

THINKING WITH DIAGRAMS IN ARCHITECTURAL DESIGN

Ellen Yi-Luen Do[†] & Mark D. Gross

Design Machine Group
College of Architecture and Urban Planning
University of Washington, Seattle

Published in Artificial Intelligence Review 15: 135-149, 2001
2001 Kluwer Academic Publishers. Printed in the Netherlands.

Abstract

The paper discusses the use of freehand diagrams in architectural design. It examines the roles of diagrams in various contexts: pedagogical books, design studies, designers' introspective accounts and empirical studies of drawing in design. It offers several examples of thinking with diagrams in design and concludes with a discussion of the requirements for computational support for the diagrams in design thinking.

Keywords

diagram, sketch, architectural design, protocol analysis, computational support

1. Introduction

Diagrams are essential representations for thinking, problem solving, and communication in the design disciplines, in particular those concerned with making physical form: mechanical and civil engineering, graphic design, and architecture and physical planning. In architecture, drawings are the primary form of representation; they carry a design from conception to construction. Except for physical models (which can be considered a kind of three-dimensional drawing) all external design representations in architecture are drawings.

Architectural design is ultimately about the configurations, connections, shape, and orientations of physical forms. Even the most abstract design diagrams are early efforts to explore and resolve spatial layout concerns. Architectural diagrams represent not only physical elements, but also forces and flows (e.g., forces of sun and wind and flows of people and materials). Thus arrows, lines, and other symbolic representations of forces and flows appear in architectural diagrams conveying spatial characteristics such as magnitude and direction. In the early phases of designing, architects draw diagrams and sketches to develop, explore, and communicate ideas and solutions. Design drawing, an iterative and interactive act, involves recording ideas, recognizing functions, and finding new forms and adapting them into the design. Thus drawing is not only a vehicle for communication with others; it helps designers see and understand the forms they work with (Edwards 1979).

Thinking with diagrams in architectural design has much in common with thinking with diagrams in other disciplines. The graphical elements and spatial relations of the diagram map to elements and relations in the domain and the spatial representation of the design offers insights and inferences that would be more difficult to see and work with in other representations. In some respects, however, diagrams function differently in architectural design thinking than in other domains. Architectural diagrams employ a full range of graphical indicators: They use topology, shape, size, position, and direction; whereas diagrams in other domains typically employ only one or two of these characteristics. For example, electronic circuit diagrams use only shape and topology to convey the identity and connections of components; the position, direction, and size of the graphic symbols are irrelevant to the meaning of the diagram.

What most distinguishes architectural design diagrams from diagrams in other domains is that the elements and spatial relations correspond to physical elements and spatial relations in the architectural problem. In other domains to make a diagram you must map the problem domain to

a set of diagram elements and spatial relations. For example, in an Euler diagram circles represent sets and graphical overlap of the circles represents set intersections. By contrast, an architectural design problem is essentially spatial; and the diagram is simply at one end of a continuum of graphical representations used throughout the design process. The symbols used to represent elements —walls, rooms, building components—in an architectural diagram are not arbitrary, their shapes and sizes derive directly from the physical elements they represent.

In light of the continuum of graphical representations used in architectural design, it is useful to distinguish a diagram from another form of drawing also used in early design: the freehand sketch. We mean by an architectural diagram a drawing that uses geometric elements to abstractly represent phenomena such as sound, light, heat, wind, and rain; building components such as walls, windows, doors and furniture; and characteristics of human perception and behavior such as sight lines, privacy and movement, as well as territorial boundaries of space or rooms.

A diagram is made of symbols and is about concepts. It is abstract and propositional: its elements and spatial relations can be expressed as a set of statements. It explores, explains, demonstrates, or clarifies relationships among parts of a whole or it illustrates how something works (a sequence of events, movement, or a process). Its symbols may represent objects (e.g., a space or a piece of furniture) or concepts (e.g., service area, a buffer zone, accessibility or noise). For example, an arrow indicates the magnitude and direction of a force; a line indicates the ground without specifying material or exact location. A diagram omits detailed scale or realistic pictorial representations; it indicates spatial relationships only approximately using indefinite shapes. For example, a diagram may represent functional spaces in a floor plan as crude ‘bubbles’, showing only sizes, adjacencies, containment, and connections.

A sketch, in contrast, is about spatial form. It is executed with a finer resolution that indicates attributes of shape. A sketch often comprises repetitive overtraced lines made to explore precise

shape, rather than the intentionally abstract shapes of a diagram, and it uses graphic modifiers such as tone and hatching to convey additional information. For example, a plan or elevation sketch may explore the proportions of a building. A perspective sketch provides three-dimensional information about a scene, specifying the shape of physical elements and visual appearance from some location. Although a sketch falls short of precisely determining positions, dimensions, and shapes, it often provides more detailed information than a diagram.

Architects make many other kinds of drawing: softline (freehand) and hardline (drafted) schematic drawings, working drawings, as well as different projections (plans, sections, elevations, elevation oblique, axonometric). We focus on freehand diagrams. In the following section (2) we examine how architects use diagrams in several contexts. We look at books that teach students to design through drawing; we review recent studies on drawing in architecture; we look at architects' introspective accounts of design process. We then review (section 3) several empirical studies of drawing in architectural design. In section 4 we examine three examples of how diagrams support architectural design thinking. We conclude (section 5) with a brief discussion of some desiderata for computational support for thinking with diagrams in design.

2. Diagrams in architecture education and practice

Many books for architecture students focus on drawing methods and techniques. Lockard's Design Drawing Experiences proposes that the ability to "diagram" an architectural context depends on designers' knowledge of issues such as sun, wind, vegetation, traffic, and surroundings. He argues that diagramming can be used to explore variations of design problems and that it allows us to "see, comprehend and respond" to more visual information than we can remember from verbal notes (Lockard 1973). Laseau's Graphic Thinking (Laseau 1980) is a guide to making drawings for working out problems and communicating with others. He describes

drawing as a means for design development. As an abstraction of an architectural program, the diagram expresses functions, relationships between functions, and the hierarchy of those functions. It is an expression in a graphic language that consists of grammatical rules and vocabulary. Whereas verbal language is sequential, he argues that graphic language is simultaneous: “all symbols and their relationships are considered at the same time.”

In professional practice designing often begins with a diagrammatic depiction of the architectural program that is gradually transformed to more complex graphic representations by adding detail. One architect explained designing as a process of transforming and merging diagrams, “trying to take a structural diagram, a functional diagram, and a circulation diagram” and “combine them” (Rowe 1987). Designers often work by making diagrams or transcribing diagrams from their design team colleagues for further development (Graves 1977; Lockard 1977).

Case studies of architects use interviews, observations and works from portfolios to examine the integration of drawing in design practice. Lawson (Lawson 1994) interviewed ten well-known architects to study their methods, concluding that designers “find it hard to think without a pencil in their hand” (p. 141). Fraser and Henmi (Fraser and Henmi 1994) look at how different drawing techniques influence the making of architecture. They describe diagrams as drawings that engage in a “self-conscious reductive process,” clarifying a specific interpretation by excluding irrelevant information. They note that architects “symbolize...intangible factors such as movement, access, sound, view, function, and time..” (p. 110) in diagrammatic form to represent the abstraction and reduction of information. Herbert (Herbert 1993) examines graphical media in the design processes of six practicing architects. He describes a diagram as an analytic statement, a “composite of graphic marks and written notes.” This points out the common practice of mixing text labels and graphics in architectural diagrams. A diagram thus governs and transforms the meanings of verbal statements into a graphic context to help the designer solve problems. Herbert argues that drawings are more than merely a convenient strategy for solving design

problems: They are “the designer’s principal means of thinking” (p. 1). The designer “must interact with the drawing” (p. 121).

We find diagrams in the sketchbooks of famous architects such as Louis I. Kahn (Brownlee and Long 1991), Le Corbusier (Sekler and Curtis 1978; Guiton 1987), and Peter Eisenman (Eisenman 1987). Kahn in The Value and Aim in Sketching (Kahn 1931) mentions that sketches are as important to him as design problems. He argues that “drawing is a mode of representation” and regardless of the medium used, the value of a drawing is in the “purpose” of making. He argues that designers need to interact and work with a sketch, not just “crystallize” thoughts on paper.

On one hand, Kahn’s design diagrams in his proposal for Midtown Philadelphia (Kahn 1953) (Figure 1) show that he has a graphic symbol vocabulary for certain concepts: a plus sign (+) to indicate intersection, an arrow for traffic direction, and a ‘greater than’ sign (>) for garage. On the other hand, he stresses the importance of shape information in a design. For example, Figure 1 (a-e) below shows design diagrams for the Goldberg House in various forms. Kahn uses geometric shapes such as squares, arrows, and lines with textual annotations to explore the spatial arrangement of functional spaces. He explains that Figure 1-a, representing a version of the final design 1-e as four void corners inside a box, is actually (=) a “discovery” from 1-b that the interior (box) can extend out and the diagonals (lines) can be framed (in the endings). Diagrams 1-c and 1-d have the same four directional forces. However, the bounding shape is different (square versus irregular shape) and Kahn has chosen the square shape to develop into the final design 1-e. Figure 1-e shows a configuration of a living room (square) in the middle and buffer zone (hatching indicating “servant area, also insulation from room to room and circulation”) and the surrounding rooms (“area served”).

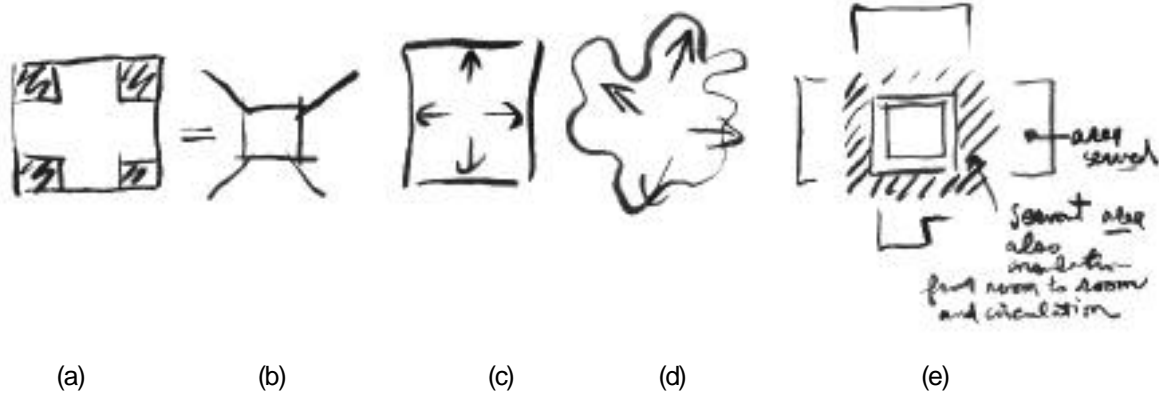


Figure 1. Diagrams with simple geometric shapes for Goldberg House design in Rydal, Pennsylvania (drawn by author after Louis I. Kahn, (Perspecta 1961).

Many architects discuss the role of drawing in their design process. For example, Michael Graves explains that the “referential sketch” serves as a “diary” or record of discovery (Graves 1977). It is a “shorthand” notation, a “reference” of an architectural theme recorded to be “used, transformed, .. engaged,” elaborated, and combined with other sketches in a later composition.

Christopher Alexander characterizes design as matching program requirements with corresponding diagram (Alexander 1966). He calls a diagram “any pattern which, by being abstracted from a real situation, conveys the physical influence of certain demands or forces....” (p. 85). The diagram is the “starting point of synthesis”; the end product is “a tree of diagrams.”

The margins of Kevin Lynch’s Image of the City contain a fascinating collection of tiny diagrams that illustrate ideas about the built environment and cognitive maps described in the text (Lynch 1960). Each diagram comprises only a few lines or symbols, yet the hundred or so diagrams demonstrate the diversity of meaning that a small set of symbols and spatial relations can convey.

These examples all argue that architectural diagramming is necessary for design thinking and the act of making; the shapes drawn influence how architects see and think about design problems.

These design diagrams facilitate the designer's reflection, dialogue and self critique and therefore serve the purpose of representing and testing an architect's intent. These anecdotal remarks attest to the widespread use of diagrams in architectural design, and to the belief among designers that diagrams do not merely communicate ideas to others, but that diagrams serve as a primary vehicle for thinking and solving problems. We turn now to look at some of the studies done by design researchers and cognitive scientists that examine what designers do when they draw, and how the visual and verbal modes correlate and interact in design.

3. Empirical studies of drawing in architectural design

Most empirical studies of design problem solving have been examinations of design protocols. Protocol analysis studies of design problem solving typically collect both verbal and graphical data. Eastman showed that the representations designers use – words and drawings correlate with the problems they find and solve (Eastman 1968). In his study of six subjects performing the task of improving a bathroom layout Eastman documents the design operations used, the objects manipulated and the “control mechanisms” employed.

Akin's (Akin 1986) protocol studies analyzed the “chunking” of architects' design acts while drawing and their shifts in attention. He used the time interval between drawing events to identify how architects group elements in memory. His study revealed several chunks: wall and window segments, steps, furniture elements of similar sizes that have close spatial relations.

Suwa and Tversky videotaped architects sketching to design an art museum (Suwa and Tversky 1996) and asked the architects watching the tape later, to report what they had been thinking (verbal post-design review protocol). Suwa and Tversky studied the relation between concepts as identified by chunks in these protocols and drawing acts. They classified the words mentioned in the verbal protocols into different categories: spaces, things, shapes, views, lights and

circulation. They argued that seeing different types of information in sketches drives the refinement of design ideas.

Akin and Lin observed that most protocol studies emphasize recorded verbalizations rather than drawings (Akin and Lin 1995). They discussed symbolic encoding of different modes such as drawing, thinking, examining, and speaking. They asked subjects (1) to reproduce a drawing from a printed transcript, and (2) to predict verbal data from a video tape of the design drawing process with the sound track suppressed. They found that verbal transcripts and drawings were complementary, echoing one other, and that novel design decisions often occurred when the designer was in a “triple mode period”: simultaneously drawing, thinking, and examining.

Schön portrays designing as a “reflective conversation with materials.” Using protocols of architects sketching to explore possible entrance locations for a library, he argued that design reasoning employs rules (Schön 1988) that derive from previously known types and may be “subjected to test and criticism” by reference to these types. Designers frame a design problem, “set its boundaries, select particular things and relations for attention, and impose on the situation a coherence that guides subsequent moves.” Schön uses design sketching protocols to illustrate “reflection-in-action.” He argued that designers first ‘see’ then ‘move’ design objects. The structure of design is a structure of “seeing-moving-seeing,” an alternation of designing (moving) and discovering (seeing). He categorized the kinds of seeing as (1) literal visual apprehension of marks on a page, (2) appreciative judgments of quality, and (3) apprehension of spatial gestalts (Schön 1992; Schön and Wiggins 1992).

Goldschmidt’s design protocol studies, like Akin’s, examine drawing together with verbalization. She proposes that sketching is a mode of visual thinking and imagery is a conceptual framework for investigation. Sketching, a sequence of design moves and arguments, an “oscillation of arguments” resulting in the gradual transformation of images, is a systematic dialectic between

“seeing as” and “seeing that” reasoning modalities. Goldschmidt showed that sketches are not merely representations of images designers already have in mind; rather, the act of sketching is a vehicle for design thinking.

Goel argues that, in sketching, designers employ a different kind of thinking than can be accounted for by the traditional computational theory of mind that is widely held in cognitive science (Goel 1995). Whereas this theory has worked well for building computational models of intelligent behaviors in well defined problem domains such as cryptarithmic, puzzles, and traveling, Goel argues that designers work with ill-defined problems, and that sketching exemplifies a kind of representation that can support problem solving in ill-defined domains. He argues that such representations are essentially different from those used in the problem solving domains in which the traditional computational theory of mind has succeeded. By these distinctions, thinking with diagrams (as opposed to sketches) in design would fall largely (though not necessarily entirely) within the symbol processing domains in which AI has been most successful.

In two studies Do demonstrated that the elements of a graphic vocabulary are associated with specific design concerns (Do 1995). In her first study, which used diagrams and stories from the case based design aid Archie (Zimring, Do, Domeshek and Kolodner 1995), sixty-two architecture students performed four tasks: making diagrams from stories, writing stories from diagrams, pairing diagrams and stories, and commenting on given Archie diagram-story pairs. She found: 1) a small lexicon of symbols were used and arranged in conventional and consistent ways (Figure 2); 2) certain views were used consistently to illustrate certain concerns (e.g., plan views for spatial arrangement and section views for getting light into a building); 3) keywords from stories served as labels in diagrams and vice versa; and 4) most students could read and understand one another’s diagrams.

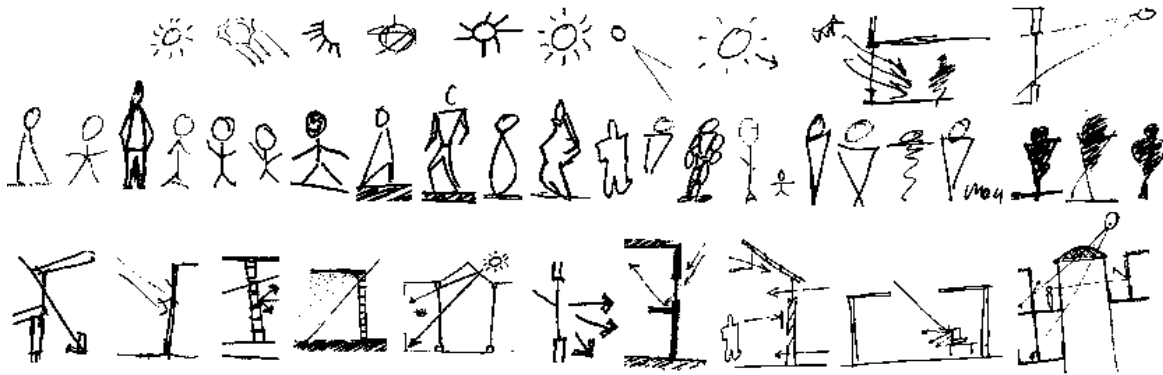


Figure 2. Designers used conventional symbols and configurations for architectural concepts in diagrams.

A second study (Do 1997; Do 1998) verified the conventional use of drawing symbols in design. Architects were given site dimensions and required functions for an office design problem, and they were videotaped designing while an observer took notes. The architects were asked to do a conceptual design and to focus sequentially on four different concerns: spatial layout for zoning different work areas, daylighting, visual access and privacy, and placing a large conference table in the design. The findings from this study of design drawing corroborated the first study in which students were asked to make drawings from stories and vice versa: Architects drew conventional diagrams that correlated with the task at hand.

In sum, the design protocols reviewed here acknowledge that designers use graphic symbols and that the drawing marks they make are linked to verbal protocols and design thinking. These studies discuss several important issues: First, they verify that designers use freehand drawings when thinking about special design concerns; second, they demonstrate that the reasoning is related to design drawings; and third, they suggest that different types of information are embedded in the design drawings. This indicates that designers do use a lexicon of graphic symbols and configurations when thinking about design and that a design drawing may employ different symbols to represent different types of information.

4. Examples of using diagrams in architectural design

With this background of anecdotes about the use of diagrams in architectural education and practice, and empirical studies of drawing in design, we turn to some examples of how architects use diagrams in design thinking. We offer three examples of thinking with diagrams in architectural design. The first two examples are similar to the use of diagrams are used in other domains: the visual representation supports analysis and inference about a problem. The third example appears to be particular to design: the diagram is an abstraction for detail to be filled in later. We argue that the diagram's imprecise quality helps the architect keep in mind its abstract intention.

Figure 3 shows a typical kind of diagram architects sometimes make to consider the layout of functions in a floor plan. Architects use these 'bubble' diagrams to explore relationships among the sizes, adjacencies, and approximate shapes of the spaces needed for various activities. The architect sometimes draws arrows or lines between functions that must communicate, or small ticks to indicate an adjacency requirement between two functions, as distinct from pairs of functions that simply happen to be adjacent in the drawing.

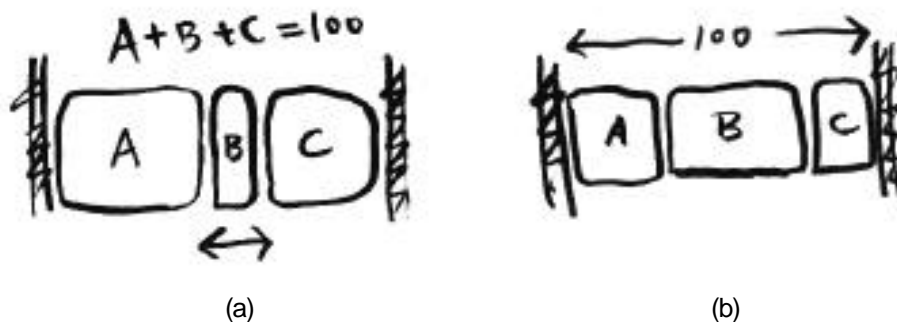


Figure 3: a bubble diagram illustrates dimensions and adjacencies among functions in a floor plan.

A bubble diagram like the ones shown in Figure 3 helps the architect consider possible changes to the design. Each bubble represents the space needed to carry out a function (living, dining, sleeping, etc.) For example, were the architect to enlarge one space, the diagram reveals how the adjacent spaces would need to be correspondingly adjusted to remain adjacent and stay within their own size constraints (e.g., $A+B+C=100$). On the other hand, the architect can see when squeezing the diagram would make the dimension of a space too small for its intended function (when space B is enlarged, space A becomes too small). In short, a bubble diagram helps the architect understand the constraints of a floor plan and the consequences of proposed changes to the design. The diagram makes adjacencies, overlaps, and relative dimensions available by inspection.

Figure 4 shows another use of a diagram in architectural design. Here the architect wants to ensure a view of a landscape feature from a certain place in the building. Sight lines allow the architect to test this predicate. “Is the mountain visible from the living room?” and immediately to see the inference: “If I extend the wing of the house out here, I’ll obstruct the view from the living room”. If the architect retains the sight lines as part of the working design diagram, they can also serve as constraints on the developing design: don’t put anything in the way.



Figure 4: sight lines to a distant landscape feature display architectural design constraints.

In these examples, diagrams help the architect keep in mind various constraints that are to apply to the emerging design. The diagram allows the architect to read off visually whether the design in its current state satisfies a certain predicate: an adjacency, a line of sight. In both examples, it would be possible, though tedious, to compute the adjacencies and sight lines geometrically or arithmetically.



Figure 5 (a) rough diagram; (b) with additional details; (c) with alternative details

The third example, in Figure 5, shows a use of diagrams in architectural thinking that is somewhat different from the first two. Here, the elements of the diagram are abstractions for more specific details that are to be filled in later. The design process is demonstrated here as incremental formalization. For example, the rough diagram in Figure 5-a may later be detailed as shown in Figure 5-b. Alternatively, it may be detailed as in Figure 5-c. The abstract diagram (a) is sufficient to think about design concerns at a certain level; in fact additional detail (b or c) is not especially relevant to decision making and is only likely to distract and perhaps confuse the designer. Thus, an important design skill is matching the level of detail of a diagram to the level of decision making.

The crude character of the freehand diagram in Figure 5-a could be ‘beautified’ by turning rough and approximate shapes and lines into regular geometric primitives as might be produced with a drawing template or by a structured CAD program (Figure 6). For the purpose of designing, that

is, moving from the diagram to a more detailed specification as in 5-a to 5-b or 5-c, this beautified drawing is less useful than the original freehand version.

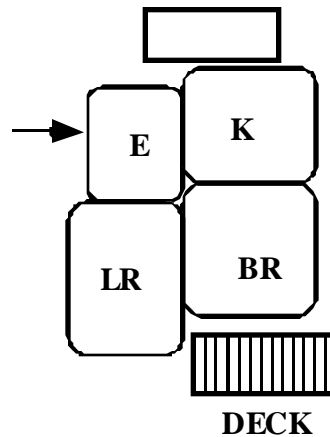


Figure 6: a 'beautified' version of the diagram in Figure 5-a

The rough nature of the freehand diagram serves to remind the architect that it is not a completed work. An inexperienced designer often 'builds the diagram', i.e., proceeds from an initial diagram such as a bubble diagram by simply converting the diagram lines into a floor plan; the seasoned architect elaborates on the diagram, adds detail, and responds to local conditions. The architect will be less tempted to build the diagram if its form is a reminder that additional detail must be worked out, just as a double-spaced typescript invites revision more readily than a tightly formatted page layout of text. The 'beautified' diagram of Figure 6 looks like a floor plan. It is not one. No new information has been added. The architect is likely to keep the geometric shapes and sizes for rooms, the lines for walls, instead of seeing the diagram for what it really is – an abstract plan that must be further refined and specified.

5. Computational support for thinking with diagrams

We come then to the question of computational support. What capabilities are needed in a computer program that would support designers in thinking with diagrams? Based on the above,

we believe that the following are desirable components of a system for computational support of thinking with diagrams in architectural design:

- Freehand drawing input, as opposed to structured diagram entry and editing.
- Maintaining spatial relations among elements as the diagram is transformed.
- Recognizing ‘emergent’ patterns and configurations in a diagram.
- Performing transformations that carry one diagram to another.
- Identifying similarities and differences among diagrams.
- Representing designs at varying levels of abstraction and detail.

These desiderata are represented, to varying degrees, in our prototype system, The Electronic Cocktail Napkin (Gross 1996; Gross and Do 1996). We consider each below.

Freehand drawing input, as opposed to structured diagram entry and editing

Perhaps the strongest argument for freehand drawing is that architects have learned to work that way. They think with a pencil in hand. The kinesthetic act of drawing the diagram seems to help the designer focus on the problems at hand, to consider the relationships among the parts of the design. Although structured diagram entry (choosing elements from a palette or menu) is simpler for the computer program because it avoids the problem of recognition, it also seems to short-circuit the opportunity for the designer to reflect while thinking.

We have observed that the imprecise appearance of a freehand diagram serves to remind the designer that the diagram is not a completed design, and that additional refining and specification must be done to develop the diagram. Therefore diagram beautification and cleanup is not an advantage, at least during the early concept making and manipulation steps of design.

Finally, the designer can draw a diagram faster using a pencil than with a structured editor.

Laseau offers time data for making diagrams with a structured drawing program (Laseau 1986): A

simple diagram that can be drawn in seconds with a pencil takes several minutes to make with a structured draw program (p. 135).

Recognizing spatial relations among elements and maintaining them as designer transforms the diagram.

The diagrams a designer draws convey elements and spatial relations. The program ought to recognize both the elements and spatial relations in the diagram. It ought to assert the spatial relations on the diagram as constraints, so that as the designer manipulates the diagram further the spatial relations continue to hold.

Recognizing 'emergent' patterns and configurations in a diagram.

One feature of diagram representations that seems to aid inference is the ability of the human eye to recognize patterns. These patterns are sometimes called 'emergent' because they are not necessarily drawn intentionally and explicitly, but can be perceived once the diagram is made. A computer system to support thinking with diagrams ought to be able to recognize patterns in the diagrams the designer makes (Edmonds, Moran and Do 1998). The machine ought to be able to recognize not only the patterns the designer draws intentionally, but also emergent patterns that result as unintended side effects.

Performing transformations that carry one diagram to another.

A computer system to support diagram reasoning ought to be able to acquire and store graphical transformations of diagrams, which it can then apply on command. A transformation is essentially a graphical production rule, of the sort used in shape grammars (Stiny 1980). The transformation allows the system, upon recognizing the 'left hand side' pattern in a diagram to replace it with the 'right hand side' pattern.

Identifying similarities and differences among diagrams.

Thinking with diagrams requires, on occasion, comparing two diagrams and identifying the similarities and differences. For example, the computer system ought to be able to observe that diagram A and diagram B are identical, except that the circles in A are squares in B; that diagram A is a superset of diagram B, containing several additional elements. We can think of several applications of this identification of similarity and difference. One obvious application is to use diagrams as index entries for a database. A similarity measure would allow near as well as exact matches to retrieve database items using a diagram index.

Representing designs at varying levels of abstraction and detail.

Finally, a computer system that supports diagram reasoning for design should allow the representation of diagrams at various levels of abstraction. As we have observed, diagrams are inherently abstract representations. In architectural design, diagrams are abstract representations of more detailed physical forms. Thus, (and perhaps different from thinking with diagrams in other domains) an architectural design may be represented by diagrams at various levels of abstraction. That is, each element in a diagram at one level of abstraction may be represented by a diagram at a more detailed level down. The computer system should allow for and support these nested levels of abstraction in the diagrams.

Acknowledgments

We thank Alan Blackwell for convening the original Thinking with Diagrams workshop, which brought together an interesting collection of people representing a diverse range of disciplines. This material is based upon work supported by the National Science Foundation under Grant No. IIS-96-19856.

References

Akin, O. (1986). Psychology of Architectural Design. London, Pion.

Akin, O. and C. Lin (1995). "Design Protocol data and novel design decisions." Design Studies 16(#2, April): 211-236.

Alexander, C. (1966). Notes on the Synthesis of Form. Cambridge, Harvard University Press.

Brownlee, D. B. and D. G. D. Long (1991). Louis I. Kahn: In the Realm of Architecture. New York, Rizzoli International Publications, Inc.

Do, E. Y.-L. (1995). What's in a diagram that a computer should understand. CAAD Futures '95: The Global Design Studio, Sixth International Conference on Computer Aided Architectural Design Futures. M. Tan and R. Teh. Singapore, National University of Singapore: 469-482.

Do, E. Y.-L. (1997). Computability of Design Diagrams -- an empirical study of diagram conventions in design. CAAD Futures 97. R. Junge. Munich, Kluwer: 171-176.

Do, E. Y.-L. (1998). The Right Tool at the Right Time -Investigation of Freehand Drawing as an Interface to Knowledge Based Design Tools. College of Architecture. Atlanta, Georgia Institute of Technology: 370.

Eastman, C. M. (1968). On the Analysis of Intuitive Design. Emerging Methods in Environmental Design and Planning. G. T. Moore. Cambridge, MIT Press: 21-37.

Edmonds, E., T. Moran and E. Do (1998). "Interactive Systems for Supporting the Emergence of Concepts and Idea, a CHI 97 workshop." SIGCHI bulletin, a quarterly publication of the ACM Special Interest Group on Computer-Human Interaction 30 (1).

Edwards, B. (1979). Drawing on the Right Side of the Brain: a course in enhancing creativity and artistic confidence. Los Angeles, J. P. Tarcher.

Eisenman, P. (1987). Peter Eisenman House of Cards. New York, Oxford University Press.

Fraser, I. and R. Henmi (1994). Envisioning Architecture - an analysis of drawing. New York, Van Nostrand Reinhold.

Goel, V. (1995). Sketches of Thought. Cambridge MA, MIT Press.

Graves, M. (1977). "The necessity for drawing: tangible speculation." Architectural Design 6 (77): 384-394.

- Gross, M. D. (1996). "The Electronic Cocktail Napkin - working with diagrams." Design Studies 17 (1): 53-69.
- Gross, M. D. and E. Y.-L. Do (1996). Ambiguous Intentions. Proceedings, ACM Symposium on User Interface Software and Technology (UIST '96). Seattle, WA: 183-192.
- Guiton, J. (1987). The Ideas of Le Corbusier: On Architecture and Urban Planning. Paris, New York, George Braziller.
- Herbert, D. M. (1993). Architectural Study Drawings. New York, Van Nostrand Reinhold.
- Kahn, L. I. (1931). "The Value and Aim in Sketching." T-Square Club 1(6): 19.
- Kahn, L. I. (1953). "Toward a Plan for Midtown Philadelphia." The Yale Architectural Journal Perspecta (2): 10-27.
- Laseau, P. (1980). Graphic Thinking for Architects and Designers. New York, Van Nostrand Reinhold.
- Laseau, P. (1986). Graphic problem solving for architects & builders. Boston, Van Nostrand Reinhold.
- Lawson, B. (1994). Design in Mind. Oxford, Butterworth.
- Lockard, W. K. (1973). Design Drawing Experience. Tucson, Pepper Publishing.
- Lockard, W. K. (1977). Drawing as a Means to Architecture. Tucson, Pepper Publishing.
- Lynch, K. (1960). The Image of the City. Cambridge, MIT Press.
- Perspecta (1961). "Louis I. Kahn." The Yale Architectural Journal 7: 9-18.
- Rowe, P. (1987). Design Thinking. Cambridge, MIT Press.
- Schön, D. A. (1988). "Designing: Rules, Types and Worlds." Design Studies 9 (#3, July): 181-190.
- Schön, D. A. (1992). "Designing as Reflective Conversation with the Materials of a Design Situation." Knowledge Based Systems 5(3).
- Schön, D. A. and G. Wiggins (1992). "Kinds of Seeing and their functions in designing." Design Studies 13 (#2): 135-156.
- Sekler, E. F. and W. Curtis (1978). Le Corbusier at Work. Cambridge, Harvard University Press.

Stiny, G. (1980). "Introduction to shape and shape grammars." Environment and Planning B 7: 345-351.

Suwa, M. and B. Tversky (1996). What Architects See in Their Sketches: A Protocol Analysis. Artificial Intelligence in Design '96, Stanford University.

Zimring, C., E. Y.-L. Do, E. Domeshek and J. Kolodner (1995). Supporting Case-Study Use in Design Education: A Computational Case-Based Design Aid for Architecture. Computing in Civil Engineering, A/E/C Systems '95. J. P. Mohsen. Atlanta, GA, American Society of Civil Engineers. II: 1635-1642.