawings. Some are done to resolve design problems that arise in doing other drawings; often the former are done in smaller scale on the side of the latter drawings—what we call *satellite drawings.* All of these represent crucial relationships between drawings. The groupings in the episodic history also represent relationships among drawings. In addition, design drawings are commonly annotated with related text, arrows, and numerical information. Thus—in addition to small “study” sketches—related diagrams, notes, tables and calculations are often found as "satellites" on the sides of larger drawings.

Some relationships among drawings are due to their sharing a common theme or exploring a common issue. Often such drawings are grouped on a single sheet of paper to show this relatedness—as for example in Figure 2.

Most relationships between drawings, however, are not explicitly documented. They are in the heads of designers during design. To identify such relationships we need to get information from designers while it is still fresh in their minds.

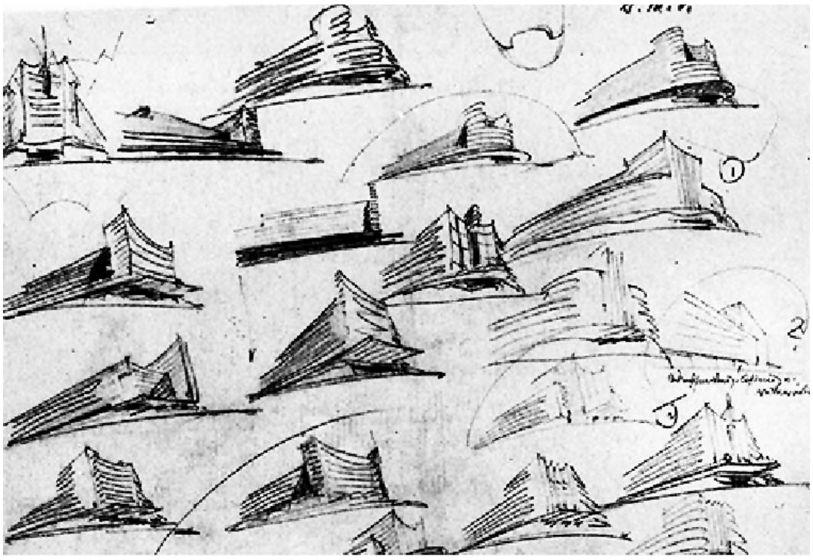

Figure 2. Sketches with a common theme on a single sheet of paper, by Erich Mendelsohn.

Figure 3 shows the results of an interview with a professional designer about the first eight hours of work on a project involving the redesign of a given house. The circles on the left edge of the figure represent the 58 individual drawings done in the eight-hour period. These are shown in the order in which the drawings were created, with the last at the bottom. (The top-most drawing is actually a site-plus-floor plan of the previous design of the house.) Note that there are a number of links directly between the individual drawings. For simplicity, we have represented all these links the same way. In reality there were six basic types of inter-drawing links that the designer felt it was important to distinguish. These included one drawing being 1) traced from (over) another drawing, 2) a further development of the design in another drawing, 3) a more abstract representation of the content of another drawing, 4) an alternative to the design in another drawing, 5) a part of another drawing and 6) a re-integration of a part into a larger whole. Some relationships combined several basic relationships—e.g., one drawing being a development of a part of another.

The small squares immediately to the right of the circles in Figure 3 represent sheets of paper. In this project it turned out that in every case where several drawings appeared on a single sheet of paper, the designer was consciously using the sheet to group related drawings. The larger squares further to the right represent major themes that group sheets of drawings. The large circle on the right hand side represents the project as a whole.

The software needed to display and navigate through linked collections of information is called *hypertext* or *hypermedia*. Current CAD systems have not been designed with the hypermedia and database capabilities needed to represent the relationships between drawings and to manage extensive networks of drawings. If we were to extend these systems to deal with large collections of drawings—as we are proposing—it would be important to record the significant relationships between these drawings. Without such relationships it would be difficult to retrieve and make sense of collections of hundreds—much less thousands—of drawings. And such collections appear from our studies to be common in architectural projects.

2. RELATED AND PREVIOUS WORK ON COMPUTER-AIDED DESIGN SKETCHING

2.1 Related work

A number of researchers have created software to support sketching in design. These include Gross' extensive work with the "Electronic Cocktail Napkin" (Gross 1996), and the work of Tolba, Dorsey and McMillan (Tolba, Dorsey et al. 1999). Sketching has also been a recent theme at SIGGRAPH—e.g., (Ohen and Zelnik 1999) and (Igarashi, Matsuoka, et al. 1999). Some research has focused on sketch recognition—e.g., (Do and Gross 1995) and (Herot 1976); some on sketching as an interface for creating 3D models—e.g., (Zeleznik, Herndon et al. 1996) and (Schweikardt and Gross 1998).

Almost all of the attention of researchers has been on individual drawings. Relatively little has been written about *collections* of drawings and how they function in design. To our knowledge, no one other than ourselves has written about the way in which design drawings are implicitly or explicitly linked together to form a complex, graphical hyperdocument.

2.2 Our previous work

Since the 1970's we have used hypermedia in design (McCall 1979). In the mid-1980's we coupled hypermedia with CAD (McCall, Fischer, et al. 1989). Following this, we created PHIDIAS, a software system based on a concept we call *HyperCAD* (McCall et al. 1990)(McCall, Bennett, et al. 1994). HyperCAD integrates CAD graphics, knowledge-based computation

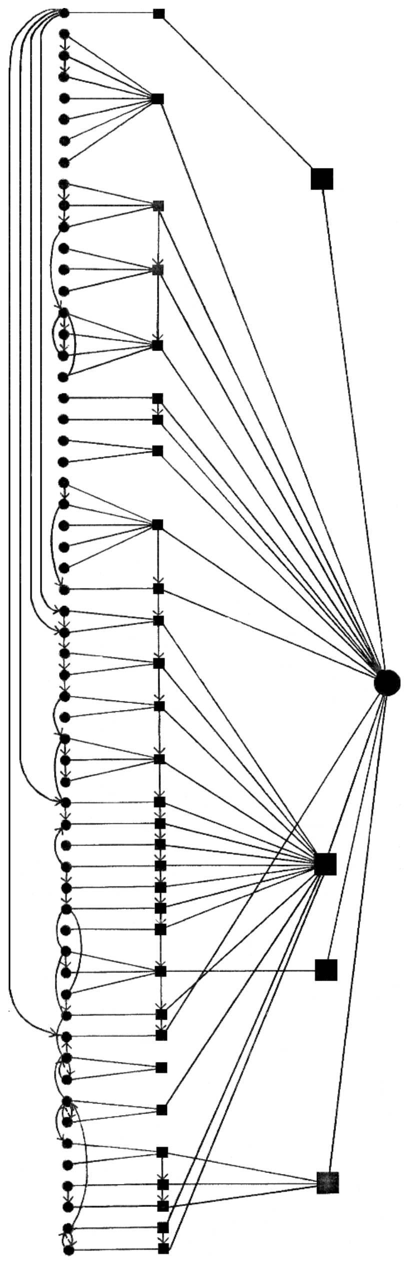

Figure 3. Graph of relationships among the 58 drawings done in the first eight hours of a house design project. All the relationships were explicitly identified by the designer as being important for an understanding of this collection of drawings. This graph is simplified in that different types of relationships are shown with the same type of arrows and without labels.

and hypermedia navigation using a unified computational substrate that manages databases of linked nodes.

In the 1990s we used HyperCAD to support of architectural sketching. The first results were 1) the prototype we call HyperSketch I and 2) an integration of hypersketching into PHIDIAS (McCall, Johnson, et al. 1996).

HyperSketch I was meant to do little more than simulate tracing paper on computer using a pen-based input on a touch sensitive LCD screen. It simulated the creation of multiple stacks of tracing paper but contained only the simplest drawing capabilities—i.e., one line thickness and no color—and very limited linking capabilities between drawings—i.e., primarily the linking of drawings based on their order in a stack of tracing paper. Tests of the system with computer-phobic architects (who were still in good supply the early 1990s) showed that it could immediately be used by designers with no CAD experience. But the HyperSketch I prototype did not attempt to show the advantages of using a computer for sketching.

We thought the most basic advantage of computer support would be the ability to manage large collections of linked drawings. In the mid-1990s, the best system at our disposal for managing large collections of linked data was the PHIDIAS system, whose hypermedia database engine we had been developing since 1981. We therefore integrated hypersketching capabilities into PHIDIAS (McCall, Johnson et al. 1997).

PHIDIAS was originally conceived as a stand-alone, desktop system. With the explosive spread of the World Wide Web we became convinced that the future of PHIDIAS' approach to hypermedia was to merge it with the Web. Online design offers crucial new opportunities for information access and collaboration, and these were precisely the goals that HyperCAD had originally been devised to achieve.

Initially we developed a prototype of PHIDIAS that could work over the Web (McCall 1999), but limitations of this prototype and the availability of new, software tools convinced us to scrap PHIDIAS and design a new, Internet-based HyperCAD architecture from scratch. This became the basis for the HyperSketching prototype described here.

3. THE HYPERSKETCH II PROTOTYPE

With the HyperSketch II system, as with its predecessors, designers create form by sketching with electronic pens on tablets. These system, however, is intended to go beyond pencil and paper alone by using the Internet to provide designers with "anytime, any place, any platform" access to both the software and the data management resources needed to create and retrieve large collections of interrelated sketches.

To provide access to sketching software and databases of sketches over the Internet, we opted for a three-tiered system architecture of the type has become standard and widespread in net-centric computing. The three tiers are as follows: 1) a client that runs on any Internet-enabled computer, 2) server-side software that combines a Web server with software for dealing with linked drawings, and 3) a back-end database management system. The middle tier mediates all communication between the client and the back end in such a way that the designer appears to have use of a database server that manages linked drawings—something that off-the-shelf database systems do not otherwise do. In reality, all the link management that connects the different drawings is accomplished by software in the middle tier. The linking in our case is hypermedia-type linking with links that are both labelled and typed and that are represented as first-class objects in the system. These links are also stored in the database. Such links can be followed in both directions in our HyperSketch II prototype. In other words, the linking mechanism is of the same basic type as that which provided the basis for PHIDIAS over the past decade of our research on that system.

For implementing the client, Java seemed an obvious choice. It combines platform independence with sophisticated Internet and interface functionality. Figure 4 shows the client in use. Note that the structure of drawings is shown on the right-hand side in the form of a *tree control*—of the same basic type seen in the Windows Explorer. This sort of interface is known in hypertext research as a *structure browser* and is generally regarded as an valuable aid to navigation in linked collections of information. It might seem at first that a tree is too limited to represent the sort of structure we saw in Figure 3, but trees—e.g., outlines—can represent arbitrary graph structures if any node is allowed to appear at more than one place in the tree. In addition to the tree representation, our client provides a thumbnail view of all sketches.

When the client starts, it provides the option of starting a new project or opening an existing one. If the user chooses the latter, the structure of the whole project is downloaded immediately; the individual drawings are only downloaded when displayed. This appears to be the only practical approach if hundreds of drawings—or more—are to be managed in a project.

We also opted for Java on the server. In particular, we used *servlets* to implement the software that mediates between client and database. Servlets run faster than CGI and are reputedly both more reliable and easier to manage. A servlet on this middle tier contains all the functionality for link management that makes the back-end database appear to the client like a hypermedia database—or *hyperbase*. In our prototype, the link-management servlet communicates with the back-end database using the JDBC (Java DataBase Connectivity) API (Application Programming Interface).

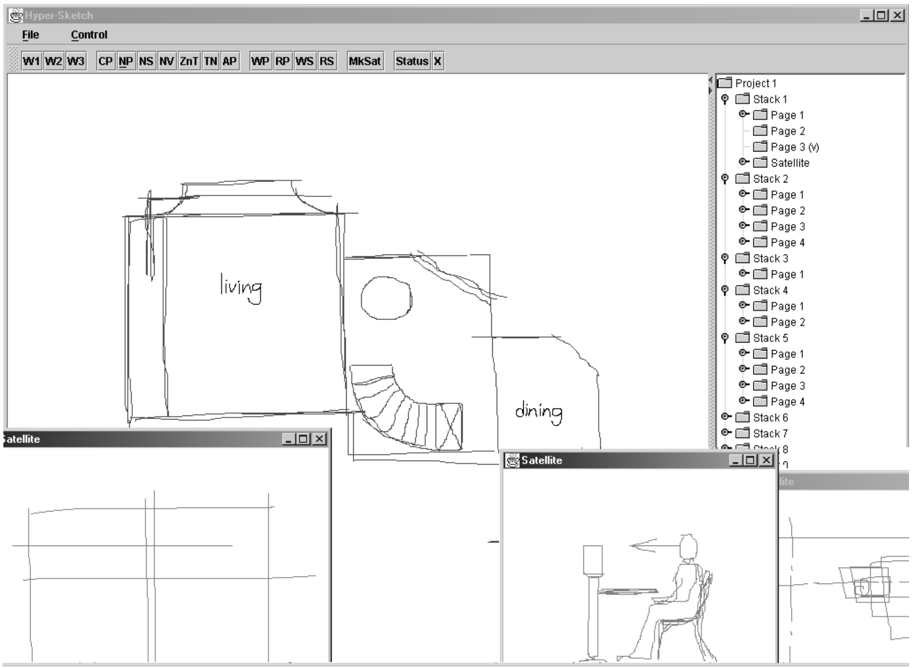

Figure 4. Screen image of the HyperSketch II prototype in use. Note the use (at the bottom of the screen) of separate windows for "satellite" drawings.

4. CONCLUSION AND FUTURE WORK

We have developed a new prototype, called HyperSketch II, that is aimed at showing how the Internet can aid design by enabling the creation, storage and retrieval of large collections of interrelated sketches from any Internet-enabled computer in the world. Our system is based on what we see as important advantages that hand-done drawing has for design, but we see our system's functionality as complementary to, rather than competitive with, conventional CAD. Hand-done design drawing and current CAD both have their strengths and weaknesses. Future systems can better aid design by providing integrated support for both.

It might seem that some of our functionality for creating multiple sketches could be achieved through clever use of existing, off-the-shelf software. But we suggest that such an approach is inelegant to the point of being impractical. As one software engineer we know is fond of saying, "Try it; you'll dislike it." It is precisely the business of CAD researchers to devise explicit and well-designed support for design processes and not to be content with half-baked solutions that use off-the-shelf systems in ways for which they were not designed and for which they are ill suited.

Our system currently has been tested on desktop computers only. Soon we hope to use it on a variety of wireless devices, including tablet computers. This system enables designers to enjoy the benefits of sketching for the conceptual stages of design while also taking advantage of hypermedia functionality to manage large collections of related sketches.

Much remains to be done to exploit the full potential of networked computers for support of conceptual design sketching. Our future efforts will address the possibilities of using sketch recognition for informing designers and bringing knowledge-based computational aids into the earliest stages of design. We also hope to explore the use of sketching in collaboration. We plan to investigation additional navigational and retrieval aids. And, of course, we plan to build far more sophisticated drawing tools into future prototypes.

5. REFERENCES

Cohen, J.M., L. Markosian, R. C. Zeleznik, J.F. Hughes and R. Barzel, 1999, "An Interface for Sketching 3D Curves," ACM Symposium on Interactive 3D Graphics, pp. 17-22 (April 1999). ACM SIGGRAPH. Edited by Jessica Hodgins and James D. Foley. ISBN 1-58113-082-1 .

Do, E. and M.D. Gross, 1995, "Finding visual references by sketching," Proceedings, Association for Computer Aided Design in Architecture, 1995 National Conference, Seattle, 35-52.

Gross, M. D., 1996, "The Electronic Cocktail Napkin - computer support for working with diagrams," Design Studies. 17(1), 53-70.

Herot, C. F., 1976, "Graphical Input through Machine Recognition of Sketches.," in *SIGGRAPH '76 Conference Proceedings*, pages 97-102, July.

Igarashi, T, S. Matsuoka, and H. Tanaka, "Teddy: A Sketching interface for 3D freeform design," *SIGGRAPH 99*, 409-416.

McCall, R. "A Web-centric CAD System for Collaborative Design", in *Proceedings of the Eighth International Conference on Computer Aided Architectural Design Futures*, Atlanta, 7-8 June 1999, pp. 6.

McCall, R; E. Johnson, and M. Smith, "Design as Creation of a Graphical Hyperdocument," *CAAD Future '97 Conference*, R. Hunge (ed.), Kluwer Academic Publishers, 1997, pp. 849-854.

McCall, R., P. Bennett, and E. Johnson, "An Overview of the PHIDIAS II HyperCAD System," in *Reconnecting: ACADIA '94 (Proceedings of the 1994 Conference of The Association for Computer Aided Design in Architecture)*, A. Harfman (ed.), Association for Computer Aided Design in Architecture, 1994, pp. 63-74.

McCall, R.; P. Bennett, P. d'Oronzio, J. Ostwald, F. Shipman, and N. Wallace "PHIDIAS: Integrating CAD Graphics into Dynamic Hypertext," in *Hypertext: Concepts, Systems* *and Applications,* Rizk, A.; Streitz, N. and Andre, J. (eds.), Cambridge University Press, 1990, pp. 152-165.

McCall, R.; A. Morch, and G. Fischer, "Supporting Reflection-in-Action in the Janus Design Environment" in *The Electronic Design Studio*, Mitchell, w.; McCullough, M. and Purcell, P. (eds.), MIT Press, 1990, pp. 247-259.

McCall, R., *On the Structure and Use of Systems in Design*, Dissertation, University of California, Berkeley, 1978, University Microfilms, 1979.

Schweikardt, E., and M.D. Gross, 1998, "Digital Clay: Deriving Digital Models from Freehand Sketches", *in Digital Design Studios: Do Computers Make A Difference? ACADIA 98*, T. Seebohm and S. V. Wyk, eds., ACADIA: Association for Computer-Aided Design in Architecture, Quebec City, Canada, 202-211

Tolba, O., J. Dorsey, and L. McMillan, "Sketching with Projective 2D Strokes, *UIST '99: Proceedings of the 12th Annual ACM Symposium on User Interface Software & Technology*, Asheville, North Carolina, November 7-10, 1999, CHI Letters, Vol 1, Issue 1, pp. 149-157.

Zeleznik, R.C., K.P. Herndon and J.F. Hughes. SKETCH: An Interface for Sketching 3D Scenes. In *SIGGRAPH '96 Conference Proceedings*, pages 163-170, August 1996.