A COMPUTER-AIDED PARTICIPATORY DESIGN SYSTEM

by

Steven Yale Handel

B.S., M.I.T.
1968

M.A. S.U.N.Y. at Stony Brook
1969

SUBMITTED IN PARTIAL FULFULLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ARCHITECTURE

at the

MASSACHUSETTS INSTITUTE of
TECHNOLOGY

June, 1975

Signature of Author .....
Department of Architecture
May 9, 1975

Certified by .....N. Negroponte...
Thesis Supervisor

Accepted by .....
Departmental Chairman

ABSTRACT

This report describes a computerized design aide called U-DESIGN. It is a system of programs designed and produced to demonstrate a role for the computer that has often been overlooked - that is testing. In view of the difficulties that "problem-solving" approaches are having in being accepted, this may be the only effective role for the machine in the creative portion of the design process. The thesis suggests methods for using computer testing in the process that minimize the interruptions to the user. It proposes a method of using testing in the "background" of an interactive process. It doesn't try to solve spacial arrangement problems. It leaves that task to the user, but it helps him in a number of ways. It uses a problem-solving-like method adapted from IMAGE to correct minor errors and to indicate where larger ones are occuring. The computer's activity is seen on the screen motion. The motion signals the location and source of problems in the arrangment.

ACKNOWLEDGMENTS

Special thanks is due to all the people associated with the Architecture Machine Group for help in a lot of little and big ways. I am particularly grateful to Guy Weinzapfel who devoted a lot of time and energy to this project.

Many thanks to Nicholas Negroponte for providing me with the opportunity to work with him and the Architecture Machine.

And thanks the the National Science Foundation who provided funds for Nicholas who in turn could provide funds to support this work.

And finally thanks to my wife Phyllis for her help and understanding during the hectic days of the final preparations for this report.

Steve Handel

TABLE OF CONTENTS

I.

INTRODUCTION

This project has been supported by the National Science Foundation through a grant to the Architecture Machine Group at M.I.T. (1) It's an attempt to produce a computerized system that will help make difficult architectural design decisions.

In the recent past, there have been two previous attempts at producing computer aided design systems at M.I.T. Both attempts have been directed at the architect or the novice interested in doing architecture. One is IMAGE (2), a tool to solve complicated space planning problems for the architect in professional practice. The other is YONA (3), ar approach currently being developed to lead a novice user through the process of designing his own home, and to solve his spacial arrangment problems with him.

This thesis is, in many ways, a response to the difficulties encountered by these projects, and others not so unlike them, developed elsewhere. None of these systems has been successful or complete enough to gain any serious use in real practice. They don't provide enough of a service to justify the inconvenience of using them, let alone the cost. This project suggests an approach that avoids the area of problem solving, a source of mojor difficulty, and offers a new system called U-DESIGN which demonstrates how that approach is likely to work.

The project has evolved around a few philosophical precepts. Central to them all is the thought that the role for the computer in an architectural design process is less as a problem solver than as a drawing aide and critical advisor. The user can make better problem solving decisions himself. The other ideas concern details of that critic-client relationship: 1) that the machine should do testing and produce reports on its own initiative in a non-obtrusive manner; 2) that the testing calculations can be performed continuously as a background process; 3) that the machine should provide a hierarchy of design diagrams, where each level increases in complexity and where the diagram itself serves as a testing device as well as a design medium; and 4) that there is a role in this process for dynamic diagrams (that is pictures in motion). An important aspect of the system is that it has can operate on small inexpensive computer hardware. It is anticipated that sometime in the near future these machines will be cheap enough to be owned by the individual architecture firm.

The major portion of the effort of this project has been directed toward the creation of a working product - a system of computer programs that can demonstrate and permit experimentation with the ideas proposed here. This product, U-DESIGN is now available as a demonstration on the Architecture Machine computers.

The system attempts to be broad enough to carry the user through a signifigant portion of a design process. In this case, as is common with computer aided design attempts, that means through only the space planning (or parti) stage, though there is a framework that promises to permit extension of the process into its more refined stages.

To achieve the required breadth in a short time, it has been necessary to sacrafice depth in many areas. Often, where many facilities are envisioned, only a few demonstration capabilities are actually programmed. It is expected that missing portions will be filled in as the system develops over the next few years. Toward that end, the elements of the system have been kept especially modular ard the organization has left many "hooks" onto which future additions can be hung.

II.

ANOTHER APPROACH

A. THE NEED FOR SOMETHING BETTER

1. FAILURES WITH PROBLEM SOLVING

To date there is no computerized problem solving tool to help the architect design that has gained any use in the "real world." All attempts at producing such a tool have been hampered by considering too narrow a range of issues and succeeding poorly at that.

Most of the effort has been concentrated in the area of space planning or computer graphics. This is generally acknowledged to be a useful starting point, and this thesis has no quarrel with that approach. But there must ultimately be some facility for carrying the work on to the more sophisticated issues of architectural design.

There has been no lack of previous effort in the problem solving area. The long list of attempts has been summarized in a number of articles.(4) The attempts fall into two categories. There are those, like the predecessor of them all CRAFT (5), which deal with a very limited set of considerations, usually just distance, and thus have obvious shortcomings. And there are those like IMAGE (6) which attempt to consider the broader range of form issues. IMAGE, for example, addresses the issues of visual connectivity, building envelope and site features. It has a theoretical

framework that can be made to deal with any geometric relationship between two spaces.

But there is a penalty to be paid for such a generalized approach. Experience with IMAGE has shown that the program is apt not to find a solution even in relatively simple situations where one is obvious to the user. The more recent IMAGE publications (*) acknowledge this failure and therefore have pointed toward a new application, testing, that bypasses the solution-generation question altogether.

2. UNCOMFORTABLE INTERACTION LEADING TO NON-CREATIVE BEHAVIOR

Computer aided design systems have been characterized by a failure to provide an interaction environment that has been comfortable for anyone but the dedicated computer user. Computer systems have forced inconvenient, unfamiliar modes of working on potential users. It may be unreasonable to expect that dealing with a machine can ever become comfortable, but the interaction should at least avoid being counter productive, tedious, and alienating.

A lot if this difficulty can be attributed to inadequate communications hardware. Some systems have had to go to the extreme of producing graphics on typewriters. But even with the best hardware, systems have been difficult to use. Their

first problem is they require intricate command languages that are unintelligable without lengthy explanations and tutorials.

Succeeding at that, there is still the problem that the common styles of operation tend to restrict the creativity of the user. Computer systems usually leave a person continually waiting for the computer. When there are long calculations to be done, and the calculations are long in problem solving applications, the user issues a command and then has to sit passively while the machine works. And when the machine finishes, he issues another command and waits again. Whatever creative thinking he starts is continually interupted by the machine.

This is another problem pointed out by IMAGE. IMAGE has shown that such a design process does tend to encourage non-productive behavior and stifle creativity. The user often ends up thinking in the limited terms of the machine. By not being encouraged, a user is essentially discouraged from exploring possibilities not realized by the machine. By having to be the driver of the machine, he is discouraged from thinking about considerations not modeled in the machine. He has no good opportunity to explore those tangential issues that may only be hinted at by the machine's solution. For example, if the machine has not been programmed model materials, one won't see the user considering materials either.

B. THE COMPUTER AS GRAPHIC AIDE AND PARTNER IN THE DESIGN PROCESS

1. AS A REPRESENTATIONAL TOOL AND MEMORY AIDE

Even though it may be inappropriate as a problem solver, there are a lot of contributions that a computer system can make. The computer can simplify the job of creating the sketches by which a designer models his design solutions. It can refine his quickly drawn lines. It can facilitate quick alteration of drawings and diagrams. It can make inferences from the designer's stated intentions and slightly modify incorrect elements where need be. And with a fast interactive system where mistakes can be corrected quickly and easily, there is no need to be afraid of an occasional incorrect inference.

With the description of a design solution in its memory, the machine could provide a variety of "modeling" diagrams and pictures to let the designer evaluate his own design. A common vision is of the machine producing perspective views for the design, taking him on a walking tour through his proposed building. To whatever extent it is carried, however, the idea is to give a designer a better representation of his product for less effort.

The machine can serve as a memory aide for the designer. In a complex project, this can be quite an important tool. It can remind him of intentions generated at an earlier time. It can help him keep track of a multitude of desires as he struggles to find an effective tallance between them. Also there is a growing body of evidence that suggests that designers use graphic images as a memory device.(8) It seems that the more successful designers use this more often. This would mean that a good graphic capability is, in itself, a memory aide. The machine can make this kind of tool available to the novice who does not have the drafting skills of the more accomplished architect.

2. AS A TESTING DEVICE

A computer can participate in the design process as a testing device in two ways. First, it can warn about future road blocks that a designer may be inadvertantly introducing into his program. For example, consider one application programmed for U-DESIGN. The computer can test the planarity of the connection scheme specified amongst rooms of a building. If the graph is not planar, the designer can be warned that there will never be a solution for his problem if he has to stick to one floor level. Many other such tests can be postulated. For example; does a scheme mean some rooms can have no outside exposure, will all rooms be able to face in the desired directions, will it be possible to meet standard building code requirements.

Second, the computer can test how well a given configuration meets the designer's stated intentions. That testing of this sort can be an effective role has been demonstrated by IMAGE.

A good testing tool is an aide in problem solving. Recent publications acknowledge failures in problem solving. (9) They suggest a strategy of mixing generation (computer problem solving) with manual rearrangements by the use and using a new "testing" feature to identify problem areas. During the generation process, a user is directed to ask for a report of the most severe problems and then make some corrections himself before requesting further generation. This way, he helps the generator find a solution and improves the overall performance of the system.

The change in emphasis toward testing has effected IMAGE so much that today its most successful application avoids generation altogether and uses only the testing feature to evaluate conventionally produced designs.

U-DESIGN has adopted some features of IMAGE's approach, but makes a number of departures so that testing can work in the midst of a design process rather than as an interuption or as a step tacked onto the end of the process. Care is taken so testing activity, primarily the machine's calculations, won't interfere with the designer. He doesn't

have to interupt his process to ask for a test, he doesn't have to wait for the results of a test, and he doesn't have to be interupted because of test results.

The approach used takes advantage of the fact that the computer is a "dedicated processor" - that is the computer is not time-shared and it normally sits idle while it waits for user requests. In U-DESIGN, test calculations are being done constantly as an invisible background process even while the computer is waiting and watching for user actions. Indications of the results appear unobtrusively on the user's display screen and he is not otherwise interrupted. Many times calculations are begun only to have the results thrown out and the test restarted because the user has changed something. But, when the user does want a test result, it will probably be ready for him.

Test reports should be easy to read and quick to decipher. Textual reports, that is those with words and numbers, often are not. Pictorial reports, those in terms of diagrams, usually are. Whenever possible test reports in U-DESIGN appear in the terms of the graphic representation the designer is working with. The planar graph diagram is a good example of a case where this works. Representing a spacial arrangement by its planar graph, means that conflicts among connections automatically stand out as points where lines cross. This identifies the existence of a problem,

its exact location, and the spaces involved. It is far more effective than any verbal report. The verbal alternative would be something like "the connection between A and B interferes with the connection between C and D."

With computer system support as is available at the Architecture Machine, it is possible to take advantage of the dynamic capabilities of modern computer display terminals, and use motion as well as static lines as a reporting medium. Seeing a space move is a perfect way of identifying a problem and perceiving a direction and magnitude for it.

C. DESIGN IN TERMS OF A HIERARCHY OF TASKS AND GRAPHIC REPRESENTATIONS

1. THE NEED FOR A SEGMENTED PROCESS

For the novice designer, end even the more experienced one, the task of juggling parameters to find a solution in terms of many different kinds of relationships simultaneously can be very difficult if not impossible. In practice, designers often start by considering their problem in its simplest diagrammatic forms and only once a satisfactory arrangement on that level is assured do they carry it to further elaboration.

A novice designer might fail because he tries to deal simultaneously with too many issues on too many levels. In a design-your-own-home experiment called "The Falco Experiment" at MIT (10), this inability became very apparent. The design efforts of the "designers" of the experiment were unsuccessful as long as they tried to deal with spaces as solids with areas. However, when they were shown how they could consider the problem as a planar graph, they readily identified those areas causing them difficulty and were able to adjust their designs accordingly. They needed someone to segment the process for them. It seems that finding the right representation of a problem is often half the battle of solving it.

U-DESIGN has been written to permit a segmented step-by-step approach. The value of this has been demonstrated by Yona Friedman's experience with novice designers in France. (11) An ability to deal with problems in terms of a segmented series of issues and a layered hierarchy of diagrams is a major feature of this thesis. Details of the approach are described below. The approach offers obvious advantages. It allows a designer to deal with a problem at the level best suited to his grasp of the problem and the current specificity of his solution. And it allows him to quickly progress through or skip stages so it need not slow him down.

2. DIAGRAMS CAN SERVE AS TESTING DEVICES

To deal with a particular set of issues one needs an appropriate tool. In the field of architectural design, it is apt to be a graphic one. One needs a facility to deal with issues in terms appropriate for those issues. For problems of connectivity, the planar graph diagram is ideal. To design for the subtle issues of the exact nature of a connection existing between two spaces, a far more refined diagram is required. An architect is used to working with a wide variety of diagram types; even his finished floor plan is a diagram. U-DESIGN envisions a series of diagramatic hierarchies that starts with planar graphs, progresses to

rectangular shapes and then to the refinements of walls, doors, windows, furniture, materials, etc.

Effective diagrams must clearly represent the issues being addressed. They must indicate the form of the solution and at the same time must be consistant with the information being dealt with. They need not present extra information for which the designer is not ready, but at the same time, this doesn't mean that a designer should be prevented from thinking about issues not shown by the diagram. In practice, while arranging spaces in a sparce representation a user does think about factors or more complexity and projects them onto his diagram. Usage of U-DESIGN and YONA has shown that if he's working with a planar graph diagram he maintains some notion of size and determines the separation between nodes accordingly. He thinks about building envelopes, site features and many other issues as well.

3. DIAGRAMS USED IN U-DESIGN

Two diagram types have been implemented for this project. They are to be viewed as merely two steps in a hierarchy of ever increasingly complex diagrams that will eventually lead to a finished design. The planar graph type diagram has been mentioned above. It has been used as an example to show how a diagram can provide a medium for working toward a solution, indicate a form for a solution,

as well as its topology, and test that solution - i.e. indicate where problems are occuring. This kind of diagram has been chosen as one of the two types developed for U-DESIGN.

For a second type, a diagram with areas and shapes was required. The diagram of rectangular areas fills this need. In addition, there exists a framework for dealing with the many other relationships between spaces besides connectivity in this representation. It is IMAGE. The diagram indicates overlay and size-shape problems, but unfortunately sheds little light on other violated relationships. To introduce a vocabulary of lines representing relationships as is done in the planar graph case would create a very confused diagram. There are too many types of relationships possible to represent them all by lines. Also one of the chief advantages of the lines in the first place is lost. Two crossed connectivity lines meant something; two crossed distance lines means nothing. This is not to totally discount the use of lines in conjunction with the diagram, rather it is to say something more is needed.

This is where the dynamic display capabilities of the Architecture Machine can be well utilized. The IMAGE generation procedure provides a method for determining errors in a space's position and moving the space incrementally toward a better position. If IMAGE's changes to all of the

spaces in a problem arrangement are continuously displayed on the computer screen, the user will see moving spaces. The signifigance of this is that the same diagram that gives a picture of the static state of a solution will also provide clues to problems and inconsistancies when they occur. If the user sees a space moving toward or away from another, he'll know that some distance or adjacency relationship is not satisfied.

III.

ORIGINS OF THE SYSTEM - DIAGRAMS USED

Out of the hierarchy of diagrammatic languages appropriate for design, only two have been implemented. These correspond to the diagrams of YONA and IMAGE. They are called YONA-NODES and IMAGE-RECTANGLES respectively. While no parts of either could be applied directly, important methods and concepts from both have been adopted. It is expected that higher level diagrams will be added in the future, but for this first demonstration attempt, only those that could be quickly implemented were used.

A. IMAGE

IMAGE (12) represents spaces by rectangles in its pictorial displays. For most purposes, these shapes are better viewed as rectangularized "bubbles", since they constitute approximations of space boundaries rather than exact intentions about wall elements. The system has a protocol for combining several of its rectangles to model irregularly shaped spaces. It also has facilities for viewing spaces as activity settings rather than as rooms to model situations where functions overlap in space. The last two special features have not been implemented, but the basic rectangular representation of spaces has been adopted.

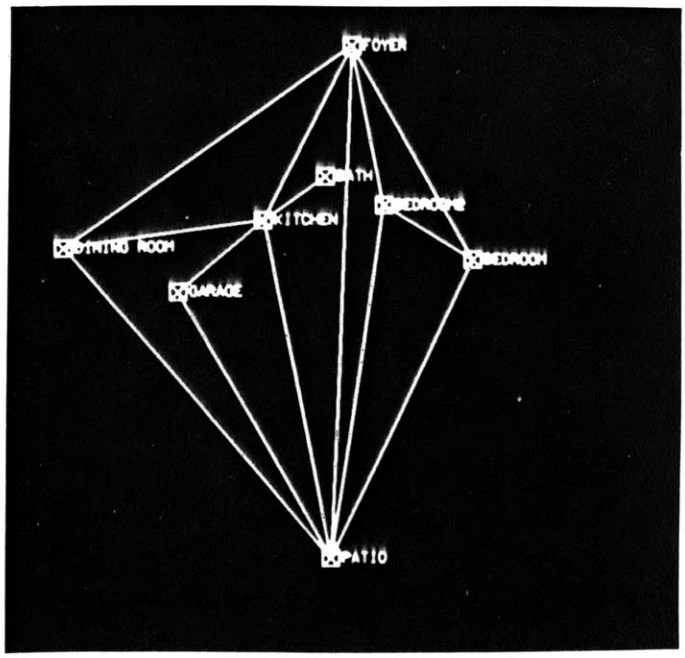

Figure iii-1

A planar graph representation. Spaces are dimensionless nodes and the connections between them appear as lines. A graph that can be drawn with none of these lines crossing is "planar"

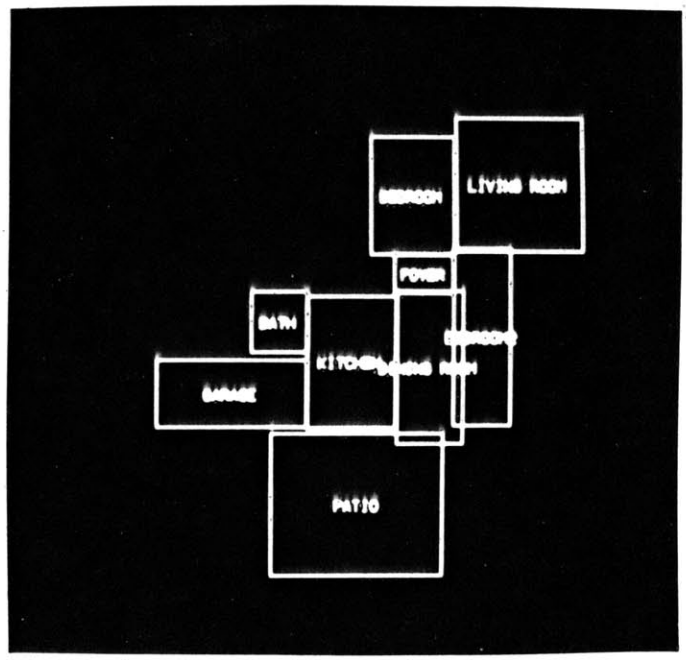

Figure iii-2

An IMAGE-type representation of a design. Spaces are modeled as rectangles which move and deform in response to unsatisfied relationships and conflicting requirements.

IMAGE has a vocabulary of relationships, alternately called constraints or links, which the designer uses to indicate his intentions about a space. It extends to just about every simple geometrical relationship between spaces. With judicious combinations of the simpler relationships, many very complex intentions can be modeled. For example, one can model a view through a window by creating a window element, attaching it to a wall and using the visual connection relationship to link the window to the object viewed.

U-DESIGN has adopted IMAGE's method of dealing with relationships. A part of IMAGE's rather extensive vocabulary in this area has been included, specifically: AREA, RATIO, FIXED/RIGID, OVERLAPPABLE, DISTANCE, NEAR, FAR, KEEPOUT, and ADJACENCY (now called LINK). They are described later in this document and more extensively in many of the IMAGE publications. The ability to expand this list has been built into the system.

IMAGE's method of problem solving is designed to find solutions to problems which are "over constrained"; that is in situations where more intentions have been indicated than can possibly be satisfied. It is supposed to find a best compromise in such a case. The mathematical framework of its problem solving algorithm handles different relationship

types in a trade-off method. Opposing tendencies from two relationships are averaged. A space that wants to be in two opposite places at the same time moves toward a middle point. When this process is repeated over and over, with all the spaces moving in this manner, the overall arrangement gradually reaches a solution - at least that is the hope.

As has been mentioned above, this is often not the case. The difficulty can be understood in terms of an analogy of hill climbing. The solution space is a field of hills and the highest hill represents the best solution. The computer analyses the situation in its own neighborhood to determine which way is up and moves that way. The trouble is that the machine may have climbed the wrong hill and once at the top motion in any direction looks downward to it. Thus it is apt not to find the highest peak. In other words, it misses the optimal solution and gets hung up on a sub-optimal one.

U-DESIGN has kept this generation procedure, but changed its purpose. Now its role is to make minor corrections to arrangements and to signal major problems as they occur. The program has been written to work as a background process and the IMAGE features that made the system leap from one sub-optimum to another have been removed.

The logic now is much simpler though the descriptions of it sound about the same. Spaces are considered one at

a time, simply going down the list of them all. All of the intentions for one particular space are examined. For each relationship involving that space a calculation of an error amount and needed correction is made. For most relationships there will be no error, but the errors that do exist are accumulated and an average of them all is determined. Since an error is reported as a displacement or distortion the space must undergo in order to correct the error, the accumulated average can be interpreted as being the best compromise motion for that space. This compromise motion becomes the amount the space is moved in one cycle. Since the motion is displayed on the computer screen and since each space is checked this way over and over again, they all will all gradually move about toward their own local optima. In practice this means that a space will do either of two things: 1) move to a satisfactory position and stop - this will mean all intentions are satisfied, or 2) move around and eventually start to oscillate - this means that it trying to ballance conflicting intentions and is stuck at some sub-optimum.

As opposed to the IMAGE situation, in U-DESIGN the motion is shown to the user and the process allows him to participate simultaneously in the rearranging when he chooses. If he sees a way to solve a problem that the machine missed or wants to try a new idea, he can try rearrangements himself.

If he moves a space and introduces a new violation, the machine will remind him of his oversight. The new position will mean that other previously stationary spaces related to it are now in error and will therefore move to correct the situation. Through the combination of observing what moves when he does nothing, and watching what happens when he makes a tentative change, the user can determine where a problem lies and even get some idea as to how to correct it. Major topological problems will have been resolved in the vocabulary of the planar graph diagram before this IMAGE-like phase has been started, so the problems will not be as tough as they might have been.

The generation motion serves as a fast immediate testing device. As a secondary testing resource, IMAGE's regular testing routine will be used. If the motion alone is not clear enough he can try it.

B. YONA

The second of the important antecedents for U-DESIGN is one just in its beginning stages of development. It is called YONA after the man responsible for its basic philosophy, Friedman. (13) Its aim is to guide the "man off the street" through the process of designing his own home. It has him progress in a controlled step-by-step fashion considering one issue at a time. First he lists the spaces

of his house, second he specifies the connections he wants between them, third either he or the machine finds an arrangement of the linkage graph that has no crossing lines (i.e. is planar) and gives each space the desired orientation or exposure, and forth, in a manner yet to be determined, the nodes of this graph are given size and shape. Included in this scheme are a large number of testing routines to check issues such as planarity, exposure, or orientation possibilities. Just how the system will operate with these routines is as yet undetermined.

YONA has made two contributions to U-DESIGN. One, it has demonstrated the unique value of using a linkage graph as a design diagram. Two, it has demonstrated that there is value to the segmented approach toward helping someone design.

The planar graph method of working was taken as the first stage implemented in the hierarchy of diagramatic stages. Two manipulation and testing routines developed for YONA were adapted for U-DESIGN and it is expected that more of YONA's test routines will be incorporated as they are developed.

IV.

THE SYSTEM

A. GRAPHICS CONVENTIONS

A few words about U-DESIGN's operating conventions are in order. Most light pen actions are initiated by pointing at an element on the screen - specifically, aiming the pen and pressing a button on the pen barrel. Often times it is an item in a "menu". (A "menu" is a list of options displayed on the screen from which the user may choose.) Other times it is a space. Spaces can be moved by pointing at them and "dragging" them around; as the pen moves the spaces move. In the diagram modes that consider shape, walls can be moved by pointing at them and dragging them to new positions. At any time commands from the keyboard can be given.

B. OPERATING ENVIRONMENT

The user starts the program by typing "UDESIGN". Directions and suggestions for using it will appear on the screen. He'll start with a completely blank problem description, but will have the option of referencing a previously begun problem.

He'll have a choice of several graphic functions. These fall into two categories, those in terms of planar graphs, YONA-NODES, and those in terms of the IMAGE-type rectangles, IMAGE-RECTANGLES. They include the following:

	make spaces
YONA-NODES	link spaces
	move spaces
IMAGE-RECTANGLES	make spaces
	arrange spaces

These options are available in a "menu". Picking one prepares the computer to interpret the user's subsequent light pen actions according to the function chosen. It also prepares the appropriate background testing routines.

When a function is chosen, the diagram appropriate for it will be displayed on the screen and a new sub-menu of additional options will appear. The user can work with that diagram, use its sub-menu, and still at any time choose new functions from the main menu.

For every function, there is a set of testing routines which monitor the state of the problem and provide warnings about future road blocks or dead-ends towards which the user may be heading. The testing routines operate even while the computer is watching the user. From the user's point of view, the computer is always ready for his commands.

C. KEYBOARD FUNCTIONS

There are several keyboard functions the user has available to him. Some duplicate functions from the menus but others are only available from the keyboard. They can be issued at any time, irrespective of the menu function or diagram type on the screen.

Intentions between pairs of spaces:

LINK	connect two spaces.
NEAR	don't exceed the maximum separation.
FAR	don't come closer than the a minimum separation.
DIST	indicate a specific separation.
KEEPO	disallow overlap.
LAP	allow overlap.
REMOVE	preceeding any of the above removes the specified relations.

Intentions about a single space:

AREA	specify an area in square feet.
RATIO	specify a maximum elongation; long side must be less than X percent of short side.

FIX	prevent any computer-generated repositioning.
MOVE	turn off FIX.
DELETE	remove a space.
MAKE	make a new space.

Saving and retrieving descriptions from permanent storage:

SAVE	save (a file name is optional).
GET	retrieve (ditto).

The formats of these commands can be indicated by a few examples. "NEAR KITCHEN BATH GARAGE 20" places the kitchen "near" the bath and the garage. "Near" refers to the closest face-to-face separation between two spaces, and the "20" means that distance must be no greater then twenty feet. "REMOVE NEAR KITCHEN BATH GARAGE" removes the previous two relationships. "RATIO BATH KITCHEN 150" restricts the bath and kitchen to being no more elongated than 150 percent (3-to-2) in either direction.

D. THE MAKE SPACES FUNCTION

The proceedure for making spaces is identical for both representations. If the user indicates a problem type, for example "housing", the program, will offer a sub-menu list of possible spaces, providing that type has been previously defined. For the housing example, the list of suggested

spaces includes rooms commonly mentioned in a house. The user is free to alter this list if he chooses.

Picking spaces from a premade list provides several advantages. It eliminates a lot of typing, helps avoid spelling mistakes and suggests a standard vocabulary which ir the future may allow the machine to infer some of the designer's intentions from the space names.

The user makes a space by pointing at the name in the sub-menu and dragging it into the center area of the screen. The space is created where the name is dropped. Spaces can stacked up or positioned to reflect preliminary ideas. One deletes a space by pointing at it.

E. THE LINK SPACES FUNCTION:

Spaces are linked in a two step process. First the user "activates" a space by pointing at it thus making it blink. Then he points at any other spaces he wants to link to the activated space. He can activate a different space if he first "deactivates" the active one. This he does by pointing. He can remove links by the same process. The motions that make a link delete one that is already there.

While the user is adding and deleting links in this way, the machine is also working. It is calculating whether or

not the graph of the user's spaces and links is planar. This is irrespective of whether or not the diagram appears in a planar representation on the screen (that is with no lines crossing). A report of the planarity status is put on the screen, but the user is not interrupted. Planarity of the graph does not guarantee that the diagram can be translated into an acceptable building design, but non-planarity means it will be impossible.

F. THE MOVE SPACES FUNCTION

Spaces can be rearranged with the "move spaces" function. This function works in terms of the planar graph representation. The user's main task here is to find a satisfactory topological arangement for his problem; that is an arrangement which has no crossing lines. He will probably, but won't have to, arrange spaces in response to other goals as well.

The primary repositioning action consists of pointing at spaces with the light pen and dragging them to new positions. There are several options provided to help with large-scale transformations that move several spaces at once.

MIRROR X	reflection about x axis.
MIRROR Y	reflection about y axis.
SPREAD	move apart.

SHRINK	move together.
ROTATE +	rotate counter clockwise.
ROTATE -	rotate clockwise.

These operations can apply to the entire diagram or to just a part of it. To operate on just part, draw a "circle" around that part and then point at the sub-menu item you want. A "circle" in this case is actually any closed curve no matter how convoluted. The user can move a group of spaces just as he would a single space if he first draws a circle around the group. All of the circled spaces will move as a single unit.

G. THE ARRANGE FUNCTION

The user can advance from the limited terms of the planar graph diagram to a representation involving size and shape. He does this by picking the "arrange" function out of the menu list. His spaces are then presented as rectangular areas. One space is represented by one rectangle. The user can move spaces by pointing at their names and dragging them as he did with the "move spaces" function. In addition, since he is now dealing with shape, he can change the shape of a space. This he does by pointing at an edge of the space and dragging it. The opposite edge remains stationary, so the space changes shape. He can drag two edges at once by pointing at a corner. In this case, the opposite corner stays stationary.

Because the light pen has a very blunt point it is sometimes difficult to point accurately enough to see the edge of a small space without seeing the name inside as well. Two special sub-menu features come with this function to help with this problem. "MOVE ONLY" disables the reshape feature so one won't get into reshape mode when trying to move a space. And "RESHAPE ONLY" works in the corrsponding way.

One additional sub-menu option that has proven necessary is the ability to look back at the linkage diagram (planar graph) without leaving the "arrange" function. There is an item called "SEE LINKS" for this purpose.

H. BACKGROUND ACTIVITY WITH ARRANGE

Meanwhile, the computer is not inactive. It takes a very active role serving two purposes. Firstl'it continually makes fine adjustments to the design in accordance with the user's expressed intentions. This frees the designer from the need to be particularly precise about his positioning of elements. And second, it uses the same facility that determines when and where to make fine adjustments to move grossly misplaced spaces. This motion may or may not lead to a satisfactory arrangement by itself, but its main purpose is to report errors in the arrangement. In response to the motion, the designer can find a more satisfactory location

for a space. If he moves a space forgetting about other spaces that need to be near it, he will be reminded of his oversight by seeing those other spaces move to follow the first.

I. AN EXAMPLE WITH PICTURES

The following are photographs taken in the coarse of a design experiment with U-DESIGN.

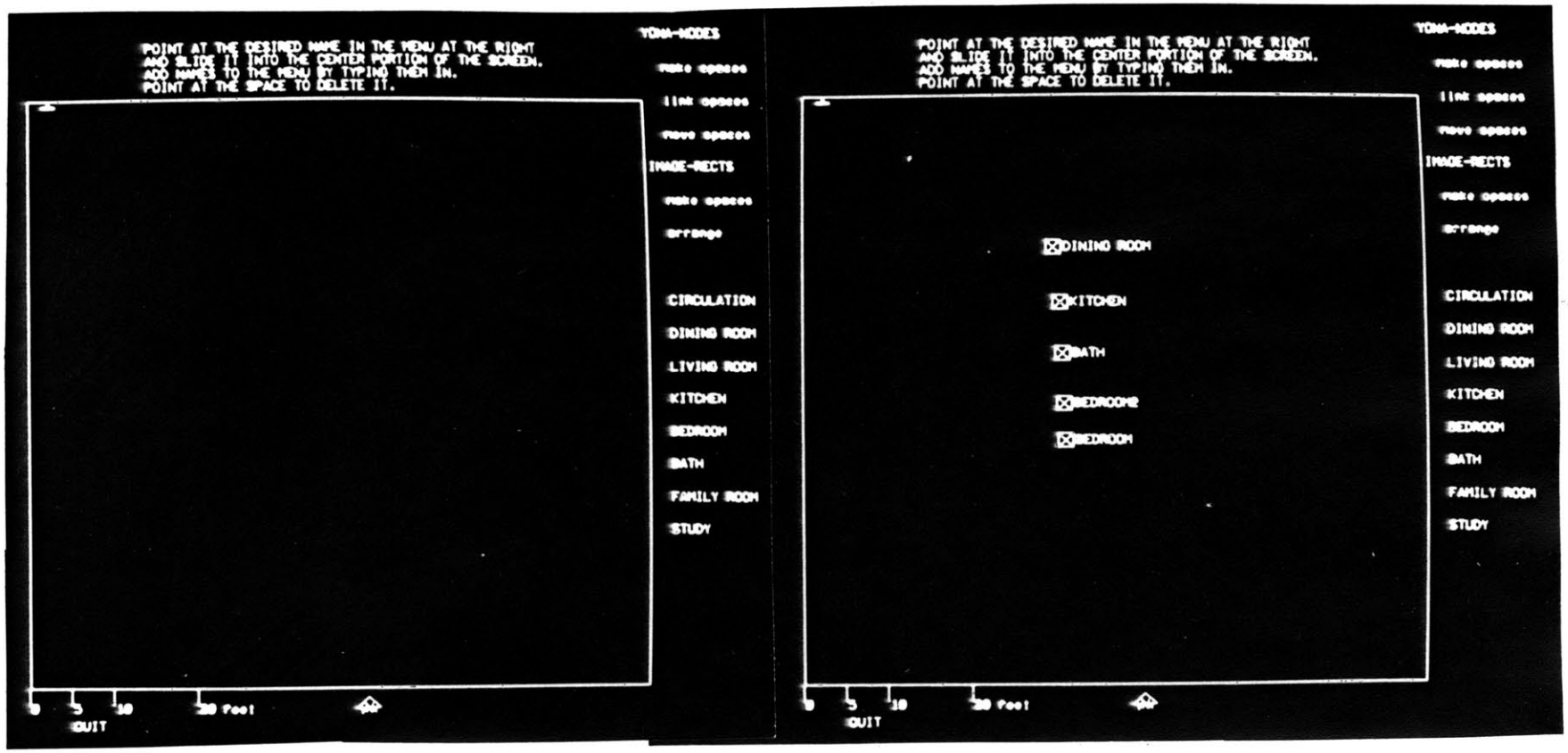

Figure iv-1

The computer display at the start of a design session. The user has already indicated that he is designing a house and the computer has found a previously defined list of rooms in a house and presented those as a "menu" of possible spaces from which the user can choose.

Figure iv-2

The user has created five spaces by pointing at the name in the "menu" and moving his pen to a position inside the framed part of the display screen. At this point he has simply stacked the spaces without considering position.

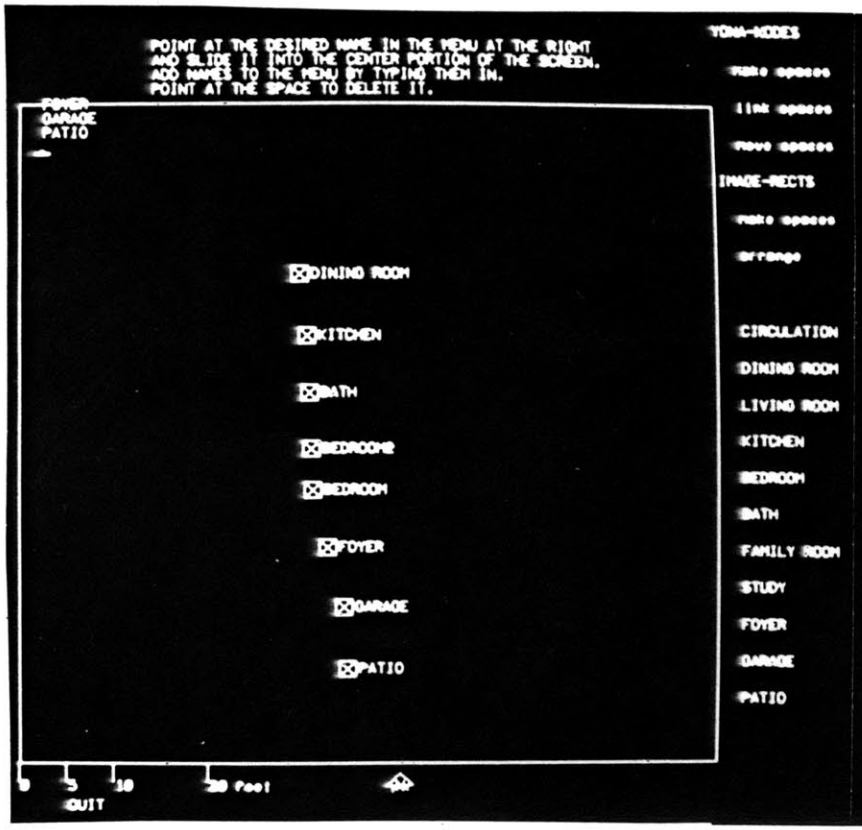

Figure iv-3

Three spaces he wanted to create were not in the "menu". He has added these by typing the names himself and creating the spaces in the usual way.

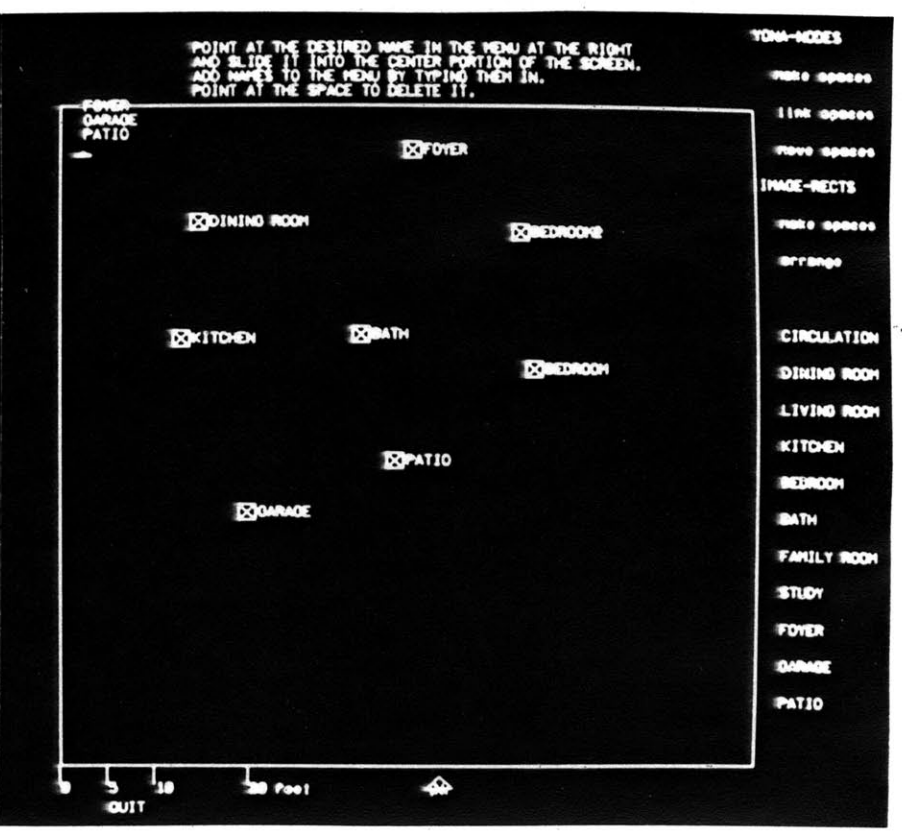

Figure iv-4

He has started to consider the positions of his spaces relative to one another. However, there is not enough information at this stage to make serious decisions.

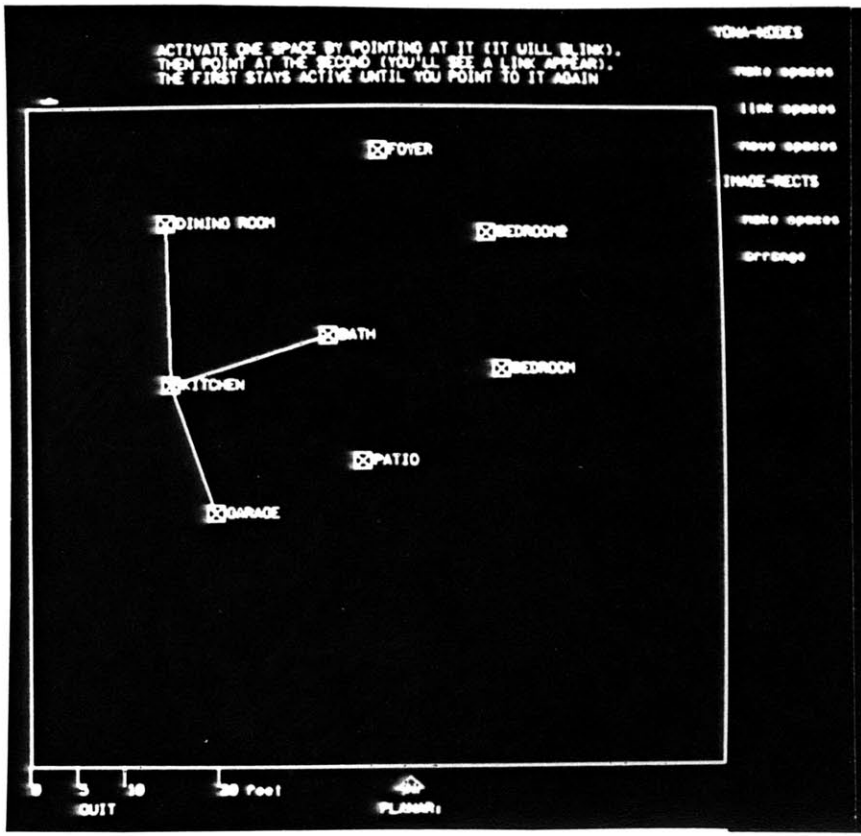

Figure iv-5

The user has indicated he intends to create links between spaces and the system has prepared the light-pen function to do this task. At this point, he has just activated "kitchen" and has linked several spaces to it.

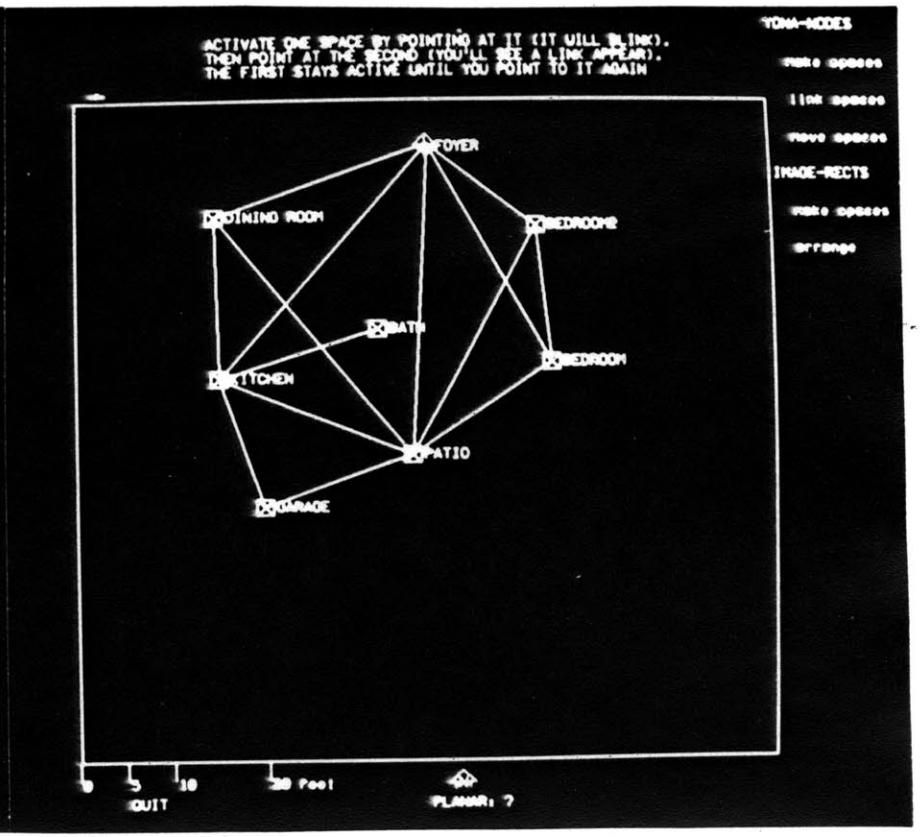

Figure iv-6

All desired links have been made. The computer has been saying if the scheme is planar or nor during this process. Now it has just started a new calculation because a link had just been added. It will display the message "planar ?" until the calculation finishes.

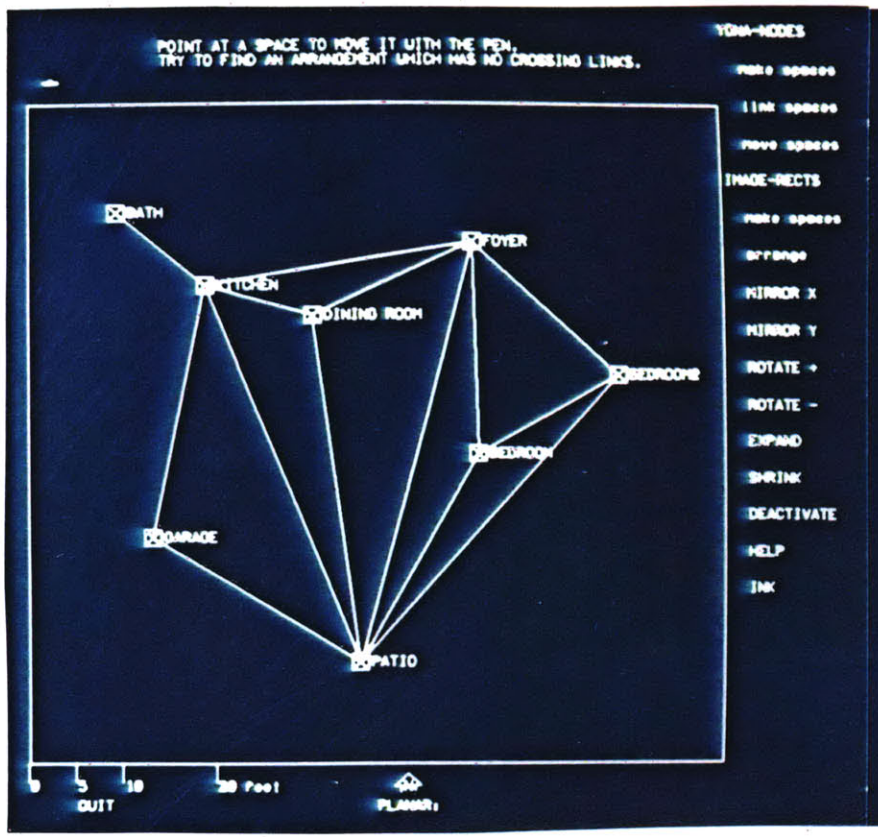

Figure iv-7

The graph is planar and the user has found a planar topology for his problem. He is now experimenting with several alternatives. He'll find that there will be no way to avoid having spaces that are totally interior. In this case it is the bedroom and dining room that are inside.

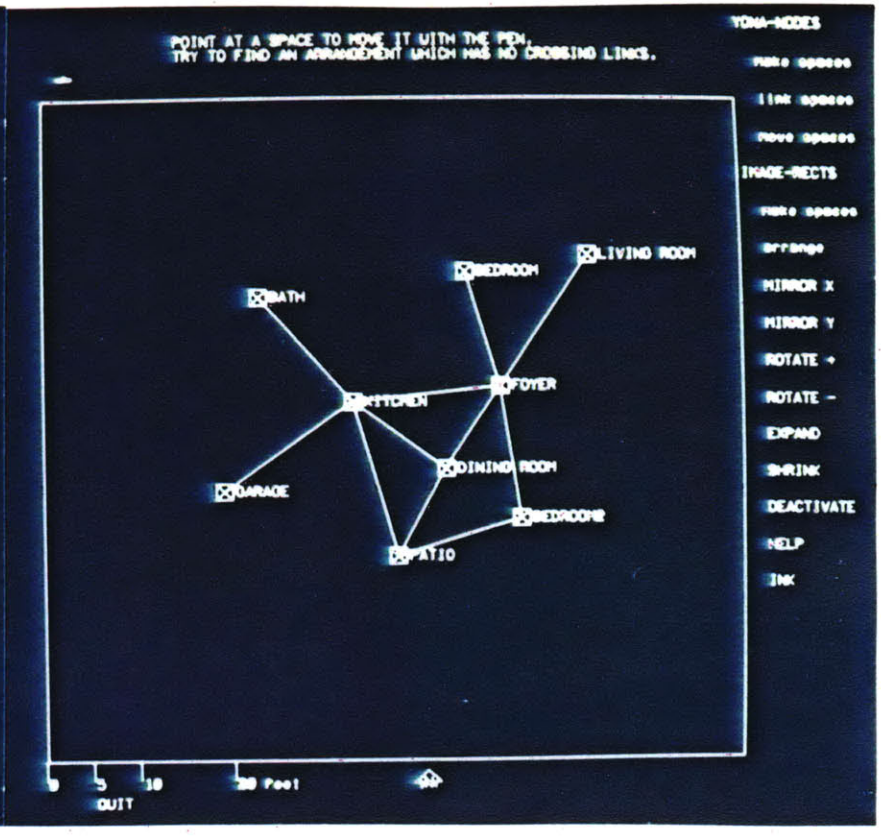

Figure iv-8

He has decided to delete some of the connections he previously wanted and has found a satisfactory topology. It is still necessary to have an interior space, but that is acceptable. Note, that he has added a "living room," something he forgot earlier.

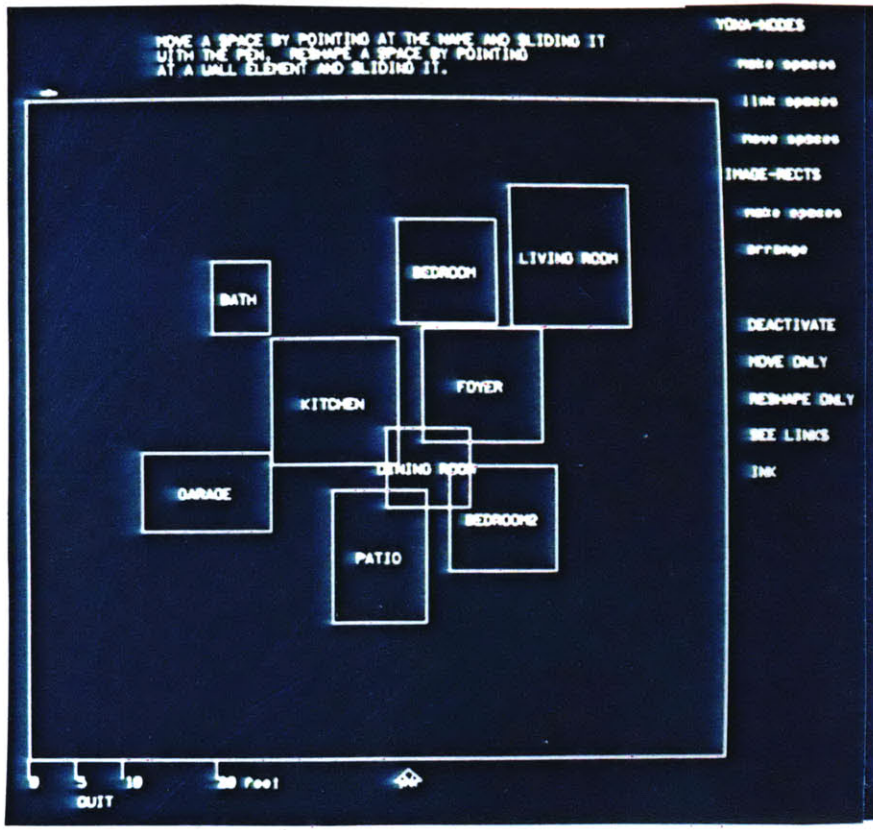

Figure iv-9

The user has asked for a representation in terms of rectangles (IMAGE-RECTS) and is ready to rearrange the spaces himself. While he ponders what to do, the computer is solving some of theminor problems itself.

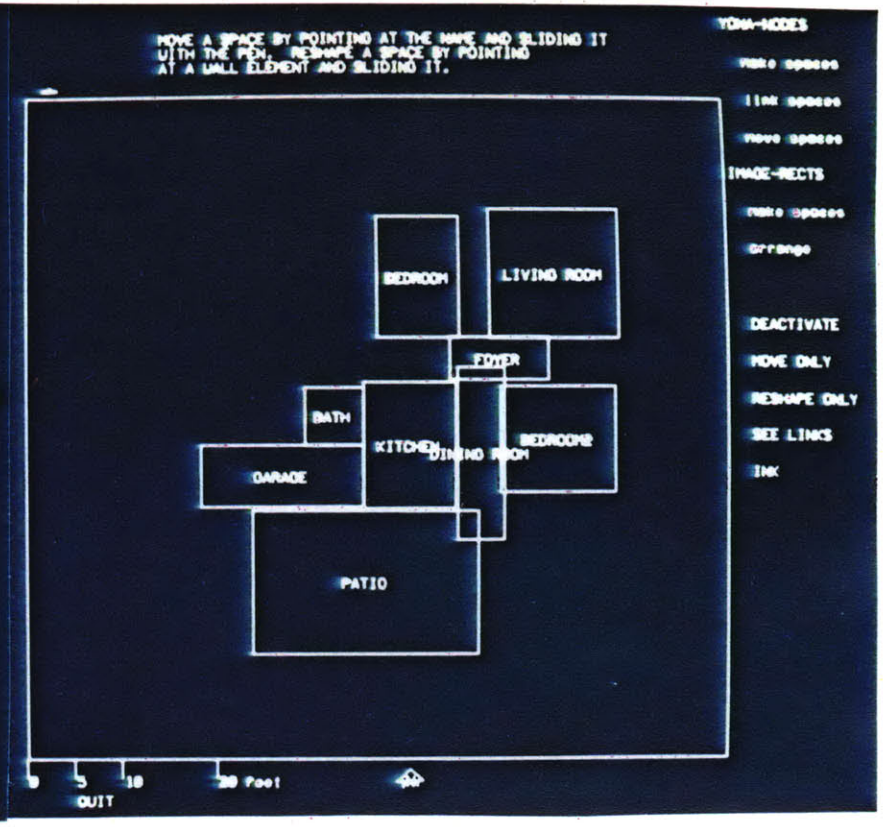

Figure iv-10

The computer has resolved some minor problems. The user has been working on the arrangement of the spaces around the kitchen. There is a problem because the patio and bedroom2 keep trying to move together and sre squeezing the dining room.

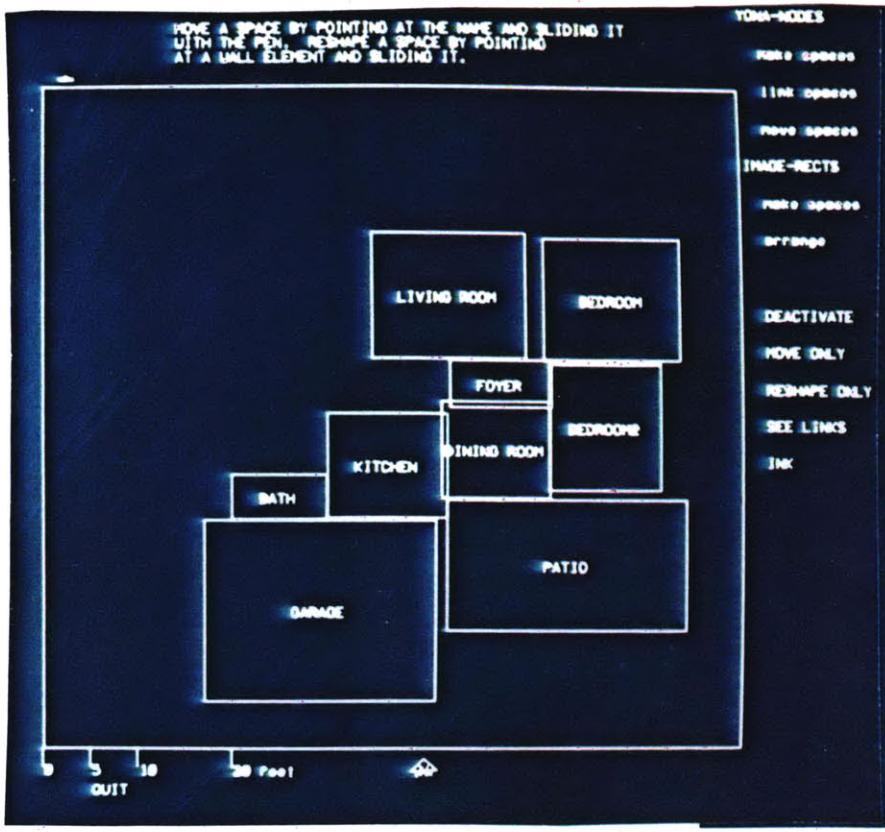

Figure iv-11

The user has corrected the problems around the dining room and has adjusted a few shapes and positions. He's decided to swap the positions of the living room and bedroom. This manages to resalvage the connection between the two bedrooms which he had previously abandoned.

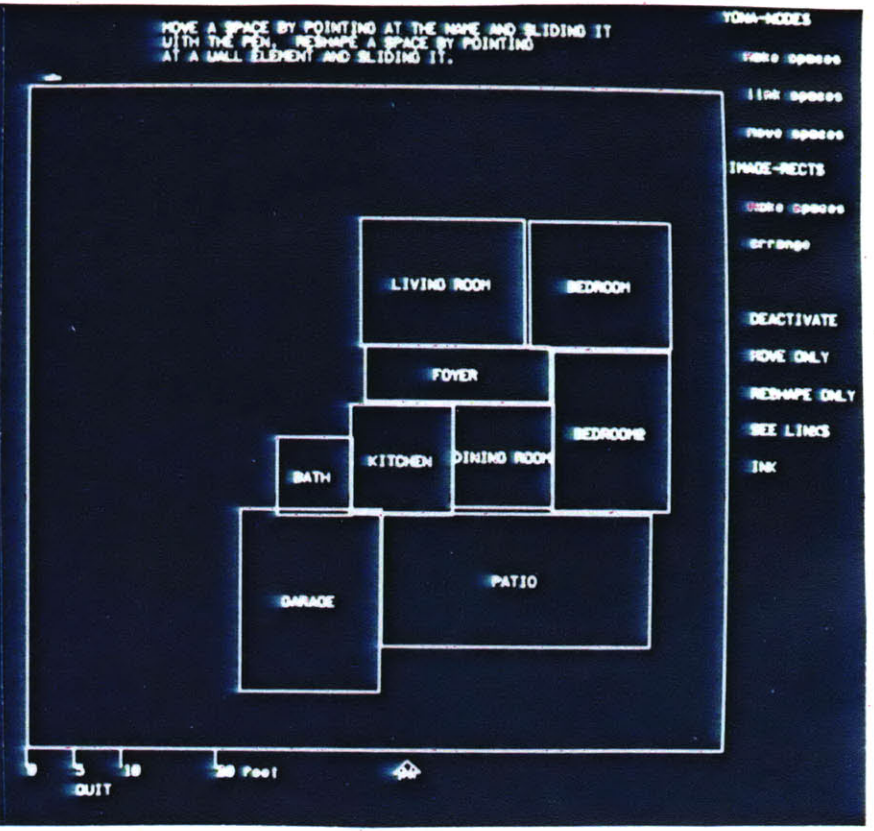

Figure iv-12

The are just a few minor adjustments to be made. (the computer will be able to do some.) Otherwise the space-planning stage of this design process is finished. The next step would be to consider more details like positions of walls, doors, and windows.

U-DESIGN operates on the Architecture Machine computers. Specifically an INTERDATA small-size computer and an IMLAC mini graphics computer used as a display terminal. The user communicates with the INTERDATA through the IMLAC's typewriter keyboard, light pen, and display screen. The design in some diagramatic form appears on the display screen and the user modifies it with the light pen. Specifics about the hardware and the Architecture Machine system are available in the references.

The key to the whole system is a program called DISPATCH. Its role is to constantly watch for any user actions - typed commands or light pen activities and call the proper interpreter functions to process those actions. These functions are called "foreground" routines in this system because they are always ready to respond to the user and are the only routines he interacts with directly. Whenever DISPATCH sees no user action it calls a "background" test routine of some sort. They are always there and running, the user is aware of their presence only when they have finished a calculation.

When DISPATCH calls a foreground routine, it indicates what kind of activity was observed - light pen pointing, drawing, or typed command. The foreground routine must interpret this activity and determine what further

V.

SYSTEM ORGANIZATION

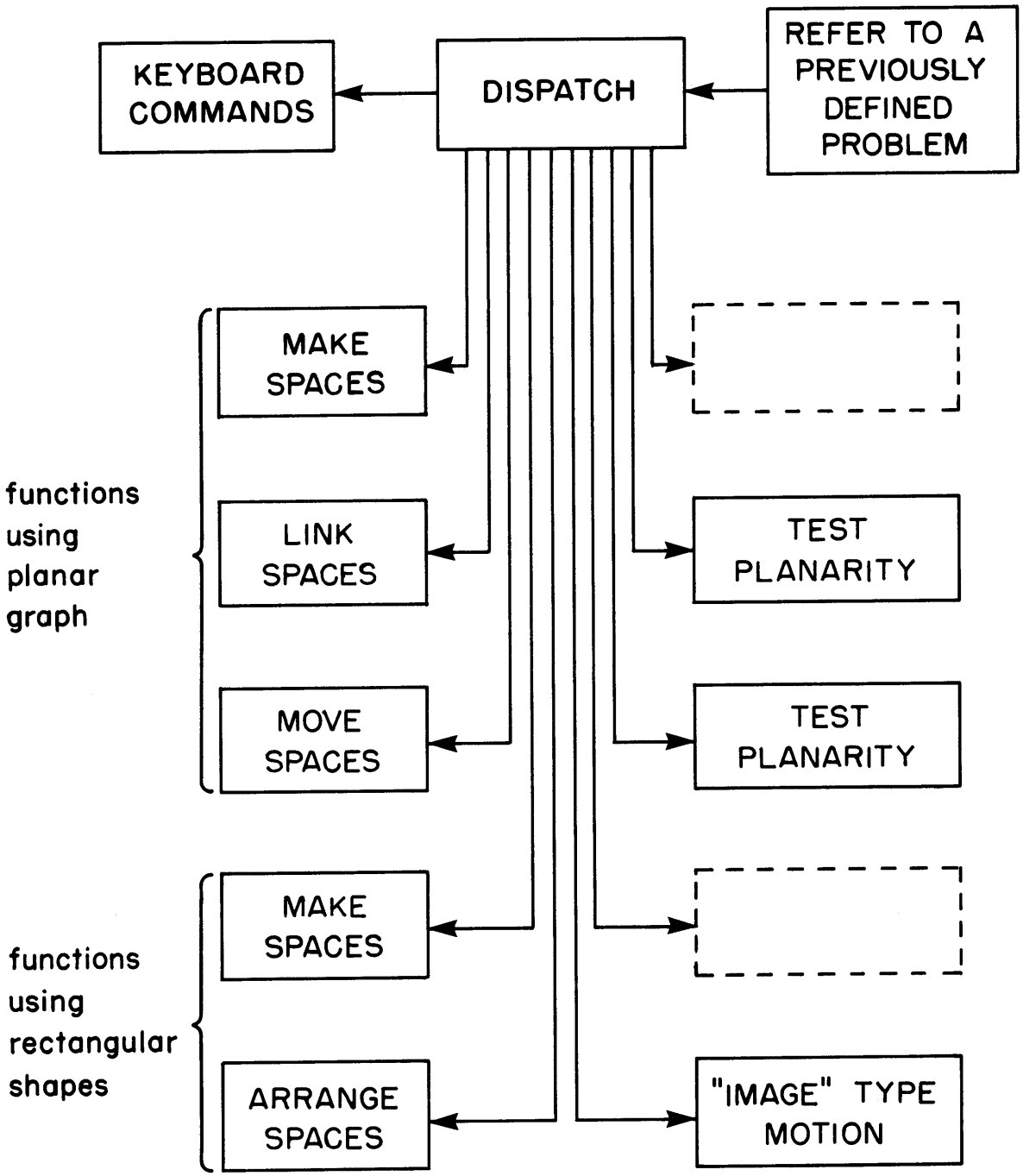

Figure v-1

Schematic diagram showing the relationship between the central dispatch program and all the functions it calls.

observations and calculations it has to make. The various interpreter functions must determine for themselves what has to be done in their own special cases. Their actions were described in the "operating scenario" section of this paper. For further details the reader is referred to the listings of the programs themselves (*).

To allow for a background process on a computer system that doesn't support interrupts or multiprocessing, the background routines themselves have to be written so they are constantly interrupting themselves. In most cases, this is achieved by doing computations in a cyclical iterative process and stopping the process after each iteration, saving only the information needed to reenter the calculation where it left off. Each time, DISPATCH tests for user activity and goes back to the background routine unless there was some activity.

DISPATCH also determines which foreground and background functions the user needs by watching for his selections from the menu list of such functions. It makes sure the proper routines are set up and initialized in the computer's core memory. In a small computer,an overlaying of routines is needed. DISPATCH does this.

A final task of the DISPATCH routine is to watch for indications from the foreground routines that mean the

background process, with its possibly incomplete calculations may have be aborted and restarted. It passes these indications to the background routines and they actually determine if reinitialization of the calculations is necessary. For example, if a space has been moved, the background planarity checking routine doesn't need to be concerned, but if a new connection between spaces was added, the routine might have to be restarted. It would not have to be restarted if the linkage graph was already non-planar, because adding a link to a non-planar graph only creates a "more" non-planar graph. But if the graph was planar, or the calculation hadn't finished, it would have to be restarted to incorporate the new information.

VI.

CONCLUSIONS AND FUTURE DIRECTIONS

U-DESIGN is not yet a complete architectural design tool. To be useful, a tool will have to assist the user not only with topology and space planning, but also with the more specific and detailed issues that come beyond. It will have to help him make decisions about details of the ways things are put together; the exact nature of a connection between two spaces or the exact nature of a structural connection.

There is an obvious next layer to be added to the hierarchy of representational diagrams. That is a representation that deals with walls and the other elements out of which buildings are maid. The system needs a routine to transform a diagram of rectangles into one of walls with actual thickness. It needs a simple way to represent doors and windows and a way to manipulate these elements.

The present system not only fails to go beyond the space planning stage, but is incomplete in the two stages it does cover. Lessons about the usefulness of U-DESIGN's approach have to be inferred fron a demonstration that is only the skeleton of a system. But some lessons have been learned. Some concern the Architecture Machine's hardware and systems support, but the more interesting ones have to do with insights concerning needs for future development and flaws in the present system. Hopefully, the biggest lesson learned

will be that the system is a promising tool and that it should be developed more fully.

A. LESSONS CONCERNING HARDWARE AND SYSTEMS

1.) The speed of operation and the quickness of the INTERDATA's responses can be improved if more of the interactive-graphics tasks are handled directly by the IMLAC mini computer. This will leave the main processor and memory free to concentrate on testing and evaluative tasks. In the present system, the main processor has to both control the graphics interaction and do all the test calculations. The mini computer is only a "slave" drawing lines from a display buffer.

2.) The present light pen is a difficult device to use. Its sensor is about as broad as one's finger and therefore cannot distinguish between elements that are much closer than a finger width. Frequently, one points at one element , but the pen senses another that is next to it. A second difficulty with the light pen is that it has to be held perpendicular to the display screen and most people want to hold it at an angle. What is needed is a pen with a much finer point and one that can be used at varying angles.

3.) Better facilities for handling overlays are needed. Many of the possibliities envisioned will not be possible unless a fast efficient overlay system is developed for U-DESIGN.

B. AREAS FOR FUTURE DEVELOPMENT

Obviously "more" and "better" can always apply to

routines in any portion of the system, but a few particular areas stand out.

1.) the planar graph has been a successful first step in the process, but it can be developed more fully. Additional issues about the graph can be considered. Orientation (that is facing in the proper compas direction) and enclosure (or lack of exposure to the outside) are two issues which can be examined in terms of planar graphs. There are two kinds of tests for each of these issues: a) is it possible to find a solution and b) if so, is the present arrangement a solution. Tests of all these kinds are needed. Right now, for planarity the is a test of type A but none of type B.

2.) Additional relationship types need to be added to the system's vocabulary. The methods for dealing with these are known from IMAGE. (14) The most important are:

a) SHARED-WALL, to be able to specify that there be a particular amount common shared wall between two spaces. This is important in providing for doorways.

b) ENCLOSURE, to specify that one space must lie inside another.

c) VISUAL-ACCESS, to specify that one space must "see" another, i.e. that no third space come between them unless it is "transparent"

d) WEIGHT, to be able to assign different relative importance to different relationships.

e) ALIGNMENT, to require that one space line up with another along a particular axis.

f) RELATIVE-POSITION, to require that one space maintain a constant position relative to another.

3.) in many real life applications, relationships are conditional. A room is near one exit or another for example. To model any serious problem, a facility for dealing with conditional intentions is necessary.

4.) The system is very helpful for problems that have no rigid boundary requirements, but it isn't much help in dealing with problems that do have tight boundary conditions or other absolute constraints. Some better aide for "packing" problems is needed. Also some better way of dealiing with absolute constraints, as opposed to the present ones which can be violated, is needed.

5.) Better facilities for indicating intentions about relationships and attributes are needed. The "link spaces" graphic function is rigid in the formats it accepts. And there is no graphic way to indicate intentions other than connectivity.

6.) Often, the motion cues alone are not sufficient to identify a problem. A program similar to IMAGE's TESTER/RANKER is needed to produce verbal reports of the errors in a design for use in these situations. It should work as a background routine, constantly

displaying an updated report of the spaces and relationship are the worst violated.

7.) Some method for dealing with shapes other than rectangles is needed. IMAGE's idea was to build spaces from combinations of rectangles. A better facility would allow polygons of any shape.

REFERENCE BIBLIOGRAPHY

Alexander, C.
NOTES ON THE SYNTHESIS OF FORM
Cambridge. 1964

Archictecture Machine Group
Artificial Intelligence and Inference Making in Computer Aids to design
Proposal to the National Science Foundation 1974 (1)

Eastman, Charles
"Toward a Theory of Automated Design"
Institute of Physical planning, Carnegie Mellon University, Pittsburg

Foz, Adel
"Some Observations of Designer Behavior in the Parti"
Unpublished thesis, MIT, 1972 (8)

Friedman, Yona
Discusions with the Architecture Machine Group at MIT
1974 - 1975 (3) (11) (13) (14)

Johnson, T.E., Weinzapfel, Guy, et al
"IMAGE: an Interactive Graphics-Based Computer System for Multi- constrained Spacial Synthesis", MIT 1970 (2) (6) (12)

Johnson, T. E. and Weinzapfel, Guy
"Computer Assisted Space Synthesis under Geometric Constraints"
MIT, 1973

Miller, William
"Bibliography: Computer Aided Space Planning"
DMG Newsletter, 1971 (4) (5)

Mitchel, William J.
"Techniques of automated design in architecture: a survey and evaluation", UCLA, 1974 (4)

Negroponte, N. and Groisser, Leon
"Urban 5: An on-line Urban Design Partner"
IBM Report 320-2012

Porter, W. L.
"The Development of Discourse: A Language fo Computer Assisted City Design"
Unpublished Phd Thesis, MIT, 1969

REFERENCE BIBLIOGRAPHY (continued)

Purcell, Patrick
"The Computer as an Aid for the Architect"
The symposium of Interactive Graphics, Delft, 1970

Weinzapfel, Guy and Johnson, T.
"The IMAGE System and Its role in Design"
MIT 1973 (7)

Weinzapfel, Guy
"The Function of Testing During Architectural Design"
Unpublished Thesis, MIT, 1971 (9)

Weinzapfel, Guy
"Following the Yellow Brick Road" in
REFLECTIONS ON COMPUTER AIDS TO DESIGN AND ARCHITECTURE,
Negroponte, 1974

Weinzapfel, Guy
"The Falco Experiment"
Architecture Machine Group report 75-203, MIT 1975 (10)

Wood, J.
"The Design of a Telephone Engineering Centre" A CEDAR Working paper Royal College of Art, 1971