Expertise of the Architect

by

O.Akin

EDRC-48-09-88

Expertise of the Architect

Omer Akin
Department of Architecture
Carnegie Mellon University

Pittsburgh, PA 15213

November 1987

Abstract

One of the areas where the expertise of the seasoned architect comes out is in the initial structuring of design problems. During problem structuring the parameters and processes used in design are defined. Experienced architects modify these parameters both at global and local levels as a function of the success of their search process. Also experienced architects rely on "scenarios," acquired through previous experiences with similar problems, in order to initialize their problem structures or to redefine them.

Expertise of the Architect

Öner Akin

1.0 Expertise and the Professional

Architecture, like most of the engineering fields, entered the age of computing through the use of Computer-Aided Drafting tools during the '60s [15]. Subsequently, as the struggle to realize the levels of efficiency promised by automation kept intensifying, new research goals for computing applications in architecture emerged. These included the undertaking of mundane tasks with greater speed and accuracy, improving communication between various building-design professionals, responding to a greater number of design constraints in a shorter time, and reaching greater levels of precision and rigor in the design of buildings. These new avenues lead to the development of a myriad of tools suitable for design and production of building specifications, such as integrated databases, solids modelers, rectangular packing routines, scheduling and other information management tods.

As architects got busy with integrating these tools into the daily routine of the office, universities and R&D divisions of corporations were busy with the development of a new set of tools for design. These, generally called knowledge-based expert systems, attempt to bring techniques developed in the area of Artificial Intelligence to bear on design problems. Today a variety of automated tools exist starting with ones that are for the initial conception phase of designs through to ones for the production of construction documents; tools which can generate alternatives based on parameters specified by the architect; tools which can verify consistency and desired performance levels in alternative solutions; tools which bring a body of expert knowledge to bear on these generative and evaluative tasks.

While these goals present enormous challenges for architects and researchers, there is an equally, if not more, important simultaneous challenge for them. This is the codification of the architect's expertise. Inspite of the very long and by some measures illustrious history of the building activity carried out by architects, there is precious little known about the expertise of the architect. There is also constant debate and disagreement about the correctness or goodness of design even among experts.[1] Therefore, before we consider the expertise of the architect in empirical terms and the implications of this for automated design, it is necessary to consider, albeit briefly, the sources and definitions of the expertise of the architect.

Architects have consistently tried to distinguish their profession from its sister professions, or in some instances crafts, since its early beginnings in the 17th and 18th centuries. The primary reason for this has been the need to protect the business of the architect from invaders of a hostile kind. In the past these invaders have been the craftsmen and the artisans involved in building trades. More recently the threats have also been felt with respect to builders and developers as well. In the meantime, the area of expertise of the architect has been defined and redefined numerous times, sometimes as a result of reactionary positions towards potential invaders and at other times in order to identify it with those of existlng and more sympathetic practices.

in the 1Sth Century the architects aligned their gcais with that of the artist in an attempt to elevate themselves above the craftsman within the building industry. At the turn of the century, this was followed by a realignment with the goals of the political elite, then in the '20s and '30s the industrial revolution; next in the '50s ancTSOs the medical and legal professions; and finally in the 70s the manager and the developer. It is in such a complex cultural milieu that the definition of the area of expertise of the architect as a professional has evolved. Thus, the current popular image of the architect as one who is knowledgable about design and aesthetic concerns dates back to the early days of self-identification.[2]

Today, institutions of architecture both in educational and professional terms are inheritors of these historical circumstances. The salient assumption underlying the entire process of evolution is that architects possess an expertise which is germane to their field of practice. And in a sense every bock ever written on the subject from the first known source, entitled *Ten Books on Architecture* by Vitruvius[28] on describes an aspect of this expertise. In spite of the abundance of scholarly references of this sort, there is very little known about what as a professional the architect is most qualified and skilled at doing.

During the last two decades we have seen the emergence of a number of studies that deal with this subject. Some of these works see the architect's expertise in terms of a skill for representation [19][27], Others see it in terms of methodology and attempt to *prescribe* the design process [6][7][15][21][3O]. Yet others see it in terms of codifying and *describing* the intuitive design process as a form of information processing [4][9][25][26][29].

Work in the area of representation, the first approach, particularly in the area of shape grammars, has enabled the formal and systematic exploration of building types and made the study of rational decision-making easier. Attempts at prescribing how the design process ought to work, have lead to new insights for designers and suggested new forms of practice. Participatory planning, design by patterns, performance measures, and specifications are some of the concrete results of this approach. In spite of these remarkable advances neither of these two approaches explains the expertise underlying the use of the method and the skill the architect generally brings to his practice.

The third approach, the description of the intuitive design process, in essence is both an illusive and strangely enough a more traditional preoccupation than the former two. Vitruvius opens the first chapter of his first book [28] with a definition of the architect which foreshadows even contemporary ones:

> "The architect should be equipped with knowledge of many branches of study and varied kinds of learning, for it is by his judgement that all work done by the other arts is put to test. This knowledge is the child of practice and theory. Practice is the continuous and regular exercise of employment where manual work is done with any necessary material according to the design of a drawing. Theory, on the other hand, is the ability to demonstrate and explain the productions of dexterity on the principles of proportion.[11] (p. 5)

To cite a considerably more recent source, *Encyclopaedia Britannica* [12] defines architect as:

> "one who skilled in the art of architecture, designs buildings, determining the disposition of both their interior spaces and exterior masses, together with the structural embellishments of each, and generally supervises their erection."

The same source goes on to explain the involvement of the architect of the past with the construction process and his diminishing roie, during current times, in this respect.

As both sources suggest, the task for which the architect seems to bear the greatest responsibility, and therefore at which is most skilled, is design. Vitruvius attributes both practice and theory to skills directly related to designing, namely, Translating from drawings and using principles of proportion. *Encyclopaedia Sritannica* refers to determining disposition of spaces and massing and structural embellishment as aspects of design which constitute the architect's expertise. These descriptions, while insightful and probably correct, at best rely on their author's personal observations or, at worst, on second hand narrations of similar observations by others.

Systematic and explicit studies of the design process is a recent area of study triggered in the '60s and 70s after the advent of Systems Theory, Operations Research, and computers. In spite of the relative immaturity of the area the results so far are sufficient to show that the architect's design process is both more diverse and heterogeneous than what is suggested in the two sources quoted above, or in other scholarly works in the area, for that matter[4J. Design decisions, from architectural programming^ to construction or shop-drawing phases of the design dellvery process, are made with the participation of many others, such as clients, users, engineers, public officials, community organizations, site designers, developers, financiers, project managers, and contractors.

Knowledge brought to bear on the problem and the procedures of decision making also vary with each participant. Due to the diversity of the sources of this knowledge and the power of control which comes with the possession of knowledge, architects more often than not are mere participants rather than leaders in this process. The single phase of this complex process in which the architect is still the sole decision maker is that of *preliminary design.* It is generally believed that the essentials of the architects' creation are shaped during this phase. This is where the designer exercises his creative input and develops the central concept for the entire design which is critical to the development of all of the other phases. As a consequence, preliminary design among all the other phases of the design process, such as programming, design development, working drawings, bidding, construction, and so on, is the one which conforms to standards and conventions the least. And also, it is regarded both as one that is most relevant to the architect's expertise and one that is most difficult, if not impossible, to describe with any degree of precision.

Recent work shows both tangible progress and promise towards acceleration of research results in the future [2][3][[5][11][13][17][29]. In this Chapter we shall review some of the recent findings about preliminary design and try to describe the expertise of the architect based on these findings. Obviously, in light of the scope of the entire volume, our effort in this chapter will be confined to only a few of the most salient issues of this broad topic. Section 1 introduces concepts fundamental to this area. Section 2 presents findings of recent empirical work about the expertise of the architect. And Section 3 reviews the salient findings in relation to implications for Computer-Aided Design applications.

1.1 Architectural Design Problems

Some of the most important insights about the architect's expertise come from studies that show, in terms of preliminary or conceptual design, how well architects do in comparison to lay people [2][3][5][13][12]. FOZ[13] reports that architects, during the development of a *parti,*[4] perform much better than people not trained as architects because they: (a) examine the problem at breadth before selecting a particular approach to the solution, (b) sketch profusely as they consider ideas, (c) debate the full implications of even these ideas which have no *a priori* likelihood to succeed before they are discarded, (d) avoid adoption of any solution until after a number of strong alternatives are considered, and (e) use solutions known from prior experience to develop solutions for the present problem.

Henrion[i4]f more so than highlighting the differences between Architects and Non-Architects, has shown some of the remarkable similarities that exist between them. In solving well-defined space planning problems, both architects and lay people use similar approaches while working towards satisfying predefined constraints. These results suggest that architects, while clearly different in their approaches to designing in general, are indistinguishable from others when it comes to satisfying a set of predefined constraints.

Is this a contradiction in terms or is there a way of explaining how it may in fact be possible? It turns out that the answer to this question points towards a paradigm which represents one of the critical ingredients of the architect's expertise. This paradigm is the extra ingredient which is needed for solving ill-defined problems and thus explains the differences in the findings of Foz and Henrion as well as many other researchers who have studied the same topic. This is the central question we will try to address in this chapter.

1.2 Ill-defined Problems, Well-defined Sub-problems

Let us now consider problem solving in general terms before reviewing specific observations about design problem solving. Many ordinary problems, puzzles, and questions are solvable because they exist in a context of well-understood ground rules. When familiar with the principles of algebra, it is trivial to solve a set of simultaneous equations that have a matching number of equations and unknowns. When knowledgeable about reading road maps and signs, it is an easy task to travel from Pittsburgh to Washington, D. C. These are typical examples of well-defined problems.

Other problems, some extraordinary, others quite ordinary, present more challenging circumstances. Finding a new house to buy, especially in a new town; playing the stock market; starting an automobile which refuses to start; designing a new kitchen; are all examples of this category. Here, the problem-solver or the designer also has to use principles and conventions at least similar in form to those used in solving well-defined problems. The difference lies in finding ways to bring these principles to bear on the problems at hand which ordinarily neither beg nor readily accept such applications.

For example, the automobile which refuses to run may have stalled due to a failure of the distribution system or alternatively due to the failure of the condenser. One may or may not have all of the necessary tools to make the diagnosis or the repair that is needed. Furthermore the problem may be solved completely extraneously by taking a bus, taking a taxi, or towing the automobile to a garage. Hence the solution to the problem is a function of the statement of the problem. Is the problem that one can not go to work or that one can not sell the auto due to the breakdown? Is the problem to know what is wrong or is it to rectify it; and in this context what does "to rectify it" really mean?

In the case of problems resembling this latter set, which are usually called ill-defined, it is necessary to know: 1) how to decompose the ill-defined problem into well-defined parts, 2) how to resolve these well-defined parts, and 3) how to reassemble these partial solutions into a general solution for the entire problem [25]. In most recent literature in the area, this skill has been called prob'sm-structuring [2] or Puzzle-Making [£]. Problem structuring turns out to be one activity in which the experienced architect, compared to the lay person, displays a remarkable skill providing evidence about the true nature of his field of expertise [4][5].

1.3 Problem Structuring

The first step in solving any design problem involves the description of what needs to be accomplished and with what elements and resources this must be accomplished. Designing a house for example can be described as a need to organize a particular set of rooms (i.e., kitchen, dining room, living room, bedrooms, bathroom, and so on) in a particular way on a particular site.[5] The determination of the rooms which will constitute the house and their attributes forms the initial structure of the design problem. Given this or rather having described this in some form the architect can begin to manipulate the elements of the house with a clear evaluation function^ in mind.

As the architect develops solutions or partial solutions that begin to meet some of the requirements of the initial problem description, comprehensive evaluations of these solutions are performed. Next, the architect invariably alters the structure of the problem in ways which lead him to more successful results. A common form this restructuring takes is the addition or deletion of problem constraints or solution parts (rooms, furnishings, etc.) from the initial problem description.

Restructuring through constraint modification means the alteration of both the data used by the architect and the process to be applied to this data. For example, adding a set of new constraints or solution parts to the problem during restructuring implies that, in addition to satisfying these new constraints in the new solution the architect's focus of attention must also shift to these components of the design problem almost immediately. Similarly, deleting a set of constraints or solution parts implies that these constraints or solution parts should not be included in the solution and other parts of the solution affected by these changes have to be considered first during the next iteration of design.[7]

Studies of architects' behaviors [4] show that constraint modification occurs as a result of detecting conflicts in a given partial solution. As the architect realizes, for example, that two functions placed side by side interfere with each other's privacy he will modify the constraints of the problem to induce design measures which will eradicate the conflict either by relocating one of the functions or by introducing walls to separate them. This example illustrates the point that conflict detection is one of the keys to problem restructuring viewed as a process of developing successive approximations towards a viable solution.

Design, obviously, is not purely a process of successive approximations. In fact more often than not architects shift their orderly strategy of "evolving" a solution, almost without warning. This suggests that problem restructuring takes place in response to things other than conflict detection, for example through the examination of alternative solutions which may or may not be related to the ones under consideration. Even in cases where successive refinements of current solutions are viable, alternatives may be preferred over them. This is largely due to the recognition of alternatives as counterpoints to current considerations or as opportunities that open the door for multiple solutions.

1.4 Categories of Expertise

Before we examine any experimental results in detail let us consider a general description of the architect's expertise based on the preliminary notions reviewed up to this point. As implied above, the architect, in order to resolve ill-defined design problems, must be skilled both at resolving well-defined ones and at redefining the ii!-defined problem as a sequence of well-defined ones. In more concrete terms, a sizable portion of his training is geared towards configuring structural elements, stairs, door swings, and so on. These are well-defined sub-problems as they exist in completely specified contexts, as part of a design or a site. In addition, as the architect gains experience in design, he becomes even more skilled in knowing when and how to perform these sub-tasks, in other words, how to structure the design problem to match his personal capabilities.

It has been shown repeatedly in protocol studies of designers that, given a design problem, the architect first sets out to identify the important requirements of the problem [4]. Then he selects from these requirements a well-defined subset of the design problem: for example, configure the roof form, develop a *plan parti,* layout the stairs, locate the driveway, and so on. Each subset of requirements defines a certain sub-problem. As each sub-problem is solved the architect realizes new requirements that must be met and priorities that must exist between these requirements. As he incorporates these new priorities he in effect restructures the problem, setting up new sub-problems to solve. Cycling between different problem structures leads him eventually to the best set of requirements and responses which he can develop.

In (re)structuring problems, particularly ones that deal with composing functional entities, SUCH as the ones given in an architectural program, the architect uses several important strategies. These can be grouped under four topics: scenarios, alternatives, evaluation, and prototypes.

Scenarios. Architects create scenarios that organize parts of the architectural program into a plausible operational order. A scenario is an organizational idea, such as a hierarchical office, an open classroom school, a theatre in the round, where a consistent behavioral idea is in evidence. Such a scenario defines the principal proximities, hierarchical relationships, privacy and access patterns which have to exist between parts of the program. It also provides conceptual constructs which can be consulted in answering questions that arise during design: Is the program consistent with its context? Is the site appropriate? Should there be other functions anticipated? How can change of uses be accommodated overtime? In summary, the scenario is the proverbial "better" check-list of issues which must be considered during design.

Alternatives. Architects create new problem structures, often with the help of alternative scenarios, in order to avoid settling for a mediocre solution. In operational terms, alternatives allow the architect to select among several satisficing solutions [25] bringing the final solution closer to a *pareto optimal* one [22]. Different scenarios often enable the designer to study solutions which are of completely different types. This leads to the consideration of diverse possibilities and a more comprehensive understanding of the ramifications of design choices.

Evaluation. As solutions or partial solutions are developed architects evaluate the degree to which **these satisfy the overall** goals of their designs. If they find that certain requirements are restricting **the emergence of** "good" solution ideas, then these requirements become candidates for being discarded. If some desired solutions suggest requirements not yet identified in **the program,** these become addenda to the requirement list. If new scenarios are suggested by the earlier problem structures, then an entirely new set of requirements are developed and a new agenda of explorations is identified. Thus, evaluation of earlier design steps becomes the key for finding successful future steps for the design process.

Prototypes. Architects use formal and physical ideas to create problem structures. What if the site were over the waterfall rather than on the opposite bank? What if the building had no interior partitions? What if the building was a glass box? These hypothetical "what-if" questions illustrate historical circumstances surrounding the design of Failingwater by F. L. Wright or the Farnsworth House by Mies van der Rohe. These circumstances emerged from physical considerations and were so a!l-enconpassing that the requirements for the entire problem were developed from these decisions. !n other words the problem structure was the clear result of a physical order rather than an operational one, such as the ones cited above. In the following section we shall examine each of these strategies in greater detail, based on empirical results.

2.0 Empirical Study of Problem Structuring by Architects

in a series of publications by Akin, *et.af.* [2][3][5] the problem structuring behavior of designers as well as non-designers has been closely studied. In their latest publication, entitled "A Paradigm for Problem Structuring in Design," the authors focus on the mechanics of the problem structuring process and draw specific conclusions about the expertise of Architects (A) in comparison to both Students (S) of architecture and professionals who are Non-Architects (N).

In this study, protocols of six subjects from each of these categories were collected. Each subject was given half an hour and asked to solve a space layout problem. The problem was to allocate four functional areas, a Conference room (C), a Chief Engineer's room (CE), a room for two Staff Engineers (SE) and a Secretary area (S) in a given site. There were three different sites, square, rectangular, and L-shaped, each equipped with two exterior entry ways and three windows, shown in cardboard cutout form. The functional components of the problem were also represented as two-dimensional cardboard cutouts of the furniture pieces in 1/4" = V-0^M scale. Experimental design consisted of two subjects from each of the three subject categories solving the layout problem for each of the three sites.

Design behavior of all subjects were recorded on videotape. These protocols were transcribed as text and diagrams. The designs developed at the end of each experimental session are shown in diagrammatic form in Figures 1, 2, and 3. Transcriptions of subjects' verbalizations, which we shall refer to throughout this chapter, were in turn codified as operational segments and subsequently analyzed for underlying problem solving and problem structuring behaviors. Below we shall discuss the results of this work in terms of the four strategies outlined above.

2.1 Architects vs Non-Architects

A primary question we asked was how the performance of the Architects, compared to the Non-Architects and Students in general terms. Furthermore, we asked how these differences explained aspects of the problem structuring process in design. In evaluating the subjects' performance, a primary criterium used was the satisfaction of design constraints in the final solutions proposed. In all of the protocols examined, these constraints were related to at least one of five general categories: zoning of functions, efficiency of use, privacy of use, circulation and control of flow, and use of windows.

Zoning of functions deals with the division of available floor area into parts which correspond to individual or groups of functions required in the program. This is not only for the allocation of adequate space for each function but also for insuring proper spatial contiguity among the parts.

Efficiency of use is concerned with the appropriateness of the floor area allocated to the various functions called out or implicit in the program, such as circulation areas. Cramped arrangements as well as ones that are too loose are equally objectionable problems, because, often, looseness in one part implies that other parts are deprived of space which might have been otherwise available. Privacy of use includes constraints that require privacy needs of each function and avoidance of privacy violation due to proximity of other functions or circulation areas. A private function too close to the main entrance, for instance, is problematic, just as a public function which is isolated from public access.

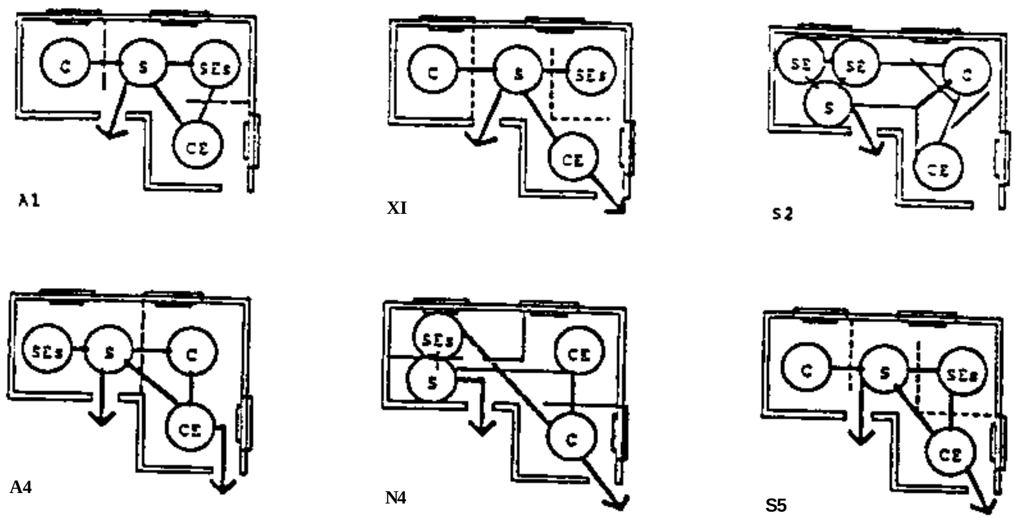
Circulation and control of flow has to do with establishing proper access links between functions that require them. Furthermore, control of public access through the strategic placement of the "reception" function and ease of access without trespassing through other use spaces are also important issues for proper circulation.

Use of windows stipulates the allocation of natural daylight and ventilation to those functions that need it without violation of the operationally of the existing windows. By the same token, proximity of windows to human functions is a generic requirement which must be met in most circumstances.

The sites, due to their own formal configurations, allowed or disallowed certain geometric layouts as solutions and influenced the ability of the subjects in satisfying these constraints. Let us now turn our attention to the designs developed for each of these three sites.

2.1.1 **The** L Site

In the case of Site 1, that is the L-shaped site (Figure 1) a natural, topological match between the site and the required functions (such as the one for Site 2, which is discussed below) did not exist. Thus, it was necessary to partition the site into two or three rectangles, each of which corresponded to a topological part of the L-shape, such as the wings and the corner, in order to accommodate the major components of the program, namely, Chief Engineer (CE), Conference (C), Staff Engineers (SE), and Secretary (S). These programmatic components, in turn, had to be organized into two or three logical clusters in order to match them with the partitions of the site.



Rgure 1: Solutions for the L-Shaped site.

This was accomplished in the case of the two Architect's (A1f A4) solutions by linking S with SE and pairing C with CE. In one case (A1) S and SE occupy the corner of the L-shaped site leaving the wings of the L to C and CE, and in the other (A4) the same functions occupy one of the wings of the L leaving the other wing to C and CE. In each case the access, entry, circulation, zoning of functions, use of windows, efficient use of floor area, and privacy issues are virtually problem free.

In the case of the Non-Architects and the Students there is no indication supporting a similar interpretation of the topology of the site. The outcome is a haphazard partitioning of the space into rooms and areas resulting in the division of windows by partitions (N1, S5), cramped and inefficient use of floor area (N1, N4, S2), artificially lit spaces (S2, N4), and unclear circulation paths (N4, S2).

Architect's Expertise

2.1.2 The "square" site

in **the case of site 2,** the "square" site (Figure 2), the topcbgical structure of the Site and its correspondence to the program is an obvious clue and was utilized by all subjects in their solutions, without exception. Given the proportions of the site and the location and number of windows and doors the only viable solution has been to place the three engineers near the window side and C and S on the door side, in spite of the need to provide natural light and ventilation for S none of the subjects were able to solve this problem and decided that it was not possible to do so without giving up more important things from their solutions, namely the zoning of the entire layout.

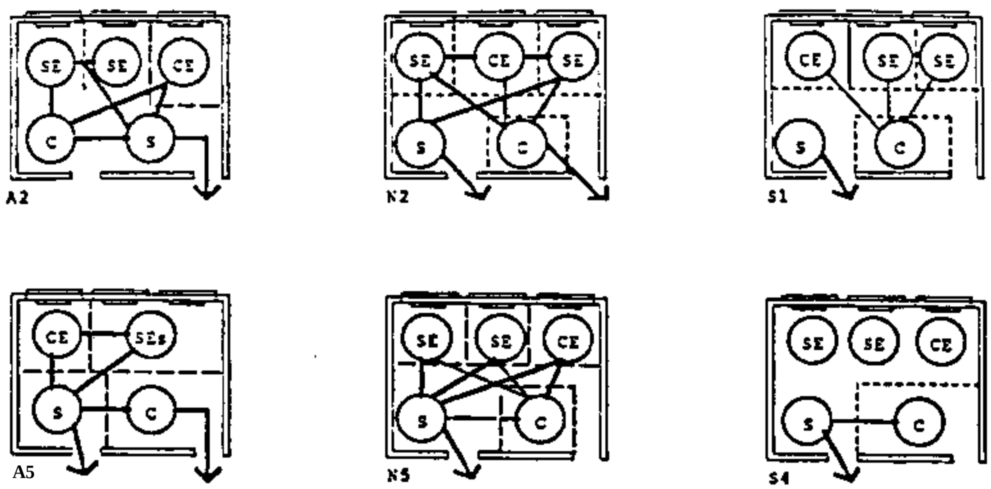


Figure 2: Solutions for the "Square" Site

Having resolved the general solution in at least topological terms the only improvements the subjects could affect on top of this had to do with the efficiency of use of the floor area, access between rooms, privacy, and organizational needs of the offices. Five of the solutions (two by Students, two by Non-Architects, and one by an Architect) enclosed C by walls. Three of these (N2, N5, S1) created hallways on all three sides of C causing severe inefficiencies in floor area usage. The other two (A5, S4) took advantage of the second entrance and created a private entrance way into C thereby including more useful floor area in C. The sixth subject (A2) avoided the problem entirely by enclosing CE and thus eliminating privacy-related conflicts between C and CE.

Both Architects (A2, A5) placed S in close proximity to the main entrance and paired up the two SE in such a way that they enabled the secretary to perform the role of "receptionist" with respect to all three engineers. Also S became a natural circulation hub and social center for the entire office. In the case of the two Non-Architects and the Student (N2, N5, S1) who enclosed C on all three sides this was not possible. The other Student (S4) who enclosed C on two sides placed CE behind C creating a very difficult circulation path between S and CE which had to pass through SE.

2.1.3 The Rectangular Site

In the **case of** Site **3,** the rectangular site, the problem was one of laying out four roons on a linear **relationship based** on **a** set of non-linear functional requirements and then to fit it into a long **rectangular space** with three windows. To resolve the difficulty of three windows for four functions, a zoning strategy similar to the one used on Site 1 is needed. To solve the problem of circulation in a long and narrow site requires the placement of the most frequently accessed functions in the center. Finally it is also necessary to minimize the doubling up of functions along the short, critical dimension of the site.

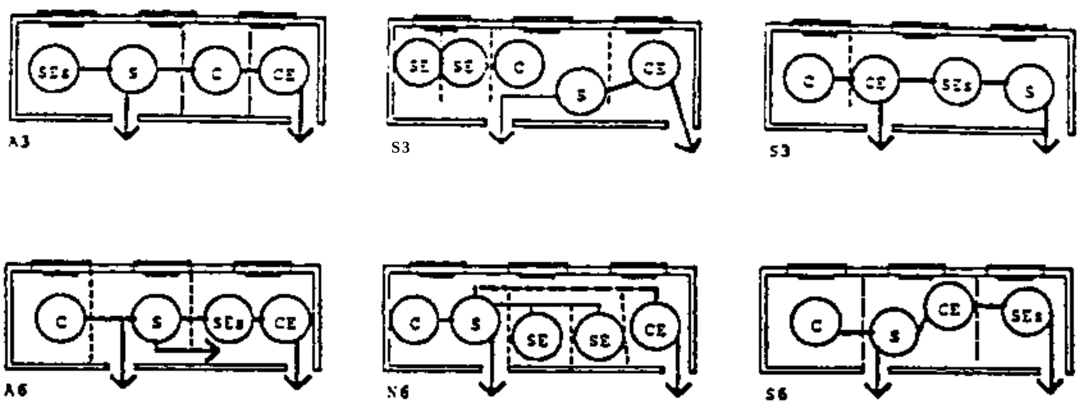


Figure 3: Solutions for the Rectangular Site

Four of the subjects (A6, N3, N6, S6) attempt to double up two functions or a function and circulation hallway along the narrow dimension of the site. This created tightness (N6, S6) and* disconnection from windows for some functions (N3, N6, S6). The most successful zoning strategy developed for this site seemed to be the one developed by the two Architects and one of the Students (S3). They placed all functions linearly on the site. The Architects also placed S near the center door and C and SE on either side leaving the other door for Chief Engineer's private use. The infrequent yet ceremonial connection between S and CE was served by two paths, either directly from the outside or through the function placed in between them. This clearly is a compromise, but one considered worth making in light of other compromises that would have been necessary in order to avoid it.

The problem of three windows versus four functions also did not find a graceful solution in this case. The strategy which comes closest to an acceptable solution was the clustering of two functions around a single window (A6, S6, N3).

Table 1 provides a checklist of the constraints satisfied by each subject.[8] In the end, it appears that, with the probable exception of Site 3, the solutions provided by Architects resolve more constraints than either of the other subjects. Architects, while generally more successful, did not perform better than Non-Architects, however in response to needs of "privacy." Also, they performed marginally better than Non-Architects in terms of access and Students in terms of use of windows. Non-Architects and Students on the other hand did generally poorer than Architects, with the solutions to Site 2 providing a notable exception, largely due to difficulties Architects encountered in dealing with this site.

TABLE' 1
Rating of Subject's Solutions

Site	S1: L-shaped						S2: "square"						S3: rectangular					
Subject	A1	A4	N1	N4	S2	S5	A2	A5	N2	N5	S1	S4	A3	A6	N3	N6	S3	S6
Zoning	•	•	0	0	0	O	•	•	•	•	•	•	•	0	0	0	•	0
Efficiency	•	•	0	0	0	•	•	•	0	0	0	•	•	•	•	0	•	0
Privacy	0	•	•	•	0	•	•	•	•	•	•	0	0	0	0	0	0	0
Access	•	•	•	0	0	•	•	•	0	0	0	0	0	0	0	•	0	0
Windows	•	•	0	0	0	0	0	0	0	0	0	0	0	•	0	0	•	0

Key • constraint satisfied
0 constraint partially satisfied
0 constraint not satisfied

2.2 Design Scenarios

The comprehensiveness displayed in the Architects' solutions is partially accounted for in their explicit use of scenarios. There is ample evidence in the protocols supporting this point. Consider for example subject A2 when he says:

> "Placed the three higher paid, more skilled people closest to the windows in deference to the secretarial space, (line 93)"

Clearly what he is considering is the hierarchical organization one finds in **a** traditional office setting in order to organize the physical layout of the functional components of the program. Later, A2's explicit remarks about the undesirable nature of "landscaped" office layouts, an alternative to the traditional layout, also reinforces this point:

> "Personally found lacking in offices, being able to carry on certain operations with the confidence that is required. I've occasionally had to ask the employees to leave the room. Landscaped office arrangements are [found] to be inadequate, (lines 141-143)"

Scenarios are also used, as stated earlier, to develop viable, alternative solutions. For example, Subject A4 after working with a formal entrance remarks about his desire to explore alternative scenarios:

> **"What that** means is this is private and you don't put the public, .clients back in the **drafting room.** They **don't** really go back there. They (SE) work here, this space here than **becomes** the main work room. Next strategy I would use in a different version is to sacrifice some of the better qualities, (lines 65-72)"

Subsequently, he goes on to reverse the entire layout in order to follow up on his stated intentions .

Scenarios provide for the architect topological templates which are adaptable to different programmatic requirements. Scenarios are topological in the sense that they define physical relationships without fixed geometric attributes. These relationships link functions in desired ways and still allow malleability in geometric terms. Thus, they can be accommodated in sites with specific geometric dimensions and shapes and fixed window and door locations. Non-Architects and particularly Students did not display any evidence that they were using scenarios and consequently, their solutions did not seem to benefit from known, topological patterns, as did the Architects'.

Non-Architects and Students, while evaluating partial solutions relied primarily on specific constraints and pragmatic conflicts. In doing so, the Non-Architects were preoccupied with drawing from their personal experiences of the office setting. Students, on the other hand, were relying almost solely on analytical techniques. After having developed his final solution, for example, subject N1 explains:

> "I am trying to fit the pieces in a way that I perceive to be functional organization. I can put the secretary over here and have people walk in the front door and find the CE. I feel they ought to see the secretary first, (lines 34)"

No doubt, the subject is concerned with making an office like the ones he has seen before, worked in or likes, if for no other reason, than for the reason of familiarity. As a result he can propose solutions which meet a number of performance criteria normally satisfied by these familiar patterns. However, the less than perfect results achieved are due to the difficulty of mapping solutions expressed as geometric entities into specific sites. The geometric properties of these sites — dimensions, locations of doors and windows - not being in agreement with the geometrically fixed physical features of the pattern recalled from experience, result in significant compromises. In all sites, with the exception of Site 2 which happens to be proportioned to accommodate just about any kind of small office layout, the solutions by Non-Architects have severe zoning difficulties (Table 1).

Students, in comparison to Non-Architects, operated from the point of view of a more liberal perspective, i.e., generating new layouts to fit the given problem. Yet, they confined their efforts only to analytical considerations. For example, subject S2 evaluated the final design in the following terms:

> "Seems entrance is all right. Because lot of people come in here. But there is tightness around SE desk.. Although they probably don't do all that much circulating. This seems very tight here. And there is a lot of space here. Need more space in the reception area...(so on)..(line 115)."

The strategy for developing a solution in this case is accomplished by isolating all performance issues and meeting them one by one. Because of such an analytical approach, Students in general were less comprehensive in their responses, ended up attempting to reinvent each layout from scratch and did not benefit from prototypical solutions, either geometric or topological. In the end, this strategy also resulted in solutions with shortcomings in terms of circulation and layout (Table 1).

It is not surprising then that in general the most number of constraints were recognized and met by Architects, while Non-Architects satisfied fever constraints but did it with less effort than Students who expended the most effort and satisfied almost just as few constraints. Architects were the only ones who explicitly and consistently used scenarios in structuring their problems as well as their solutions.

2.3 **Design Alternatives**

As stated earlier one form of problem structuring occurs due to a desire to consider other options or alternatives to the solution at hand. This represents a mechanism equivalent to searching for a *pareto optimal* solution as opposed to a satisfying one [25]. Accepting the first solution which satisfies the number of constraints necessary for a minimum level of acceptability is essentially equivalent to settling for a satisfying solution. Most experienced designers, including the Architects, however consider alternative solutions even if a satisfying solution is available. This results in the consideration of a much greater portion of the solution domain and possibly a solution better than the satisfying one, if not a *pareto optimal* one.

in the protocols we examined Subjects simply came right out and stated that they were about to do just that as they started to examine an alternative solution. There were a total of eighteen instances of this in the protocols (Architects 9, Students 6, Non-Architects 3 times). In the majority of these cases the alternative considered was one which reversed a problem parameter. The most common example of this was the switching of the main entrance from one exterior door of the site to the other.

Even in cases when a viable solution was at hand some subjects (A1, A4, A6) chose to consider alternatives. Some of these alternative solutions, which invariably resulted in restructuring the problem, lead to global modifications of the problem, such as reversal of main entry location, reorientation of the entire scheme, or swapping the locations of the two major components of the layout. Both Non-Architects and Students used similar problem restructuring strategies, and the operations they used were similar to those used by Architects. However neither Non-Architects nor Students came up with global conflicts or restructuring operations, while the Architects did.

2.4 Design Evaluation

Problem structuring ultimately hinged on the evaluation of the previous solutions or attempts at solutions. More often than not this took the form of detecting conflicts within a solution or partial solution. In the protocols there were five conflict categories roughly corresponding to the constraint categories indicated in Table 1: privacy, access-proximity, space, outside-opening match, and light and ventilation. Out of these the access-proximity category showed the greatest variance between subjects. Partly for this reason we shall devote more time later to discussing it. In considering the other conflict categories that lead to problem structuring we observe some important differences between the behaviors of the three subject categories.

First of all, Architects on the average restructured the problem more than (3.83 times, 40% of all conflicts) both other subjects (3.0 or 31.2%, and 2.66 or 27.9%, N and S, respectively). In case of the Privacy issue his pattern is most pronounced, 12.2% versus 5.2% and 3.5%, respectively.[9] In terms of Space (tightness and looseness problems) however, all categories were equally involved, 12.2%, 12.2% and 10.5%, respectively.

In the remaining conflict categories there were too few data points to draw any significant conclusions (total of 14 data points or 24.5% of all data points in a five by three space, in other words, on the average, less than one data point per category). However, some interesting patterns can still be discerned. One is the absence of light and ventilation conflicts in the Architect's and the Student's protocols. Another one is the oversight of a major programmatic element (i.e., the conference room) which took place only in two of the Student protocols.

It was also evident in the data that some conflicts used in the restructuring of the problem were local (particularly **for** privacy, access, and space conflicts) while others were global. It seemed that the restructuring responses of the subjects treated their domain in a consistent fashion: global conflicts resulted in global modifications of the problem and local conflicts in local modifications. **For** example, local conflicts such as lack of privacy in a room resulted either in moving that **room** to a more private part of the site or blocking the intruding spaces around it by buffer activities, such as reception area. On the other hand, when these conflicts were of a global nature the entire topological solution was modified in some way or a series of constraints were added to the problem definition. These global responses, often resulting from spatial conflicts of tightness or looseness, caused modifications of the entire layout and the arrangement of functions in the solutions.

In dealing with global conflicts or alternatives the designers treated the solution space in chunks, groups of design elements larger than the individual elements given in the problem (i.e., chairs, desks, typewriter desks, file cabinets, and so on). It is obvious that during design some chunking mechanism is at work which organizes the problem into manageable subparts in a hierarchic manner[4]. For example, the two SE were almost always chunked together. Architects in particular seemed to have more complex chunks which they manipulated with ease, such as the Entrance-Reception-S-CE or the S-CE-C sequence. This is consistent with findings linking expertise with chunk size in certain problem solving domains such as Chess [10], Go [23] and design [1].

2.5 Design Prototypes

It is clear from the above discussion that qualitative differences between the Architects' design process and those of Students and particularly of Non-Architects can be suggested. Non-Architects, for whom the typical office layout in a professional settings is a familiar entity, seemed to rely on prototypical patterns known to most lay people. This is consistent with the background of our subjects included in the category of Non-Architects, who were selected from full-time faculty in the professional colleges of Carnegie Mellon University. In contrast Architects,oe while familiar with similar layouts, spent a great deal more time trying to develop *new* solutions and layout patterns from scenarios. Students seemed to behave like the architects except they relied a lot less on typical solution patterns and a lot more on performance analysis.

These observations are further supported by the number of attempts made at restructuring design problems in the protocols. In the access-proximity category on the average Architects explicitly discussed and satisfied 11.67 constraints in their protocols. Corresponding numbers for Non-Architects and Students are 7.17 and 6.67, respectively. This indicates that Architects articulated and satisfied more constraints than either of the other two subject categories. Non-Architects came next and Students last.

Perhaps a more interesting implication of this can be seen by comparing these numbers against the number of times each subject group recognized conflicts due to the violation of an access-proximity constraint and then subsequently restructured the problem (Table 2). Here we see that the Students encounter the most number of constraints, on the average, 1.66; Architects the next, 0.83; and Non-Architects the last, 0.33. When corrected against the number of constraints ultimately satisfied (# of constraints satisfied /# of constraints used in restructuring) we see that Architects satisfy, on the average, 14.06 constraints for each conflict they recognize in response to access-proximity needs. The same number for Non-Architects and Students is 21.72 and 4.02, respectively.

Table 2

Satisfying the Access-Proximity

Constraints by the Subjects

A-P Constraints	Architects	Non-Architects	Students
1. Discussed	11.67	7.17	6.67
2. Used in Restructuring	0.83	0.33	1.66
3. Ratio of 1 to 2	14.06	21.72	4.02

These results in one sense are startling. When we consider the number of conflicts they encounter and the number of constraints they satisfy, Non-Architects seem to be most efficient in terms of access-proximity issues. Architects are next on this scale, satisfying about two-thirds as many constraints as the Non-Architects, followed by Students who satisfied about one-third as many constraints as Non-Architects and one-fourth as Architects.

It seems that the ordering between Students and Architects is as expected and the deficiency in Student's performance compared to Architects[1] can be attributed to the relative knowledge and skill each possess of their subject area. However, the dramatically greater efficiency observed in the performance of the Non-Architects suggests that they were doing something drastically different than both Architects and Students. On the surface this suggests that they were simply restructuring the problem fewer times than both Architects and Students. But why?

One plausible explanation is that they were relying on prototypical solutions familiar to them from their own work environments, as was argued earlier, rather than trying to create or invent new designs. As a consequence they were able to generate solutions which satisfied a number of constraints with ease and a small number of restructurings were necessary to develop a satisfying solution. This is supported by the total number of constraints explicitly considered and satisfied in the Architect's protocols in comparison to both Non-Architects and Students.

3.0 Conclusions

While one could say a great deal more about the specifics of problem structuring and its significance for the architect's expertise, we have covered many of the salient issues here and it is time to bring our exploration to a close. This will be done through two vehicles. One is summarizing a **few** of the major findings discussed above. The other is indicating the implications of these for computing applications in architectural design.

3.1 Summary of Observations

One of the significant results of the empirical work described here is the models of knowledge brought to bear on the restructuring function. There seems to be differences between the models relevant to each of the three subject categories. Architects, for example, use scenario-like constructs to represent knowledge about a given functional type, such as hierarchical, landscaped versus participatory office layouts. On the other hand, Non-Architects use actual physical templates and Students rely on performance evaluation, to bring appropriate knowledge to bear on the design problem.

Scenarios used by the Architects embody topoiogical assemblies which are instrumental in satisfying the essential relationships required by different prototypical office layouts. Scenarios are also representations of malleable, geometric relationships between the functional units of the program. As they are used to create layouts in the context of an existing envelope or site, their topoiogical parameters are kept and their geometric parameters adapted to the particulars of these external constraints. In this way they enabled the meeting of a large proportion, if not all, of the constraints called for in the specific site. Furthermore, as scenarios are selected and their parameters modified, new alternatives are generated. This is reflected in the design process as restructuring. Thus, understanding of scenarios and their use in design provides valuable insights about the problem restructuring function.

Physical templates, used by Non-Architects in lieu of scenarios, are potentially as powerful as scenarios. They embody *geometric* assemblages which satisfy the essential relationships required in different office types. And herein lies the reason why the information they contain about the relationships of functional components is less malleable and the adaptation of the template to a specific site is much more problematic. This is borne out by the results of the protocols of Non-Architects. With the notable exception of Site-2, which naturally lends itself to both geometric and topoiogical templates with ease, Non-Architects experienced severe difficulties in adopting their solutions to the sites. Consequently, while meeting many of the internal proximity and space requirements these solutions violated many other constraints, such as entry, window use, and privacy.

Students, who employed neither scenarios nor templates, approached the design problem in a constructivist manner, assembling their solutions from individual analytical observations about the way each partial solution performed in terms of each problem constraint. While theoretically sound, this approach failed to take advantage of known solution patterns and as a result did not resolve as many constraints as it otherwise would.

The second set of significant findings to be discussed here have to do with global versus local modifications of solutions. In restructuring the design problem all subjects relied on conflicts that arose and alternatives which suggested themselves during search. Some of these conflicts and alternatives were local. These were simply remedied by local modifications to the current design. Such conflicts and their remedies do not normally infringe on any aspects of the problem other than the location to which they are confined. Dealing with global conflicts, on the other hand, involved alterations in all or nearly all parts of the solution. Tightness in one part of the solution, lack of proximity between two or more functions, and unsuitably of the location of the main entrance into the office suit, for example, are global conflicts which normally require global modifications.

Architects, as evidenced by their behavior in the protocols, dealt with global conflicts and alternatives initially before bothering with local ones. Non-Architects and Students, on the other hand, consistently engaged in resolving local conflicts, first and foremost. They also tried to resolve the design problem altogether without getting involved in global modifications.

3.2 Implications **for CAD**

Study of problem solving behavior at this level of detail is motivated by the desire to learn more about human problem solving and as a result, to develop models and strategies which can be used in automating parts of the design processes. Thus, before concluding, it is necessary to refer to a number of ideas about how these results may benefit system designers particularly in the area of architecture. It is also necessary to caution the reader about their preliminary nature. Naturally, before effort is spent on building systems on these ideas, greater effort is needed to verify and develop them further.

First, it is important to recognize that one of the invariants in all of the protocols we examined was the distinction between local versus global constraints. Data in any CAD system should be organized to reflect these distinctions. Based on the experience of the designer, the scope and range of remedies necessary to resolve design conflicts can be seen at several levels of hierarchy. It should be possible to organize problem constraints which come, either implicitly or explicitly, with the problem description, into these levels of hierarchy. In this way, dependencies between conflicts caused by these constraints and design elements can be calibrated by individual users of the CAD tool.

Second, special representations of design elements are needed so that the dependencies between hierarchically organized constraints and design elements can be automatically propagated. Such a tool would allow the designer to predict the consequences of modifications made at one level on elements and representations, at another. If the secretary is moved, for example, in order to get it closer to the chief engineer, the system should alert the designer to other constraints that are being violated, that might be violated as a consequence or that might be satisfied easier, for that matter, all due to the initial move.

Third, the models of knowledge brought to bear on the design problem by the three subject groups suggest drastically different ways of integrating knowledge-based systems with the design process. Depending on the sophistication of the user, the CAD system may assume different parameters. Professional architects, the most likely users of CAD systems, would prefer to work with topology-based schemata in organizing their initial design ideas. Subsequently, as a prerequisite for finalizing these ideas into designs, architects need ways of testing geometric properties of their ideas as well as other performance-based aspects of the solution.

Fourth, in response to the architect's tendency to return to previously encountered alternatives or alternatives generated from earlier states of the solution, some kind of memory of earlier search states must be simulated in CAD applications. In its simplest terms this would be a chronological file of significant interim results, with the capability to return to these and generate new alternatives with relatively little effort.

Finally, a myriad of evaluative tools are routinely used by all subjects in determining the manner in which a design problem must be restructured. These include testing for adjacency, proximity, access, natural light, ventilation, circulation, privacy, spatial tightness and so on. Most of these are qualitative and context sensitive measures which are extremely difficult to quantify. However, it is almost inconceivable to imagine CAD systems which can be effective in the preliminary stages of architectural design, without capabilities such as these.

4.0 Notes

[1] In fact this is an issue which presents a particular difficulty in evaluating the results of the empirical data we will discuss in section 2.0.

[2] The political and professional contributions of two of the leading firms of 19th century America, by R. M. Hunt and McKim Meade and White provide some of the better known contributions of architects to the [M]high-style[M] image attributed to them even today [24]. In fact it is the efforts of such firms in the political, economic, and intellectual arenas which has lead to the creation of the modern professional powerhouses of the free-world: AIA in the US and RIBA in Great Britain.

[3] An architectural program, distinct from a computer program, is the set of functional and performance specifications which must be adhered to in order to develop an architectural solution.

[4] A term borrowed from the *Ecoles Des Beaux Arts* to refer to a diagram, usually in the form of a floor plan, of the basic concept of a design.

[5] It is obvious that the actual design of a house is a much more complex process with extensive technical issues involved. For the purposes of this discussion it has been abbreviated to one of its essential aspects, i.e., spatial organization.

6 Evaluation function, here, refers to an objective measure of success in the sense it is used in optimization problems.

[7] However, some constraints deleted due to the overconstrainting of the problem are not totally discarded but treated as secondary constraints which can be met but do not need to be met.

[8] Although there are many shades of gray in the degree to which any of these solutions satisfy a given constraint, in the table we provide three ratings: satisfaction, partial satisfaction and no satisfaction. For our purposes this provides an accurate enough measure to observe some general patterns.

[9] This is also consistent with Architects' difficulty in meeting the privacy constraint in a large number of the final solutions.

5.0 Bibliography

0.5 Akin, 0. (1980) "Perception of Structure in three-dimensional block arrangements," IBS Report #8, Department of Architecture, Carnegie Mellon University.

1 Akin, 0. (1986) "A Formalism for Problem Restructuring and Resolution in Design," *Planning and Design,* 13: 223-232.

2 Akin, 0, C. Chen, B. Dave and S. Pithavadian (1986) "A Schematic Representation of the Designer's Logic," in *Computer Aided Design and Robotics in Architecture and Construction.*

3 Akin, Ö. (1986) *Psychology of Architectural Design,* London: Pion Ltd.

4 Akin, Ö, B. Dave and S. Pithavadian (1987) "Problem Structuring Architectural Design," Report #87-01, Department of Architecture, Carnegie Mellon University, Pittsburgh, Pa.

5 Alexander, C. (1964) *Notes on the Synthesis of Form,* Cambridge: Harvard University Press.

6 Alexander, C, S. Ishikawa and M. Silverstein(1977) *A Pattern Language,* New York: Oxford University Press.

7 Archea, J. (1986) "Puzzle-Making: What architects do when no one is looking,- in *Computational Foundations of Architecture,* edited by Yehuda Kalay.

8 Broadbent, G. (1973) *Design in Architecture,* New York; John Wiley and Sons.

8.5 Chase W. G. and H. A. Simon (1973) "Mind's Eye in Chess," in *Visual Information Processing,* ed. by W. G. Chase, New York: Academic Press.

9 Eastman, C. (1970) "On the analysis of the intuitive design process," in *Emerging Methods in Environmental, Design and Planning,* ed. by G. T. Moore, Cambridge: The MIT Press.

10 *Encyclopaedia Britannica* (1957) vol 2, pp. 270-271.

11 Foz, A. (1973) "Observations on designer behavior in the parti," in *Proceedings of Design Activity International Conference,* London.

12 Henrion, M. (1974) "Notes on the Synthesis of Problems: An Exploration of Problem Formulations Used by Human Designers and Automated Systems," Master's Thesis, Royal College of Art, London.

13 Jones, J. C. (1980) *Design Methods,* New York: John Wiley and Sons.

13.5 Kemper, A. M. (1985) *Pioneers of CAD in Architecture,* Pacifica, CA: Hurland/Swenson.

14 Krauss, R, and J. Myer (1970) "Design: A Case History," in *Emerging Methods in E nvironmental Design and Planning,* ed. by G.T. Moore, Cambridge: The MIT Press.

15 March, L. (1976) *The Architecture of Form,* New York: Cambridge University Press.

16 March, L. and P. Steadman (1971) *The Geometry of Environment,* Cambridge: The MIT Press.

17 Mitchell, W. (1977) *Computer-Aided Architectural Design,* New York: Van Nostrand Reinhokj Company.

18 Moore. G. T. (1970) *Emerging Methods in Environmental Design and Planning,* Cambridge: The MIT Press.

18.25 Radford, A. D. and J. S. Gero (1986) *Design by Optimization in Architecture and Building,* New York: Van Nostrand Reinhold.

18.5 Reitman, W. (1976) "Skill perception in Go: deducing memory structures from inter-response times/[1] *Cognitive Psychology,* 8,336-356.

19 Saint, A. (1983) *The Image of the Architect,* New Haven: Yale University Press.

20 Simon, H. (1969) *The Sciences of the Artificial,* Cambridge: The MIT Press.

21 Schon, D. (1983) *The Reflective Practitioner: How Professionals Think in Action.* New York: Basic Books Inc.

22 Steadman, P. (1983) *Architectural Morphology,* London: Pion Ltd.

23 Vitruvius (1914) *Ten Books on Architecture,* New York: Dover Publications Inc.

24 Wade, J. (1977) *Architecture, Problems and Purposes,* New York; John Wiley and Sons.

25 Zeisel, J. (1981) *Inquiry by Design: Tools For Environment-Behavior Research,* Monterey, Ca: Brooks/Cole Publishing Co.