ARCHITECTURE AND AUTOMATIZED METHODS; CRITICISMS ON THE CURRENT ISSUES

by

Anne Marie Fourcade

Dp. Unite Pedagogique d' Architecture No 6
Paris 1972

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master in Architecture in Advanced Studies

at the

Massachusetts Institute of Technology

June, 1975

Signature of the Author ______________________
Departement of Architecture June 18,1975

Certified by ______________________
Thesis Supervisor - Nicholas Negroponte

Accepted by ______________________
Chairman, Departemental Committee on Graduate Students

Abstract

ARCHITECTURE AND AUTOMATIZED METHODS:
CRITICISMS ON THE CURRENT ISSUES

Anne Marie Fourcade

This thesis is an attempt to establish the reasons for dissatisfaction with the use of automatized methods in architecture. Computer Aided Design Methods have produced some reasonable results in that field; however the most remarkable programs reach a sort of end point and, at least in the same direction, improvements seem problematic.

One of the reasons might be the simplicity of these approaches which, in the name of clarity, tend to reduce the totality of the design process to the establisment of some well-defined and functional aspects of architecture.

Investigation of the reasons for the shortcomings of these methods is the first intent of this paper.

We will present briefly, the most interesting works in Interactive Systems and Space Allocation Techniques.

Finally, since we are interested in Artificial Intelligence Research, one of the most recent A.I. points of view will be exposed, and the question of its possible utilization in the field of our interest will be raised.

ACKNOWLEDGEMENTS

I would like to express my thanks to Nicholas Negroponte who introduced me to the Computer Graphics Community. His advise and encouragement as well as his constructive criticisms were of great help to me in writing my thesis.

TABLE OF CONTENTS

INTRODUCTION

For more than a decade there has been an increasing proliferation of computer aided design techniques. Through the context of the university, teaching and research in that field became integrated in the curriculum of most of the architecture schools (at least in the USA and UK). At the same time, a certain feedback occurred in practice.

From the automation of some parts of the designer's activity to the general inquiries regarding the feasibility of the models of the design process, the object of the computer-aided design applications or attitudes is large. The range of preoccupations covers the precise and well bounded parts of the architect's activity, as well as the more general questions on the nature of the design process. As we move further away from the technical uses of computers in architecture toward less practical interrogations, the unanimity disappears. We can recall the different design methods theories which are worthy of shedding some light on the important nature of the design process.

Between these two extremes --technical points and design process in its totality-- lies a large field, in which most of the computer applications in architecture can be found. The success, if sometimes it occurred, was success in a very narrow sense. Some programs deal reasonably with one or more functional aspects of architecture; a graphic system is able to correctly recognize free-hand drawings. But in the both cases, we are not satisfied. The requirements of an architectural project cannot be reduced to its functional aspect; the free-hand drawing recognition needs some deeper semantic understanding in order to become really helpful. The most sophisticated programs seem to have reached an end point, from which new improvements, at least in the same direction, are difficult to imagine.

The first intent of the thesis is to explore our dissatisfaction with the insufficiencies of these methods. The so-called computer-aided design methods are not very successful. What are the main reasons which could explain the shortcomings of an approach which had --and still has-- a certain amount of attractiveness? The reasons may come from the methods themselves which are too simple, and whose explicity transforms the object they manipulate or the process they pretend to mimic. But maybe the difficulty of such an approach stems simply from the nature of the object to which it is applied: the architectural space. Its qualitative description is still in infancy; no language able to describe it exists, even if some interesting steps are being made in that direction.

These are certain points, among others, that will be investigated within the thesis. A parallel investigation was conducted among people directly involved --in research or practice-- with some use of these methods. That was an occasion to test satisfaction, or skepticism, as well as optimism for the future.

Another part of the thesis will briefly analyze these techniques: interactive systems and space allocation techniques. The presentation does not pretend to be exhaustive, but rather has intended to explore the most outstanding works in that field, referring to a complete bibliography when more detailed explanation is required.

Finally, we were interested in some works developed in Artificial Intelligence and, as we believe in the possible utilization of these approaches, we will present one of them. That does not guarantee any particular future application of A.I. in computer-aided design methods in architecture; it simply shows a personal interest in some aspect of this research and a belief that A.I. points of view could sometimes be helpful for architects wanting to use machines.

PART I

Chapter 1: OVERVIEW OF INTERACTIVE SYSTEMS

The following chapter will focus on the main results achieved in architecture and planning with so called "Interactive Systems". We can find all sorts of more or less easy interactions between man and machine from fastidious interaction through predefined text commands to the recognition of free hand drawinds.

The following section will review: input and output text interaction, static graphics, dynamic graphics and input techniques.

There are many issues that will not be discussed here, for further information on these issues, refer to the bibliography.

1.1 Input/Output Text Interaction:

1.1.1. SYMAP:

Computer cartography has been widely developed to communicate and analyze information (Proceedings of the IEEE, 1974,2.1); we will simply present SYMAP, which can be considered one of the most complete systems.

SYMAP (Dudnik,1971,2.6) is a computer program which displays spatially disposed information. This information can be of different sorts: physical, economic or social raw data. SYMAP allows the graphic display of numeric or ordinal variables.

Three types of maps are produced by the program:

CONTOUR: The contour (or isoline) map consists of curves, which are closed lines made up of points having the same fumeric value or height. Between any adjacent lines a continuous variation is assumed. Such a map can represent spatially continuous information (topography, population density...).

CONFORMANT: The area limits of the conformant map are significant. The areas studied are enclosed by a boundary (close polygonal lines). An area can represent a predefined spatial unit (administration map...). The entire spatial unit is given the same value and a specific graphism, depending on the value of the considered variable, is printed as output.

PROXIMAL: The spatial units are defined by the nearest neighbor methods from point information. Each character location is assigned the value of the data point nearest to it. The program determines the boundaries which are assumed along the lines where the value changes. A conformant mapping is applied.

1.1.2. DISCOURSE:

The Discourse language (Porter,1972,2.4) can be considered as the first high-level urban design-oriented language. Interested more by the course of reasoning than by the qualities of the result, the authors were concerned in finding a context and a set of explicit procedures that could replicate the design. Discourse was at first intended to be a mode of investigation of the design process. Its developments have shifted from that first intention into a computer system that assists a designer and increases the level of complexity he can handle.

The typical uses of the language are the following:

-representation of an existing environment,

-display of a proposal,

-examination of the consequences of a set of policies.

The considered environment is divided into a grid of locations. Each cell of the grid is a possible location for an 'attribute'. An attribute is an event which can occur within the space defined by one or more cells (e.g., water, building, metro-station...). Several attributes could exist in the same cell. The value of an attribute can change from cell to cell, depending on some characteristics associated with it ('charvar' as water <<depth>>). Locational qualifiers (where, when...) and relational operators (less, equal...) make possible a comparison between charvars.

Some predefined operations allow the user to manipulate and display urban data. Any rule or new localization process can be decided by the user. Some localization techniques can be carried outside Discourse, and the consequences of the new assignement can be tested within the system. The effectiveness of the Discourse language is partly related to the multi-system environment to which it belongs.

If Discourse seems to have achieved satisfactory results in the specific domain of its us --as an extension of the designer capabilities-- its performance as a conversational system are quite insufficient. From simple 'readability' --typing a predefined set of commands and interacting in this quite rigid way-- to a more natural conversational mode, there is a great distance. We will look briefly at the main results obtained by some Artificial Intelligence programs in that domain (Winograd,1974,4.1).

1.1.3. Natural Language Systems:

The early programs in language understanding did not try to understand the meaning of sentences, but merely manipulated words from simple relationships or from keywords in a sentence.

STUDENT, written by Daniel Bobrow (Bobrow, 1968,4.2.1), converts sentences describing simple algebra problems into a set of equations and solves them. The program works by using a simple method of pattern-matching, linking phrases to variables.

SIR (Semantic Information Processing) was written by Bertram Raphael (Raphael,1968,4.2.1). The system answers questions about simple relationships between objects (as 'have' or 'part'). The program is mainly concerned with logical relations between objects in a data base.

ELIZA was written by Joseph Weizenbaum (Weinzenbaum,1966, 4.1). The program mimics a dialogue between a psychiatrist and a patient. The result is quite efficient and the system seems to behave intelligently; in fact, it deals with some predefined patterns and key-words. The connection between sentences is made by the listener, and the program does not have any deep analysis of the sentences.

These quite straightforward systems were followed by a 'second generation' of programs concerned with a deeper understanding of the language. These programs reduced the scope of their concern and worked on specific narrow domains or micro-worlds.

With LUNAR (Woods,Kaplan,Nash-Webber,1972,4.2.2), a system to answer questions about mineral samples brought from the moon, Woods developed the so-called 'transition net grammars' to describe the grammatical facts about English needed to understand complicated structures.

SHRDLU (Winograd,1972,4.2.2)seems to be the most complete system developed to date. The user is talking to a robot and ask that some simple manipulations be performed in a simple world of blocks. The program combines syntax, semantic and reasoning. Most of the knowledge is represented as procedures written in MICROPLANNER (Hewitt,1968,4.3).

MARGIE (MARGIE,1972,4.2.2) was developed by Schank's student. The system makes paraphrases of a given sentence and draws simple inferences from an input.

The recent works are more concerned with the studies of the components which must go into a larger system.

The idea of case structure developed by Fillmore (Fillmore,1968,4.2.3) and implemented by Simmons (Simmons,1973,4.2.3) concerns the small number of ways in which an object can be related to an action. The process of understanding as related to a large structure of knowledge has been explored by Shafe (Shafe,1972,4.2.4) ('conceptualizations') and Abelson (Abelson,1973,4.2.4) ('scripts'). Some possible ways of drawing connections between pieces of knowledge are proposed by Schank (Schank,1973,4.2.4) ('conceptual dependency') and Charniak ('demons')(Charniak,1972,4.2.4).

Through the last systems considered, we can see the emergence of a notion: the importance of the representation of knowledge. We will come back to that point in the last part of the thesis. The meaning of a simple fact can no longer be understood per-se, but has to be related to more complex structures.

The last system we present is the only one which was purposely designed as a designer partner, able to handle the guidance and strategy control during the designer's generation phase. IDEAS (Nevill,1974,3.3) wad designed as 'a program to augment concept generation'. It tends to reach that goal using different tactics: in providing encouragment and support as well as specific stimulation; in inciting the exploration of new regions in the designer's problem space.

A specific structuring of information allows the grouping of words and key-words. Groups are build in different categories; they can orient the conversational flow or furnish specific technical responses.

IDEAS can be considered as an experimental research tool, allowing studies of methods for teaching humans to increase their creative power, as well as studies on the nature and techniques of interactive prose and conversational modes.

1.2. Static Graphics:

Graphic Outputs can be divided into two categories: Static Graphics and Dynamic Graphics. Plotters, Computer Output Microfilm and Storage Tube allow the generation of 'static' pictures. There are two classes of Dynamic Graphics Output: either drawings are decomposed in a succession of lines represented in a list in the computer (vector generator), or a picture is made in a fixed sequence as in a television set (Raster Scan).

In the following paragrah such a precise distinction will not be observed (for exemple in the case of hidden-line elimination algorithms).

1.2.1. Digital plotting systems:

Plotters allow the display of computer outputs in a graphic or literal form. Plotters work on a basic digital incremental principle: the decoded input commands from the computer provokes incremental (sometimes analogue: Benson) steps along X or Y axes or at some angle. The characteristics of a digital plotter are given by the choice of incremental step sizes (500 steps/inch is high quality), the number of basic steps (8 or 24 in the best case) and the speed (up to 50 inches/second).

In electromechanical plotters the plot is produced by the movement of a pen on a recording paper. Three types of digital plotters exist:

- drum type: the plot is produced by a rotary motion of the drum (X axis) and by a lateral motion of the pen carriage (Y axis).
- flatbed type: the plot is produced by a lateral motion of the beam and by a vertical motion of the pen carriage.
- hybrid type: belt plotter of Calcomp, that accelerates at over 4G's and drum plotter of Gerber.

Another sort of plotter is the electrostatic plotter, which is cheap and fast. It uses a dots printer, which can plot from 1 to 200 dots per inch.

All of these plotters can be used in an on-line or off-line system.

1.2.2. Computer Output Microfilm:

Another possible way of recording data is to use high speed microfilm plotters (Stromberg-Carlson) High speed microfilm recorders have been used in such applications as curve plotting, drawings, mapping, animation motion pictures... Such electronic systems accept digital signal off-line from a magnetic tape or on-line from a digital computer. It then converts binary codes into combinations of alphanumeric printing, or curve plotting. The plot is produced by a movement of a cathode ray tube electron beam. Information is recorded at high speed on both microfilm and photo-recording paper. The information recorder can be used in storage and retrieval systems or a hard copy can be obtained.

1.2.3. Storage Tube; "a Volatile Plotter":

The last device we will present was purposely designed at MIT (Slotz,1968,2.3) as a low cost terminal for a time sharing system capable of rapid alphanumeric and graphic display, but without dynamic capabilities. It is somewhere between the slow typewriter and the expensive refreshed CRT.

The Storage Tube display unit consists of three sections :

- a controller, which decodes bits from input and contains a symbol generator and a vector generator;
- a display unit, direct-view storage CRT, the surface of which is a console memory. The surface of the display unit stores and displays the image (data can remain for hours on the screen);
- a key-board.

While cheap, the storage tube has some unpleasant characteristics. It is dim, its writing speed is slow, and local erasure is not possible. These limitations make interactions somewhat difficult and limited to the display of static graphics.

1.3. Dynamic Graphics:

Picture generation and especially dynamic graphic was permitted through the use of the cathode ray tube display (CRT). A high speed beam of electrons is generated and deflected in various parts of the phosphorescent surface of the screen. As the phosphor does not persist very long (after 1/5 s. half of the brightness has disappeared), the glow produced by the electron beam goes away. The same picture must be 'refreshed' repeatedly in order to avoid 'flickers'.

A picture can be represented point by point (single point display picture). Each word in the display file represents a command to plot a single point on the screen. The complexity of the picture which can be displayed without flickers is limited (no more than 2000 points can be displayed without flicker).

More efficient systems include vectors generators and character generators. The dot or line patterns are made 'known' by the hardware. The time to plot the pattern and the amount of storage required to define them are reduced.

In the early sixties, with the works of Sutherland, Johnson and Roberts, we see the emergence of a new method of man-machine communication. Dynamic manipulation of line drawing in two dimensions, manipulation of objects in three dimensions and hidden-surface algorithms are the main steps toward a more comprehensible and easy way of manipulating objects on the screen.

1.3.1. Sketchpad:

Sketchpad (Sutherland,1963,2.6) is the first graphic system allowing real time interrelation through the medium of line drawing.

The main capabilities of the system are the following:

1- Any symbol or 'sub-picture' can be used as often as desired in order to draw a pattern.

2- Constraints satisfaction: the user can specify mathematical conditions to be satisfied on a drawing. Basic relationships such as making points on lines, or making a line vertical or horizontal are part of the 'fixing-up' process.

3- Definition Copying: we can make the side of a geometric figure be equal in length

Sketchpad stores the relations between the various parts of a drawing and the structures of the subpictures. It is easy to change the pattern of a sub picture without changing the structure of the general picture.

The data storage structure (ring structure) allows the performance of basic operations implemented in Macro instructions (insert, delete a member of a ring). Separation of general and specific in the data structure is obtained by collecting all things of one type together in a ring under a generic heading.

The most general features of the system lie in a few general functions. These functions allow:

- the expansion of instances (subpictures within subpictures)
- the recursive deletion (removal of certain picture parts will remove other pictures parts in order to maintain the coherence of the drawing)
- the recursive merging (combination of two similar parts forces combination of things, depending on the parts).

1.3.2. Sketchpad III:

From Sketchpad to Sketchpad III (Johnson,1963,2.6) graphical techniques are expanded to the representation and manipulation of three-dimensional objects.

Johnson's system is able to manipulate 'wire-frame' figures in the dimensional space.

The outputs are graphical images of three-dimensional objects displayed on-line on a CRT. Four views of the object are displayed : top view, front view, side view and perspective. The light-pen is used to guide drawing on the screen; the element the light-pen is pointing at can be erased or moved. It is possible to magnify, reduce or rotate the drawing, force or relax the perspective.

Such graphical transformation as rotation, magnification and perspective are performend by a single 4*4 matrix.

The rotation facility is used in order to draw three dimensions on a two-dimensional surface. Lines are drawn directly on true length on a surface which has been located parallel to a viewing quadrant.

1.3.3. First Architectural Applications:

The first generation of Computer Graphics was rapidly followed by more architecturallyy-oriented uses. The most well-known system is Urban5 (Negroponte,1969,2.6). For the first time, an architect was able to sit down in front of the machine and enter into a simple dialogue. The primary intention of the authors of Urban5 was to develop a system which, after a certain amount of communication with the user, could develop a certain design intelligence. At the same time some experiments in architectural simulation (Hendren,1969,2.1), and an important experiment in architectural practice (BOP(Milne,1969,2.1)) were developed.

1.3.4. Hidden-Line Elimination Algorithm:

In order to improve the display of three-dimensional outputs the hidden-line elimination was of first importance. We will examine two of the hidden-line elimination algorithms.

The Robert's program (Robert,1965,2.5) processes photography of some geometric objects into a line drawing. The line drawing is transformed into a three-dimensional representation, and a final output with hidden line removed can be displayed from any point of view. We will present the last part of the work --the hidden line elimination-- and compare the algorithm to another approach, which is more economical for a complicated scene.

In Robert's approach, each line is tested against every opaque surface. The lines are defined by the intersection of the planes of the polyedron considered. Lines are eliminated in three stages:

- three-dimensional clipping against the screen boundary. The part lines out of the screen or behind the observor are eliminated.
- removing back-lines: rejecting lines which are partly or wholly invisible.
- testing lines against other volumes; testing each line for obstruction by other polyedron volumes.

The algorithm is satisfactory up to thirty volumes; for a complicated scene, the large quantities of computation required makes it very slow.

The Warnock's algorithm (Warnock,1969,2.5) allows a better time computation performance. Each portion of the display screen is examined through a 'window'; one determines if anything interesting appears. There are three possibilities:

- nothing appears; it is blank;

- the features observed are simple enough; a display is generated;
- the algorithm fails because the feature is too complex to analyze; the window is subdivided into several windows, each of which are successively examined, recursively, up to the possible resolvable spot on the screen (in that case a 'display by default' is processed).

Other algorithms to the hidden-line problem have been proposed by Kubert, Szabo and Gulieri (Kubert,Szabo,Gulieri,1968 2.5), Galimberti and Montanari (Galimberti,Montanari,1969,2.5); Loutrel (Loutrel,1970,2.5); Ricci (Ricci,1970,2.5) and Encarnacao (Encarnacao,1970,2.5).

The hidden line elimination being resolved, addition of color and shade was a second step in the representation of objects; the University of Utah developed techniques allowing realistic half-tone or colored representation of objects. Shade or brightness can be computed. Gouraud (Gouraud,1971,2.5) wrote an algorithm which 'erased' the edge between facettes in which objects are partitioned; in half-tone rendering, a shiny appearence is obtained. Parke (Parke,1972,2.5) developed animated sequences of human faces changing expressions.

An exhaustive review and new promising approaches of the hidden-surface problem can be found in Sutherland's report (Sutherland,1974,2.5).

1.3.5. Evans and Sutherland Graphic System:

Among the systems most recently developed, the Evans and Sutherland Picture System is one of the most sophisticated. The system presents moving pictures of two-or three dimensional objects. Changing the perspective of an object can be displayed in real time. Objects can be translated or rotated and can change in scale. You can zoom into a picture and isolate details. Individual sub-objects can be independently manipulated.

1.3.6. Raster Scan:

Up to now, what we have described on graphics systems is mainly concerned with the display of line drawings. To extend the capabilities of Computer Graphics into picture making, the vectored display (on CRT or storage tube) which processes pictures sequentially is impracticable.

The raster scan (Entwisle,1974,2.3) is a display system that refreshed itself from left to right. The storage of a drawing and its display are synonymous. A multiple point per bit display (up to 8 bits per point image) allows a wide range of possible image (color, gray tone).

1.4 Input Techniques:

1.4.1. Light-Pen and Tablet:

Input devices allow the user to interact with the computer. We have already examined the keyboard and its insufficient capabilities to enter graphical data.

Using a CRT, a user may want to erase lines or character, or to add lines or symbols on the screen. The 'light-pen' transforms the CRT in an input device. A photo-cell placed in a shaped housing responds only when light, coming from a CRT drawing, falls in its field of view. The light-pen can be used in two ways:

- as a pointing device, pointing at some items already on the screen, for which it is well suited;
- as a positioning device --in that case, a tracking program must be running in the computer--, for which it is ill-suited.

Another device, a 'tablet', allows the user to draw on a flat surface with a stylus. Each line of a tableet carries a digital code signal that can be picked up by the stylus (Rand tablet)(Keast,1967,2.3) and will be converted to binary integer form.

1.4.2. Evans and Sutherland's Two Pens Tablet:

An interesting system --Evans and Sutherland's two pens tablet (Sutherland,1974,2.3)-- enters, through the use of several two-dimensinal views, a three-dimentional description of an object. On a large tablet area several views of a three-dimensional object are digitized. With a multiple pen the user indicates a single point by pointing simultaneously on the two views, and the three-dimensional positions of the point are defined.

1.4.3. HUNCH:

Through the use of tablet and stylus, Computer Graphics started to cope with a new and quite exciting problem: sketch recognition. The main research in that field had been, and is presently, conducted by the Architecture Machine Group. The problem of sketch recognition is viewed as 'the step by step resolution of the mismatch between the user's intentions and his graphical articulation' (Negroponte,1973,2.1).

HUNCH (Negroponte,1972,2.6), system of sketch recognition, allows
the user to keep his free hand way of drawing, as well as his inaccuracy. The system constructs two representations of the sketch as it is drawn:

- a one dimensional data structure (coordinate information and measurement of pressure upon the stylus). This feature facilitates the problem of data compression, when the geometry of elements is reduced to a list of nodes and linkages. The ambiguities on the curves, straight lines or corners are successfully handled by the program.
- a two-dimensional data-structure (two-dimensional bit-map).

Among other capabilities, the system is able to separate diagrammatic elements from projective elements; it guesses about intended connections and possible intersections.

An extension to HUNCH capabilities was brought about by C.Herot's program (Herot,1974,3.3). Some semantic knowledge about the subject matter being sketched is stored into the machine during the recognition process of the sketch. The idea is simply to direct the machine search with some decisive 'clues' in order to recognize entities in a sketch.

The present and coming works of the Architecture Machine Group, is in a certain sense, directly opposed to the current uses of Computer Graphics as restricted to the displays of generated results. They are mainly concerned with structures or programs 'that can deal with the properties of incompleteness, contradiction and vagueness which are characteristics of any design behavior...' (NSF proposal,1974,2.1). The recognition of design intentions of the designer's behavior, up to now ignored, should lead to a system able to carry the ambiguities of human behavior and able to replace the explicitness of the early Graphic Systems with some sort of responsiveness to human subtlties.

1.4.4. Machine Vision:

The interest of Artificial Intelligence (Winston,1973,4.3) in Machine Vision --as well as in Natural language Understanding-- is related to the question of knowledge interaction on many levels and to the problem of large systems organization.

With the 'body finding' problem (partition the observed regions of a scene into distinct bodies), Guzman (Guzman,1968,4.3) starts with a relatively simple syntactic theory and ends with some semantic implications. The system works in two passes. The first pass gathers local evidence. It checks vertices and decides which of the surrounding regions belong to the same body. The second pass combines the first evidence into a parsing hypothesis. The program uses a set of theories which decides how to use the evidence to best advantage. Placements of links can be 'inhibited', subject to contrary evidence from adjacent vertices. A good understanding of the body linking problem depends on semantic justifications for the generation of links.

Waltz (Waltz,1972,4.3) brought some important semantic implications to the body problem. Waltz's theory is directly related to Huffman's (Huffman,1970,4.3) and Clowes' (Clowes,1971,4.3) works. The labeling of lines in a drawing is dependent on their particular cause in the scene observed: Is a line a shadow, a crack between two adjacent objects, or a seam between an object and the background? The set of lines labels is expanded in order to include some knowledge about the physical possibilities. If the combinatorial number of possible labeling around a vertex is increased, the number of arrangements physically possible is reasonable. A filtering procedure combining physical possibilities and lighting of objects speeds the recognition process. With Waltz's program, the importance of a better ability to describe is underlined.

Learning how to do good description was the main goal of Winston's research (Winston,1970,4.3).

How can concepts be learned from a few selected examples? The machine builds a model which is 'a proper description augmented by information, about which elements of the description are essential, and by information about what, if anything, must not be present in examples of the concepts'(Winston,1970,4.3). 'Near-misses' are examples introduced in the model emphatic relations (must and must-not-be type). These relations contain information about what is essential and about what characteristics should not be present in a sample matched against the model concept.

Chapter 2: OVERVIEW OF SPACE ALLOCATION TECHNIQUES

Space Allocation techniques involve the localization of elements within a space (two dimensional or three dimensional space). Space allocation problems have been solved for a long time , more or less intuitively by architects; the methods they utilize are not necessarily explicit. In order to make some part of the design process automatized, we have to make explicit the methods utilized. The necessity of clarity has strongly reduced the scope of the problems handled by these techniques. They tend to treat functional aspects of architecture, in the most primitive case, as circulation function. In the most advanced programs, different functional constraints are precisely defined. The problem of the formalization of any qualitative aspect of the architectural space still remains unexplored.

We want to underline the necessity to consider the solutions obtained by these programs not as 'solution' or definitive plan, but as a step in the design process, as 'schemas' which have to be further manipulated and improved by the architect. This is the only reasonable way of considering and using these techniques.

We will examine successively the solution criteria, the solution resolution and the solution evaluation. Finally, the general problem of data representation in a computer and the particular data structures utilized in space allocation problems will be considered.

2.1 Solution Criteria:

The criteria can be defined as the means by which to evaluate whether or not the goals that have been set are reached. The goals can be of several sorts: to improve an initial solution (CRAFT (Armour,Buffa,1963,3.3), IMAGE (Weinzapfel,Johnson, Perkins,1971,3.3)), to build up a solution wich has to maximize or minimize a function (Simmons'program (Simmons,1969,3.3), CRAFT (Armour,Buffa,1963,3.3), or simply to develop a solution without any optimization but with considerations for certain constraints: GSP (Eastman,1971a,3.3), Fosplan (Yessios,1972b,3.3), Grason's program (Grason,1972,3.3)).

The criteria are directly dependent on the constraints which have been stated. They can serve to stop or to orient the resolution process.

The constraints express the relationships between the elements to be located on a support (most of the time a 'grid'), or which are already localized, in the case of programs which improve an initial solution. The constraints can be simple metric relationships expressing distances between elements. These relationships are generally weighted by some measure of the interrelations between elements. In the case of Whitehead's program (Whitehead,1964,3.3), the measure of a weighted interrelationship betweem elements decides the order in which they have to be located, and the measure of the circulation costs decides which to choose among several localizations. With CRAFT (Buffa,Armour,1963,3.3), an objective function, again measuring the circulation costs, decides which solution to choose.

To overcome the simplicity of these first approaches, more recently developed programs present the possibility of defining more complex criteria. With IMAGE (Weinzapfel,Johnson, Perkins,1971,3.3) and GSP (Eastman,1971a,3.3), geometric relationships are extended (adjacency, alignement, visual acces...). Eastman defines (Eastman,1971a,3.3) a 'constraint graph' in which nodes represent elements and links represent relationships between elements; the valuation on the links expresses the degree of restriction of a relationshp; the measure of the degree of restiction of an element (sum of the links value) decides in which order to locate the elements. In IMAGE different criteria are aggregated into a single function to be minimized.

2.2 Solution Generation:

Space allocation programs localize elements in a space. The space which will receive the elements possesses a finished number of possible locations. Without taking into consideration any constraint, the possible number of affectations of n elements within a space admitting p possible locations, is A_p^n ;

in the case of programs in which an initial solution defines the boundaries of the plan (CRAFT (Armour,Buffa,1963,3.3)) the possible number of localizations becomes the number of permutations A (where n=p). In both cases, the number of the possible solutions is enormous. With the establisment of the constraints, the number of possible solutions is reduced. As any solution is no longer satisfactory, some way of guiding the production of solutions is fecessary. With the introduction of 'search strategies', the problem space is reduced: The problem space defines the total possible paths which produce solutions.

One of the simplest strategies is called 'generate and test'. All members of the set of the possible solutions can be produced, or the process can be stopped if a satisfactory enough solution has been found. In the case of exhaustive enumeration, we have the guarantee of finding the best solution among the set of the possible ones. In that case, we have an optimization of the solution generation. This sort of optimization can only work in very limited cases. The use of simple enumeration is totally impracticable as the number of elements to be located increases (a very long time of computation). However some programs use the generate and test method as part of other strategies (CRAFT (Buffa,Armour,1963,3.3), Lokat (Bernholtz,1969,3.3)), or in the case of very limited problems (Steadman's program (Steadman,1970,3.3)). Random or constrained random sampling has also been used in site-plan solution generation by generate and test (Seehof and Evans's prog.(Seehof,Evans,1967,3.3), Teicholz's prog.(Teicholz,1969,3.3)).

To avoid the exhaustive exploration of the problem space, it is necessary to find good search srategies. The strategies are more or less efficient, generate and test being one of the most worthwhile. Heuristic procedures can be defined as a way of finding satisfactory solutions with some rapidity and efficacy. The guarantee of finding the best olution (optimum) is lost, but it is assumed we will find a reasonable result (pseudo-optimum).

Heuristic searchs are usually represented through a tree. The different states of the solution generation are the nodes of the tree, the branches represent the passage from one state to another. The possible paths through the tree are dependent on the search processes or decision rules which have been decided. The decision rules can be pre-established, or they can be dependent on the precedent state of resolution (CRAFT (Buffa, Armour,1963,3.3)). In the case of programs generating solutions, as an example of possible decision rules, we have to decide:

- the sequence in which elements are added to the arrangement,
- the sequence in which constraints have to be applied,
- the order in which to consider locations,
- which sort of back up rules have to be applied when there is no possible location for an element.

With the preceding rules, it will be very important to start with the most constrained element (Whitehead's prog.

(Whitehead,Eldars,1964,3.3), GSP (Eastman,1971a,3.3)).

An extension of the generate and test method is the hill-climbing strategy (which has not too much heuristic power). CRAFT (Buffa,Armour,1964,3.3) is a typical example of the use of the hill-climbing method. The program tries to minimize a function representing circulation costs between spaces. Given an initial solution, possible exchanges between elements are considered; the transposition bringing the best arrangement is retained, and the process is repeated until no possible improvement can be obtained. If the general search is a hill-climbing strategy, generate and test is applied in the generation of comparative solutions on each intermediate level. Hill-climbing does not produce optimal solution, but generates results which successively verge on an optimum. The insuffisency of the method, which ignores a better solution and stops on a local optimum, is slightly improved when several trials are performed in choosing different starting points.

The programs developing the most sophisticated strategies are Eastman's program, GSP (Eastman,1971a,3.3) and Pfeffercorn's Program DPS (Pfeffercorn,1971,3.3). Other programs (Lee and Moore's (Lee,More,1967,3.3), Mitchell and Dillon's (Mitchell,Dillon,1972,3.3), Lidgett and Frew's (Lidgett,Frew,1972,3.3) have developed some more or less simple heuristic rules to generate layouts.

Even in the most sophisticated case, the strategies developed within the programs are very different from a human resolution process. The first important difference lies in the definition of the constraints, which is static and definitive all along the search process. The redefinition of the constraints as well as the the introduction of new data along the resolution process are typical of human problem solving. The conception of self-modifiable programs, as well as the conception of learning programs, is still extremely problematic.

2.3 Solution Evaluation:

Little has to be said about the testing of solutions in space allocation programs. As we have noticed, tests against the measure of the performance of the solutions can serve to direct the choice o intermediate solutions (CRAFT (Buffa,Armour,1964, 3.3)). Bernholtz's program (Bernholtz,1969,3.3) generates a solution from one chosen criteria and tests the solution against the other criteria. With some programs, the solution space (number of generated solutions) is one (Whitehead's Prog. (Whitehead,Eldars,1964,3.3), Eastman's prog.(Eastman,1971a,3.3)). In that case, the solution is supposed to be satisfactory enough and, if not the best, at deast it respects the most important constraints.

Among the preceding programs simulation is never used to test the quality of solution against other possible criteria not directly taken into account.

However such programs exist, they can handle operation costs, structural stability, environmental performance of a building,(Markus,1972,3.2),(Paterson,1972,3.2),(Harper,1968,3.3),(Allwood,1972,3.3),(Hawkes,Stribs,1971a,3.2).

The last way of testing solutions is to allow human intervention during the generation process: In other words, to develop an interactive man-machine system. The introduction of human intuition to accept, reject or ponder a solution, or to direct the generation process, in a sense denies the possibility of formalizing ill-defined and qualitative requirements in a computer language and associates human judgment and creativity with an automatized process (Negroponte,1969a,3.1) (brodey,1968,3.1).

2.4 Representation of Data:

Not solely related to space allocation programs, the following treats the basic requirements of data representation in a computer. Complements on the subject can be found in articles (Gray,1970,7)(Page,Wilson,1973,7) or in Knuth (Knuth, 1970,7).

When you have a collection of data elements (numerical values, names...) stored in a computer, some way of organizing them into the memory is needed. Such organization of data elements has two purposes: to preserve the relations between the elements considered and to provide an easy way of manipulating the data (retrieve, insert, delete a data element).

First, we will examine the different basic types of data structure. Then, we will present the type of data representation used in space allocation programs.

2.4.1. Basic Types of data structure:

Some definitions: a record is a collection of elementary data items (ex: information about a student: name, age, field..). Records are collected into logical unit files. The arrangement and interrelation of records in a file form a data structure.

There are many types of data organization: sequential organization, random organization and list structures.

Sequential Organization:

Records are stored in a position relative to other records and according to a specified sequence. A key decides the order in which they are positioned (in a telephone directory, the key is the surname data element of the record). This data organization is the simplest because the mechanism for accessing the data is already built into the computer hardware (the logical order of records in the file and the physical order of records in the machine are the same). As a consequence, the retrieval operations will be fast; updating values, inserting or deleting an item will be difficult. In that case, it is necessary to push existing records (copy the entire file) to make room for a new item.

Random Organization:

An arbitrary address is associated with a record, which is then stored at that address. The address can be specified by the programmer, which is not very practical (direct address). It is better to create a table of records' keys with their associated address (dictionary method). However, the dictionary occupies as much space as the data and with a very large dictionary, searching for an item can take a long time. Another method (hashing) converts the key of the record into a single address (one can replace each letter of the key by a number for exemple). If several key record calculations give the same address, one can collect in a list (see below) all records that hash to the same address. In order to find the element, one searchs the list after performing the hashing operation. With that method, any record is retrieved by single access. Insertion and deletion does not affect other records.

List Organization:

A list is an organization in which records are chained by pointers. A pointer links one piece of data with another; it is a word which contains an address in the memory. In a list, the logical organization and the physical support are not the same: a list can be logically scattered through the memory.

Many lists can pass through a single record (multilist structure). In that case, an element can belong to several lists. Insertion or deletion of a record in a list are easily performed (e.g., to delete: the pointers to the element to be removed are made to point to the next element).

Ring structures: In a list, the last record points to the first record. Multiple ring lists can pass through a record.

More complex structures:

A Tree structure is a structure which has no closed circuits. It is a level structure in which a block at any level may point to a block further down the tree.

A Hierarchical stucture is a structure with levels of hierarchy but constructed with rings. From an initial ring, any record can be linked to logically related elements which are arranged in a ring structure. This type of structure allows starting with any record in file and moving up and down in the hierarchy.

We will conclude with programming language aspects. FORTRAN, COBOL and PL/1 handle sequentially organized files very well. LISP, which was developed to deal with heuristic programs, is a list-oriented language. In PL/1, a pointer capability allows dynamic allocation of space and storage of pointers between records which are organized in a list structure.

2.4.2. Space Representation in Space Allocation Programs:

In order to represent and manipulate elements in the architectural space, some particular data structure is needed (Eastman,1970b,3.2)(Maroy,1973,3.2).

With the euclidian representation, cartesian orthogonal coordinates are used to locate elements, IMAGE (Weinzapfel, Johnson,Perkins,1971,3.3) utilizes this representation.

The first programs have widely utilized two-dimensional arrays or 'grids'. Depending on the size of the unit domain, the representation is more or less accurate. Irregular forms can be only approximated. Distance between the elements are calculated rectangulary or diagonally. With the accuracy of the representation (size of unit domain), the memory requirements and computer time rapidly increase.

With the 'hierarchical array', less memory requirement is needed. Domains which are not homogeneous are recursively subdivided into 4*4 grid cells. With that representation, large domains are defined more economically. Hierarchical arrays were used at Stanford Research Institute as a robot's internal representation of the world.

The 'variable size domains' was utilized by Moran (Moran,1968,7). With that representation, domains are defined by the prolongation of the sides of the elements and of the sides of the envelope. Domains could be organized into a lattice which is represented by a list structure or can be represented by a variable size arrays.

Another representation which has been widely utilized in space allocation programs is the planar graph representation. A floor plan can be represented by its dual graph. The dual can be stored in a matrice or represented in a list (list of the successors). The graph expresses the adjacency relation of the plan. Each vertex can be associated with the length of the wall it crosses. This representation was used by Krejciric (Krejciric,1968,3.3), and Grason (Grason,1971,3.2). A different representation by graph was proposed by Teague (Teague,1970,3.2).

The most general representation is perhaps the one used by Pfeffercorn in DPS (Pfeffercorn,1971,3.3) and by Yessios in SIPLAN (Yessios,1973,3.3). Elements are represented by combinations of convex polygons.

Each of these representations makes certain operations easy to perform and has some influence on the resolution process.

With the array representation, the minimization of distances and the constraints of adjacency are easy to handle; other constraints, such as visual access, could be laboriously performed, and angles can be only approximated. As we have noticed, with a better accuracy of the space definition, memory and processing times increase rapidly. Hierarchical array improves the array representation by using less redundant information to define domains.

With the variable arrays, the same reduction is obtained and its list structure allows a direct treatment of the adjacency constraint.

With the array representation, the localization of an element is performed step-by-step and in an aggregative way. Euclidian representation and hierarchical array are more effective in recognizing space than in locating elements step by step. IMAGE was mainly utilized to improve an initial solution. Constraints using angle verification are efficiently performed.

The graph representation allows the use of certain procedures on the properties of planar graphs. Adjacency and length requirements are priviliged.

PART II

Chapter 1: STATE OF THOUGHT IN COMPUTER-AIDED DESIGN METHODS IN ARCHITECTURE: SOME INTERVIEWS

The general inquiry into the uses and mis-uses of the computer inshde the design process brought about the idea to interview some people who were directly or indirectly involved with the so called computer aided design methods. The Boston area furnishes some of the most successful examples of the uses of computer design methods either in research, teaching or practice. To carry out an investigation among happy or unhappy users of these methods seemed necessary, interesting and exciting.

People interviewed belong to the community of enthusiasts, skeptics or detractors of these methods. All of them had been, to some extent, touched by the increasing development of these techniques. Some of them consentrated their professional activity on it, others developed some experimental work in that field; still others spent energy in strong criticism and raised important questions as to the correct use and possible future existence, if any, of these methods.

The Boston area is a dot on the map of the users of computers in architecture. However, the range and the quality of the works conducted in the last ten years allow the presentation the following opinions as being representative enough of the state of thought in computer aided design and architecture.

Eric Teicholz, Guy Weinzapfel, Nicholas Negroponte, Aaron Fleisher, Cliff Stewart, Stanford Anderson, Alexander Tzonis, Mike Gerzso and Timothy Johnson have been interviewed.

The following questions were submited:

1- When you personally became interested in the automation of some part of the design process, what was the main reason?

2- How do you decide if a design proposal is a 'solution' to a problem; in other words, what are the criteria by which you evaluate the satisfaction of a solution?

3- What concepts seem particulary difficult to formalize in a computer language?

4- Could you discuss the results you have obtained in using computer techniques? Are these techniques satisfactory or not, and from what point of view?

5- What is the main benefit of using computer-aided design methods?

6- What further improvement do you see in computer-aided design in architecture?

7- What computer technique evolutions could influence the future of computer-aided design methods in architecture?

-Reasons for becoming interested in the automation of some part of the design process:

To be a frustrated student or pracitioner, to be driven by simple curiosity are some of the reasons for being interested in the automation of the design process.

In everyday practice, the complexity of the projects to deal with, naturally leads to some way of organizing and dealing with the data; easier handling of the technical aspect could leave more time to work on the human aspect.

Automation also can be seen as a way of making the design process more rational. It can be seen as a medium through which one can understand architecture as such or analyze the design process. In the latter case, regarding the automation of some part of the design process, one can compare what the algorithm produces and what the designer does, and on the resemblance of the two, one can decide whether the algorithm was successful or not. Automation can also be a tool which allows specification and simulation of the way a system behaves over a period of time, exploring the changes of a building over time.

In opposition to the somewhat optimistic preceding opinions, was some strong criticism against these methods, which were judged to be 'primitive exercises of thinking relative to the nature of architecture' (refers to the early sixties works). The sense of 'working inductively from faulty information and trying to make that faulty information built in a synthesis of form is contrary to the way the mind works, the science works, architecture works...' (S. Anderson).

-Criteria by which the satisfaction of a solution is evaluated:

A solution can be seen as simply meeting three sorts of requirements: function, construction and 'delight'(G. Weizapfel), aspects which prevent many buildings from becoming a solution: and which could be a clear expression of what the building is about.

When the design criteria have to be explicit in order to be translated in a computer program, testing a solution against the input parameters is easy, but there is no reason to believe in the relevance of these design parameters for a particular design solution. In the final analysis, you are the only one who can define what is the best solution. That type of design knowledge is a very individual process.

If one is not concerned by solving the problem per-se, but rather in the replication of the design process, there is no necessary success in solving the problem, but there is success in replicating what the designer does.

Theoretically, success could be decided by the time the criteria, which have been decided ahead of time, are fulfilled. Practically speaking, this is impossible and there always exists a kind of feedback or an increasing approximation to the problem process; better solutions and better programs are, in fact, developed at the same time; that does not mean that a certain solution is satisfactory because the solution satisfies the criteria which have been set as criteria for fullfiling the satisfaction of a solution. The decision on the criteria to be considered is crucial.

One of the problems is how to define a measure of performance, how to define a system, how to implicitly define a solution space. If you are honest, you also have to make clear which values are sound in the system. These are not easy problems to solve because in architecture you don't know what the solution space is, and the measure of performance has only been successful in terms of dollars. We can talk about 'wicked problems' (Rittel) because, in a sense, you solve the problem by stating a solution at first.

To the precedent views, which more or less, believe in the possible statement of some criteria which can be later transformed through feedback and readjustment, the opinion which follows brings the user into the design process. In that case, the solution of any design problem is evaluated through the eye of the user of the physical environment and not from the environment itself.

To conclude, there is no reason to think that the problem is well stated ; the solution obtained is always more complex than the problem statement . New information and feedback don't stop with the design or with the building, but go with the people using the building and continue with transformations of the building; 'I don't look for problem solutions, I look for interesting evolutions of the problem'(S. Anderson).

-Concepts difficult to formalize in computer language:

Things which cannot be quantified are extremely difficult to translate into a language. Computers are only good for carrying out a rational process. Only the structural aspect has been well addressed; the functional aspect is still not satisfactorily addressed. No system exists that is able to deal with any given set of criteria and that is able to generate successful plans.

All the concepts having metaphorical equivalents are difficult to formalize, any concept that has to do with the meaning that we attach to things.

Machines are unable to computerize a process; something that involves more than one context. The ability to change context, to cross reference, to design in a different vocabulary is difficult to implement in a computer language.

Difficulties can come out of our limitation to model diffused ideas in a formalized manner.

Problems could arise in describing what the conventions are that produce arrangements in space which the designer considers to be good. In that case, it is difficult to think a set of conventions that are primitive enough to generate a large variety of possible spatial arrangements. It is not so much a language problem.

-Results obtained in the use of computer techniques:

Most of the good results in using computer techniques come from quite straighforward applications; well defined problems with a set of quantitative criteria. The use of a system such as Image is more interesting in the evaluation of solutions than in the generation. Image brought with it the problem of translating design ideas to the machine for evaluation.

Pracitioners consider computers with optimism. Computers do a better quality work, allow more consideration for alternative solutions, and they can simulate the final product.

In the case of Discourse the success was not in exploring the designer's process and in writing an algorithm that might replicate him (which was the original idea), but it was more in building a computer aided design system, a sort of amplifier of the designer's ability.

Computers can be used as a means of testing design hypothesis, and they can serve as a model to modelize the process of decision-making within the design.

From a pessimistic point of view, we can find several reasons: we are still very far from the point of using a computer as a medium. We have not been able to incorporate it in the design process, even in its limited sense. To be able to describe things architecturally, we need some computer language for architecture.

Results have been judged unsatisfactory because the breadth of problems that these techniques can handle is so narrow, because the interface between man-machine is so uncomfortable, and because the cost and energy spent are still so high.

-Main benefit of using computer-aided design methods:

These methods are viewed simply as a 'useful entry point for some kind of designer' (T. Johnson). However, in real practice they seem to have some non discussed anterest: because they are highly accurate, they do a lot of work in less time and they allow to consider several alternatives. You can simulate the final product and also keep a record of the design process, a trail which allows you to work backward, if necessary.

Beyond these practical applications, the use of these methods forces you to think more about the design process, and by placing the machine between the designer and its artifact, it brings some beneficial isolation.

Starting with an hypothesis, one can write a program which explores the consequences of this hypothesis. In that case, the creative step is not removed and the computer is used to explore many more possible deductions from your initial hypothesis. It is not an automatic process.

One possible further application of the computer is to introduce the user inside the design . The first problem is to combine and aggregate the different opinions in a very accurate way. The computer could be used to create a quick feedback of an aggregate proposal back to the group and to incite a re-evaluation of the first decisions. As a result, the interesting aspect of the application of computers is going to come when the process of design changes.

-Further improvement in computer-aided design in architecture:

The first improvement which has been noticed is technical but has strong consequences and feedback on the user.

The construction of better data bases as well as the sharing of data is an important issue. Encouragement to present suggestions (and not solutions) in more depth gives some material or graphical cues to the architect which incite him to ask questions on the alternatives and prevent him from assigning further generations to the machine. The necessity of a powerful and easy graphic language brings the question of the representation of architectural things in data structure and the more general question of the design of an architectural language.

Another important point is the question of an easy and comfortable communication between man and machine, 'all the knowledge of competence embedded in the machine'(N. Negroponte). and in a longer range, the construction of systems which could guess the intention of the user.

Some strong disagreement was raised against the idea of computer 'aided design' which sets the problem in the wrong direction. What is the design process and how can the computer enter into dialogue with it? As long as we know what design is, then mechanization can take its part, not in aiding it but in functioning within it.

-Computer technique evolutions and their influence on the future of computer-aided design methods:

Among the technology evolutions which could influence the computer design methods in architecture, the ability to design very naturaly on large surfaces was mentioned several times.

There is a need for good symbolic graphic, and the generation of cues coming up with graphic in order to be able to give a good epresentation of space, to be able to manipulate, store aid generate the image easily.

Another research area (the need of good multivariate analysis) is to help the designers to see what their problem looks like, what their problem space is and to facilitate their course through the problem space and their arrival at solutions. Look at a series of variables and come up with some sort of abstraction that would help the designer to understand the extent and the variations of the problem.

Against the 'little gadgeteries' (as 3-D displays, dynamic graphics the size of walls...) which do not bring any conceptual change, the specific conceptual change that will affect computer aided design is the problem of learning. Machine should learn how to learn and should not be told to be a good architect.

To conclude the constraint on the production of satisfactory computer aided design systems in architecture is not in the hardware or in the software but in the understanding problem. ' I cannot think of any device that would automatically change the world and absolutely guarantee a state of grace' (A. Fleisher).

Chapter 2: CRITICISMS ON THE CURRENT ISSUES:

The first intent of this chapter, which is in some sense the intent of the thesis, is to investigate the reasons for our dissatisfaction with the use of the so-called computer aided design methods. In Part I we have briefly outlined the main results achieved by interactive systems and by space allocation techniques. Interactive systems and especially graphic systems, have been mostly concerned with the display and manipulation of information already organized. On the other hand, space allocation techniques faced the problem of the generation of 'solution' or pseudo-solution, starting from some well-defined design constraints and using some specific design strategies. With this latter approach, the choice of the design stategies, the constraints and the criteria on which to decide the fulfillment of these constraints has an important effect on the final result. As we have already mentioned, these techniques can only operate with an important reduction of the usual human design process.

We will examine each of these points successively, and we will try to state when the oversimplification of these techniques is not admissible and in what sense they completely distort and misunderstand the design process. Finally, interactive systems and especially computer graphics will be discussed.

2.1 The Way Space Allocation Programs Work:

In the case of a 'build-up' approach (adding physical elements one-by-one in order to produce an arrangement) or in the case of amelioration of an existing solution, the racteristics of these programs are nearly the same. We have a certain number elements; these elements are supposed to represent the architectural space, or a piece or sub-piece of that space, in which different sorts of activities have to be localized. We have a certain number of constraints to be satisfied. These constraints were chosen at first, obviously because they were simple and easy to modelize in a formal language; it was also presupposed that they were strong and relevant to the type of architectural problem to which they were attached. We recall that the early space allocation programs dealt mainly with the localization of 'activities' in buildings with important displacement of people and materials (hospitals, factories, warehouses..). Directly linked to these constraints, some criteria decide whether or not the constraints are satisfied and in the best case, the degree of satisfaction of the constraints can be established. In order to generate an 'arrangement' satisfying the stated constraints, one or several strategies have to be implemented. Depending the more or less heuristic power of the chosen strategies, the search will be pursued with some rapidity and with some guarantee of success. As we have already noticed, success in the case of heuristic search is never the guarantee of finding an optimum solution but rather of finding some reasonable solution. The interest of a strategy can also be judged by the number of constraints it is able to handle and by the type of back-up rules it can utilize, if at some point a blocked situation occurs. The preceding problem statement can be simply exposed in the following formulation borrowed to Eastman (Eastman,1971b,3.2):

Given:

a : a space;

$(b_1, b_2, \ldots b_n)$: a set of constraints delimiting the acceptable solutions;

$(c_1, c_2, \ldots c_n)$: a set of operators for manipulating the location of elements within the space;

(e) : the initial design state (a state is an arrangement of elements);

Find a set of operators that will generate a state (e) such that the constraints are satisfied.

This heuristic search formulation is represented as a search process through a tree whose nodes are states (partial design) and whose branches are operators which transform one state into another. A solution is a path starting from an initial state and leading to a goal state.

One interesting situation which allows us to judge the heuristic power of a program, is its ability to back-up when a blocked situation occurs. The use of feedback from the problem being solved to resolve the obstructed situation is rare in space allocation programs. Most of the programs simply back-up the design tree in a nearly blind way. Pfeffercorn developed an interesting 'diagnostic search strategy' in his program (Pfeffercorn,1971,3.3). When a difficulty is encountered which makes the program unable to locate a new element, an investigation is pursued to check which constraints have been broken and which objects are involved in the violation. Unfortunately, this checking is done only on the last set of alternatives generated by the program; earlier alternatives, already generated in the partial solution, could have been one of the possible causes of the present difficulty. The diagnosis information having been obtained, a set of remedial actions can be tried by the program (build a macro-object, enter the object earlier, or resort to exhaustive search).

This digression is only apparent and emphasizes one of the evident missing capabilities of these programs. In Pfeffercorn's program, some new information, even if very little, is obtained through the present state of the program; that information is utilized in the further generation of the arrangement. The program is provided with the possiblity of redefining objectives and strategies using the present information on the problem.. In fact, Pfeffercorn's program uses a very simple feedback strategy and does not achieve a redefinition of a goal, but simply tries to check the reasons for failure through the very last step of the generation. This leads us to examine the problem solving approach and its consequent implications.

2.2 The Problem Solving Approach:

A book recently edited by Eastman has the following title: 'Spatial Synthesis in Computer-Aided Building Design'. This title is enlightening; most of the programs are involved in synthesizing a certain number of elements using some pre-established rules. The simple juxtaposition of these 'pieces of space' in which 'activities' will occur are presupposed to produce a reasonable plan. On the other hand, the final configuration of the generated arrangement is completely dependent on a certain number of programming rules. We can enumerate some of them: the order in which the elements must be located, the way the grid will be explored (if a representation by grid had been chosen); in the case of optimization of circulation, the way the distance between two elements is calculated (Is it from center to center between pairs of elements? Is it a rectangular distance; a euclidian distance?). These are some of the technical points which will determine the final shape of the arrangement. Architects, of course, do not work in that way; they tend to work in the opposite way, the generation of a form often coming a priori (from past experience, from personal preferences...), its functional readjustment coming later as a secondary requirement. We want to emphasize the non-neutrality of the programming techniques utilized: they have a non-negligible influence on the final solution. As the number of constraints increases and the strategies utilized become more complex, the respective influence of the different routines utilized in a program are no longer perceptible, as they were in the earliest space allocation programs (where a simple change in the distance computation produces a totally different output).

The problem solving approach was strongly attacked (Anderson,1966,3.1). The establishment of a predetermined goal, of predetermined constraints, as well as well-defined strategies to reach that goal, is completly opposed to the way a human solver works. The architect's way of working is not definitively determined in advance. The problem definition is neither definitive nor complete at the first steps of the design process. A constant redefinition of goal and constraints accompagnies the design process. New constraints are discovered and old ones are neglected because they are replaced by new ones judged to be more important. In a sense, a definitive goal or an ideal state of satisfaction is never reached. The importance of feedbacks from the problem being solved is essential. As we have noticed, nearly no program incorporates feedback that could orient strategies, help to define new constraints or decide to forget old ones. Some programs working on a 'build-up' approach never can be blocked and backing rules are not implemented. Programs working gn successive ameliorations of an initial solution (CRAFT, IMAGE) use an evaluation function to decide if further generations have to be attempted (CRAFT) or if some satisfying threshold has been reached (IMAGE) which allows some satisfaction of the constraints. With Pfeffercorn's program we have a good example of design strategy determined slightly by the present state of the partial design. However, the feedback process is still too simple and incomplete.

The composition of arrangements produced by these programs proceeds from elementary and punctual relations (one-to-one element interrelations). The synthesis --rather 'aggregation'-- of the elementary items supposedly produces a satisfying solution. Designers process in a very different way. They can sometimes use this method, but at the same time they utilize, an 'overall to particular' approach. The step-by-step approach of these programs ignores the nature of the design problem. We will conclude by borrowing Rittel's terminology: Design problems are 'wicked'. We will not go through the the definition of their properties as listed by Rittel (Rittel,1972,3.1); we can notice some of their most important characteristics, completely ignored by the preceding approaches. 'Wicked problems' have no definitive formulation; with any formulation, additional questions can be asked and more information can be requested. Every formulation of the wicked problem corresponds to the formulation of the solution. The information needed to understand the problem is determined by one's idea or plan of a solution. Wicked problems have no stopping rule. A solution can be improved or worked on more; no wicked problem and no solution to it has a definitive test. Anytime a test is 'successfully' passed, it is still possible that the solution will fail in some other respect.

2.3 The Way Designers Work:

We don't pretend to establish a clear comparison between the human problem-solver and the artificial problem-solver. An abundance of literature has been written on the subject, with the earliest works those of Nilsson (Nilsson,1971,4.1) and Slage (Slage,1971,4,1) and the studies Newell and Simon on human problem solving (Newell,Simon,1972,4.1). These studies were directly related to some remarkable applications in checker and chess programs, and in theorem proving programs...

Amazingly few studies have been pursued on the resolution of architectural design tasks. The earliest study was conducted by Eastman (Eastman,1970,13) on the very simple space planning task of redesigning a bathroom so as to make it 'more luxurious and spacious'. Following the study, three types of methods seem to have been used by the designers as search strategies: generate and test (or try every possible alternative until one is found that is satisfactory); means-end analysis (utilize information that relates the testing of criteria ('ends') with operators for achieving these goals ('means'); and planning (which involves an analysis of the problem structure in order to find those elements that are more closely related to other elements).

If Eastman's studies and protocol analysis are interesting, the limited scope of the problem (a very simple design task) and the apparent lack of skill of the solvers do not allow us to draw interesting conclusions.

Another work was conducted at MIT (A. Foz's thesis 1972). Foz (Foz,1972,13) was interested in studying what the designer do in the parti stage of the design process. Starting with some hypotheses about what skilled designers do that less experienced ones do not, experiments were conductd in order to test these hypotheses. The sketch problem was to design additional facilities for the MIT School of Architecture and Planning (such as spaces for large classes and important occasional lectures). Designers with different levels of skill were selected. A two-hour experiment was conducted, at the end of which the subject was supposed to produce one (or several) parti proposals.

Among the interesting conclusions drawn by Foz from his experiment is the importance of simulation during the parti design as a way of making decisions. Design was better characterized 'as a learning activity than as an analytic dissection of a formal problem'. If human information processing capacities seem to be the same for individuals, the performance of the skill designer relies in part on his ability to organize knowledge in well structured chunks and to use these chunks in an efficient sequence. Skill designers make more tests on the ideas that occur to them and tend to delay the arrival at a building form proposal. They use three-dimensional representation often, not as a display of a completed design proposal but as a part of the information process.

Such a study seems to have great significance if we really want to inquire as to what could be a reasonable use of automatized eethods during the design process. The idea of computer-aided design has to be discarded. The use of these methods cannot be as an exterior and miraculous help whose effectiveness is dependent on the state of the art of a discipline which has nothing to do with architecture (computer science). A correct functioning of automatized methods within the design process presupposes some inquiries regarding a satisfying cooperation with the human designer. The conditions of the insertion of these techniques is not clear. Shall we consider them as an extension of the designer capabilities? Shall we use them to speed some part of the resolution process?

Will they try to replicate a human behavior or will they participate in the creative part of the information process? Any answer to those questions will carry with it very different consequences in the way we envisage the implementation and functioning of programs. But the answer will achieve some degree of feasibility if it relies on observations made on human behavior during the design process. Skill, habits and designers' failures are important things to know, and studies on architects' behavior are not negligible.

2.4 From Constraints Definition to Another Approach:

We have begun with very general questions on the problem solving approach and its shortcomings. We will focus our investigation and critique by exploring more limited and technical points. With constraints and representations, the programming conveniences have a stronger influence on the type of constraints considered, as well as on the type of representation chosen.

Constraints define the relationships between elements to be localized. They can also characterize properties of the reception space (with predetermined boundaries, with non-constructible surfaces...) or of the elements. With the earliest space allocation programs, the only constraint taken into account was the quantity of movement to minimize between activities. The quantity of movement was represented in a matrix. Measures made on existing building having the same characteristics as the one being studied were supposed feasible enough to fill the matrices (Eldars's Program , CRAFT). This first and rough modelization was followed by some attempt to formalize less quantitative relationships. Eldars (Whitehead,Eldars,1964,3.3) introduced the nuisance relationship (compatibility of pair of activities in the generation of and tolerance to various forms of nuisances such as noise, smell, etc.). Nuisances are expressed in a boolean matrix, compatibility or incompatibility between two element involving a specific localization algorithm. Bernholtz in Lokat takes into account several sorts of constraints; some are quantifiable and directly expressed in matrices; others more qualitative necessitated a translation process. The generation of plans are made from one constraint, and solutions are tested against other criteria (a ponderation of the criteria is done depending on its relative importance). With GSP (Eastman,1971a,3.3), IMAGE (Weinzapfel,Johnson,Perkins,1971,3.3) and DPS (Pfeffercorn,1971,3.3), the number of constraints increases in proportion to the difficulty in satisfying them. In GSP, a degree of restriction is attached to each constraint; to locate an element the operator dealing with the most restrictive relationship (for example adjacency against orientation by ex.) is applied first. Then the operator dealing with the second most restictive constraint is applied and so on.

IMAGE uses an optimization process called Least Mean Square Fit to compromise among different violated relationships between a space and the other spaces of an arrangement. IMAGE can deal with various relationships such as proportion, area, alignment, visual access, etc. In DPS, a function for evaluating the constraint difficulty before entering objects and a function for evaluating the constraint when objects are in the layout are associated with each constraint. DPS deals with constraints such as distance, position, orientation, and view.

Some general remarks can be made. The first program dealt only with quantifiable aspects of relationships between spaces. The expression of the circulation function was simply reduced to the number of steps between 'activities', with the corresponding amount of dollars attached to the people circulating. The minimization of circulation between activities was modelized only because of an oversimplification of the problem. The definition of activities as well-defined functions of a building which have to be localized within a well-defined piece of space is non-satisfying. Even inside that more restricted frame, the circulation modelization is questionable. The data preparation is mainly performed through observations and measurements made in similar buildings which does not involve the correctness of data for the problem under study. Activities are supposed non-changing over time and space and the interrelations between them constant. The possibility of overlapping activities or the obsolescence of some of them is not considered. The model utilized is essentially static. The circulation function itself has been superficially studied and reduced to its quantitative aspect (number of steps and number of dollars). The data collection and its ponderation is doubtful, and the overall idea of treating movement inside a building only in its negative aspect can be discussed (Tzonis,Salama,1972,3.1).

Constraints relate elements two-by-two and are never dependent on the context in which they occur. In Eldars' program the nuisance relationships are considered; either a communication does or does not exists between two adjacent spaces. In the case of smell between two adjacent rooms, for example, the constraint disappears if there is no communication between the spaces. Such relativism of the constraints does not exist. One of the obvious reasons relies on the use of matrices to indicate relationships between spaces.

With the transcription of nuisance relationship into a boolean matrix, as well as the transcription of non quantitative constraints through matrices, contraints which are not quantifiable are translated into numerical values in Bernholtz's program: smell, noise and 'esthetic point of view'). In IMAGE a constraint such as visual access was simply modelized by a pass going from the full face of a space to the full face of another space. We can emphasize the superficial modelization which has allowed the formalization of constraints other than circulation.

The transcription into arbitrary numeric values of non-quantified aspects of a constraint and the static aspect of the constraint's definition are important deficiencies. The last point is directly related to some remarks made on the characteristics of space allocation programs in the above paragraphs: these programs are non-automodifiable, they don't allow a redefinition of the goal through the solving process. Goals, perception and definition of constraints are intrinsically related. The definition of the problem at its inception involves a definitive establishment of the constraints and of the resolution strategies conducting to the satisfaction of those constraints.

A final remark can be made. As the number of constraints increases, the handling of their respective influence on the solution becomes difficult. In a program like IMAGE, which works mainly according to the successive amelioration of a solution against violated constraints, this failure is evident. The successive amelioration of a solution is influenced by the initial configuration of the spaces, by the order in which the spaces are moved and by the order in which the constraints are tested. The satisfaction of a constraint can also involve the violation of another constraint previously satisfied. These points were further discussed by one of the author (Weinzapfel,1973,3.1).

The preceding points have emphasized the difficulty in modelizing relationships which are not directly computable. In the first programs only the distance relationship was optimized. The reduction of the problem allowed the establishment of a unique evaluation function (CRAFT, Eldars's prog.). Such formalization became inadequate with more ill-defined relationships which tend to characterize architectural activities or architectural space. Eldars avoided seriously considering the nuisance relationship by translating it into a simple yes or no value between two elements. Bernolthz introduced different sorts of constraints, but he introduced them through the use of matrices which made it nessary for him to decide on numerical values and ponderation.

Quantitative techniques are simply unsuited to deal with 'humanistic systems' (systems which are too ill-defined to admit precise mathematical modelization). As a reaction against the use of methods developed for dealing with mechanist systems in the analysis of humanist systems, a different approach proposes some means of describing the behavior of too-complex systems to allow precise mathematical analysis. Briefly, we will present Zadeh's approach. The following is mainly a summary of an introductory memo (Zadeh,1972,3.1); more complete and detailed discussions can be found in (Zadeh,1965,3.1).

The main features of the approach are:

(a) The use of so-called linguistic variables in place of or in addition to numerical variables;

(b) The characterization of simple relations between variables by fuzzy conditional statements;

(c) The characterization of complex relations by fuzzy algorithms.

Linguistic or fuzzy variables are variables whose values are sentences in a specified language (e.g., the attribute color is a fuzzy variable whose values are labels of a fuzzy set-as red, blue, yellow. The value of the fuzzy variable height may be: height; tall, not tall, somewhat tall, tall but not very tall, more or less tall. The values are sentences formed from the label 'tall', the negation 'not', the correctives 'and' and 'but' and the hedges 'very', 'somewhat' and 'more or less'. Linguistic variables provide means for an approximate characterization of complex or ill-defined phenomena.

Characterization of simple relations between fuzzy variables by conditional statements:

In a conditional statement

as: IF x is 5 THEN y is 10, x and y are allowed to be variable

as: IF x is small THEN y is very large.

In the case of more complex relations, the characterizations of the dependance of y on x may require the use of fuzzy algorithms. Fuzzy algorithms provide a means of approximate characterization of fuzzy concepts and their interrelations. They can provide effective means of approximate description of objective functions, constraints, system performance, and strategies.

A fuzzy algorithm is an ordered sequence of instructions in which some of the instructions may contain labels of fuzzy sets. A fuzzy algorithm yields an approximate solution of a specified problem. Instructions in a fuzzy algorithm fall into three classes:

(a) assignment statement: $x \underset{\sim}{=} 5$ x=small

(b) fuzzy conditional statement: IF x is small THEN y is large ELSE y is not large

(c) unconditional action statements: multiply x by y, decrease x slightly...

Fuzzy algorithms are classified into several categories, each corresponding to a particular type of application:

The fuzzy definitional algorithms allow the definition of complex, ill-defined or fuzzy concepts in terms of simpler or less fuzzy concepts (fuzzy concepts such as: criteria of performance, soft constraints, measure of complexity etc.); it can also identify whether or not an element belongs to a set, or more generally determine its grade of membership.

The fuzzy generational algorithms serve to generate rather than define a fuzzy set (as generation of hand-written characters and patterns of various kinds, cooking recipes, etc.)

The fuzzy relational algorithms serves to describe a relation or relations between fuzzy variables.

The fuzzy decisional algorithm serves to provide an approximate description of a strategy or decision rule (as crossing an intersection, parking a car, etc.).

The overall preceding approach has been viewed as a means to describe the behavior of systems 'which are too complex or too ill-defined to admit the precise mathematical analysis'. Fuzzy sets can be considered as a framework allowing uncertainties where the space allocation programs have assigned abusive quantification or definitive space characterizations. Until now, no application of fuzzy sets has been done in the field of our concern; however, it seemed necessary to mention the main lines of an approach which could be able to take in account the imprecise and ill-defined nature of the problems under our investigation.

2.5 Representation of Space:

We will conclude our investigation with some remarks on the difficulties inherent in the representation of space in these programs.

As it has been mentioned in Part I, with the array representation, as the precision of the unit area (domain) increases, the operations become cumbersome. In order to localize a new element, thechecking of empty or non-empty spaces is straightforward, but the operation requires increasing computing time and memory requirements with an accurate representation. It can be noted that such a representation provides redundant information which is not necessarily used during the localization process. On the other hand, the use of a 'grid' representation in the generation process tends to produce arrangements by aggregation of sub-pieces of space. The heterogeneous nature of the means of representation and of the object to be represented is evident.

The above limitations were slightly reduced with the hierarchical array. The subdivision of space is only performed at the boundaries of elements, homogeneous domains are not divided, and an economy of memory and computing time results.

The structuration of domains becomes more interesting if the information on adjacency is immediately available. String representation allows an economic and accurate representation of space (all like points are grouped into a single domain, horizontal distance is real distance). However, the data structure utilized in string representation is not the same in both coordinates and accessing rules are different for each coordinate.

Moran (Moran,1968,7) proposed a representation which utilizes a single addressing rule in both coordinates. The four coordinate boundaries of a block are explicitly represented within it on a list; blocks adjacent to each other in either coordinates are linked. Combination of domains are organized into a lattice. Each node of the lattice represents a block set made up of adjacent smaller domains. With the lattice representation, operations such as searching for a domain or locating an element are complex and slow. They involve the redefinition of boundaries of a set of blocks in the lattice for the insertion of a new element. For example, the domain and adjacent structure can be implemented using the variable sized rectangular domains connected by adjacent relations in the two coordinates ('variable array').

As we have noticed, a better treatment of adjacent relationship is obtained with a list-structured representation. Graph is another type of representation which privileges adjacency relationships.

The above representations express laboriously non-rectangular shapes. In IPS, Pfeffercorn (Pfeffercorn,1971,3.3) proposes a spatial representation in which convex polygons are the primitive ('space blocks'). Each space block is represented as a set of sides, and each side as a set of points. Each element (block, side or point) is represented by a Lisp atom and is described by the atom's property list (the property list of a point contains three attributes: type, xcoord, ycoord). The adjacency information stored in the spatial representation can be directly used to assist the program, in the placing of an object, for exemple. Spatial operators create lists of contiguous space blocks (convex polygons), boundaries of contiguous blocks and corners of boundaries. These likely positions are used in the entering of an object. Objects are entered one space block of the object at a time. With the entering of a new space block, the old layout space block is replaced by two new space blocks. As blocks are entered, new lists of points are constructed. Macro-objects can be built and allow the program to enter groups of objects.

From the array representation to the convex polygons, representation efforts to have a better-structured information of the spatial domains are emerging. However, if DPS allows the manipulation of non-rectangular shapes an plan, the program will treat elements of space in a very close way, similar to the program described above. Programs able to cope with any possible planar configuration do not yet exist.

The insufficiencies of the proposed representations seriously rely on the fact that we really don't have any means to represent architectural space. The use of points, segments or blocks are simply programming facilities which give some way of handling the architectural space, cutting it in sub-pieces and operating on those pieces.

A quite different approach has been proposed with the development of space planning oriented languages. The first attempt in that direction was made by Yessios with Fosplan (Yessios,1972b,3.3). The assumption was that spatial environment hasz a language of its own, spatial configuration being produced by precise syntactic rules. The use of formal grammars refers to a sequential string of alphabetic characters, concatenation being the only operation applied to the characters. This definition is not sufficient to represent spatial configurations. The difficulty was emphasized by Yessios after his first experiment with Fosplan. The system deads mainly with rectangular shapes and generates only rectangular configurations. Constraints on dimensioning and on neighborhood conditions are attached with the rectangles. Composition rules allow the construction of assemblages of rectangles which have the same constraint dimensions and whose neighborhood conditions are compatible. This first program produces plans by concatenation of elements.

In a second system (SIPLAN), Yessios proposes some extensions by introducing several operators allowing the definition of more complex spatial configurations. SIPLAN (Yessios,1975,9) is a site planning system, the global space being a site, the elements to be composed within that site being lots, buildings, roads etc. Elements are composed through the use of particular patterns. Each pattern corresponds to different criteria chosen by the planner (for example, one pattern can provide a central parking pool accessible through a single road with linearly structured lots surrounding the pool; another pattern can provide a continuous road deserving unit lots on each side etc.). Given the shape of a particular site, execution of a predefined pattern is called for with respect to the specific site. The shapes are simply adapted to the shape of the chosen site. The partitioning of an irregular given space with respect to a regularly shaped top level pattern is possible because a variable, rather than a constant value, is given to define the domains. The value of the variable is determined with the matching of the top level pattern against a specific site. Spatial grammars are defined through space grammar (called 'module' in SIPLAN). A grammar is a generative system which derives spatial configurations. A grammar consists of a set of production and a set of specification. The set of specification defines the primitive elements to be composed; they are of two types in SIPLAN:

- The 'domains' which could be any area (lot, garden, parking lots, etc.). With each domain is attached its specification: the lengths of the domain's perpendicular axes (can be constant or range) and the neighborhood conditions which define with what other element(s) the domain can be composed.
- The 'linkages'. A linkage does not exist by itself but is associated with some other element; its shape and length are defined by the shape and length of the element with which it is composed. Only its width is defined. A linkage can have two 'colours': vehicular or pedestrian.

The composition operators define the sequence of composition to be executed. They are: the junction (two rectangular elements are joined along matching sides); the enclosure (an element of polygonal shape is enclosed by a rectangle); the exponentiation (multiple copies of a single element); this operator can be used with an undefined exponent);

and the envelopment (an element is positioned within a rectangular envelope). A production defines the execution of one or more composition operations to derive each new composition. A sequence of production is executed, starting with the start production (top composition which does not accept neighbouring conditions), and replacing the composite names on the right hand side of each production. This is executed until a terminal expression is derived (an expression only composed of primitive names). A global space is given by its corner points. The global space is adjusted to the regularly shaped composite derived by a module. The call command has the following form:

CALL <module>;<space name>.

Another attempt to use the linguistic approach to describe the architectural space has been proposed by Gerzso (Gerzso,1975,9). A general language for spatial organization -SNARQ- is under development. SNARQ is in gestation and we do not want to present its temporary state. The author proposed a first version of the language. The system follows the same basic principles as the one described above. Space is described through modular units. Joining operations permit recursively joining a unit to other modules.

The preceding approaches suffer from the use of one-dimensional grammar in the description of two or three dimentional space. In a second version of SNARQ, the author is presently developing a spatial grammar allowing definition of spatial primitives in two or three dimensions. The joining rules between primitives are expressed through the use of joining sides which do not necessarily correspond to the basic geometric definition of the primitive. A rapid introduction to the multi-dimensional formal languages can be found in Rosenfeld's article (Rosenfeld,1974,9).

The above works are of great interest to us for different reasons. Their attitude is opposite to the attitude of the first programs that we have discussed. They raise the basic questions which should be answered before thinking of any computer-aided tentative design; they intend to investigate some ways of representing the architectural space. The establiment of a language to describe the organization of spaces seems to be a first requirement if we want to avoid the blind generation type of the first space allocation programs. That does not mean that the generation rules of a spatial grammar should replace the architect in the generation of an architectural proposal (this point was strongly emphasized by Gerszo (Gerszo,1975,9)). The architect's responsibility in the design process remains the same, the spatial language proposed being a means of manipulating and describing the architectural space. The description of spaces is no longer totally dissimilar with the object described and participates in the very basic nature of the architectural space.

2.6 Computer Graphics and Architecture:

We will conclude with some remarks on the uses of computer graphics in architecture. We can roughly divide computer graphics application in architectural design into two different, opposing tendencies: the display and manipulation of an object already designed; and the description of an object being designed. Most of the uses of graphic systems in architecture rely on the first philosophy; very little has been done in the second direction, most of it represented by the recent works of Negroponte and of the Architecture Machine Group.

We will be brief, because some of the misadventures of the use of computer graphics in architecture are self-explanatory, and also because some of the points raised in the first part of this chapter can be directly applied to our present inquiry.

In the early sixties SKETCHPAD and SKETCHPAD III inaugurated the first interactive graphic system. Parallel propositions followed, some of which reach a quite sophisticated point (Evans and Sutherland's Picture System, EUCLID...). The use of such systems in computer aided design applications has some particular characteristics. Graphic systems tend to be used simply to display information. In the case of computer-aided design applications, these systems manipulate objects or buildings, the description of which is already computerized and complete. The operations performed on these descriptions are relatively simple: rotating an object, zooming into it, adding or deleting some sub-piece of it. Simple simulation of a building proposal can be obtained; walking through a building or testing different perspective or isometric views are now common operations. The expansion or retraction of geometric objects are also possible (EUCLID). The three-dimensional display of objects is getting more and more realistic; from the 'wire-frame' representation, through solid, half-tone and finally shiny coloured pictures, Utah University's works allowed a nearly perfect replication of reality.

What purpose do all these remarkable technical achievements accomplish? Mainly, they display information; i.e., a finished design proposal that can be tested by such operations as, for example, walking through or around a building.

In a recent article, Sutherland (Sutherland,1975,2.1) emphasizes some of the difficulties of drawing with a computer. His reflections were drawn from some experiments made with SKETCHPAD. The computer can only accept a 'structured' drawing. The description of a drawing is made through a succession of points and contiguous lines organized in a structure (use of ring in the case of SKETCHPAD). The invisible structure of the computer drawings makes them totally different from the designer's drawings. We can make the same remarks as for space representations in space allocation programs: the different nature and properties of the representation and of the drawing being represented. Drawings are evolving steps within the design process. We can recall some of the results of Foz's experiments:

drawings function as an active part of information processing; skill designers do not really use displays of completed drawings. Of greater interest, experienced designers tend to work in three-dimensional rendered drawings. Proposals are translated in perspective, section, furnished floor-plan. These representations are used mostly as cases for judgment and not for display.

HUNCH was designed with a very different state of mind. Sketch recognition was no longer concerned with the rectification gf doubtful lines, but allowed and recognized users' uncertainty and inaccuracy. HUNCH is a first step in the right direction toward a comprehensive man-computer interraction.

The last proposal and works of the Architecture Machine Group tend to concentrate on the interfaces between man and machine (among other objectives that we will not discuss here). The idea is to make the machine extremely familiar with its user, in order to communicate more rapidly, but also to allow the recognition and use of personal experiences as acquaintances People draw differently and have different preferences or limitations in their way of communicating. It is postulated that knowing these differences and responding to them can lead to more effective computer aids. This belief is somehow doubtful and arguable; it tends to confuse the making of an intelligent man-computer cooperation with the making of responsive interfaces. At best a computer able to know and to react correctly to his human partner will not bring any miraculous insight or intelligence to the design problem resolution. The question --avoided or forgotten?-- of the correct functioning of computer-aided design within the design process is not raised. The making of idiosyncratic systems does not (by itself) enlighten the correctness (or incorrectness) of the design problem statement. Perception of human preferences, acquaintances, and design habits can simply reinforce or reflect human habits, hesitancies or skill.

We are somwhat skeptical regarding the future of computer graphics and architectute. Such advanced technologies, as raster scan methods, for example, if they have enthusiastic supporters and allow escape from the structured definition of picture making, cannot bring any insight into the problem of correct functioning of man-computer cooperation.

PART III:

AN EXAMPLE OF A REPRESENTATION FOR KNOWLEDGE:

To present the main tendencies of the computer aided design approach in architecture has been one of the purposes of the thesis. Critisize of the shortcomings, weaknesses or wrong attitudes of this approach has been a necessary consequence of our interest in that domain. The next normal step of a complete analysis should be to propose some search direction. We are somewhat embarassed in this third part of the process; to propose radically new attitudes or miraculous new directions seems doubtful. However, one of the ongoing Artificial Intelligence preoccupations has drawn our attention. With the frame idea, a complete representation of knowledge was proposed. Presentation of some of the main issues attached to the frame approach may shed some light the precedent preoccupations.

Minsky's theory of frame was revealed in a first and somewhat theoretical paper (Minsky,1974,4.4). Frames can be viewed as a method for representing knowledge. Following this first paper several applications were proposed, among them Rubin's thesis on medical diagnosis (Rubin,1974b,4.4), Fahlman's undergoing thesis (Fahlman,1973,4.4), and Winograd's works (Winograd,1974,4.4).

Instead of presenting again Minsky's now quite well known and widely discussed paper, it seemed more instructive to look through the different publications which, although closely related to Minsky's proposal, offer more or less different interpretations and try to implement its main issues with practical examples. Several application fields have have been proposed: medical diagnosis (Rubin,1974a,4.4), recognition of house plants (McLennan,1975,4.4) and more general exploration of recognition problems (Fahlman1973,4.4). A paper drew our attention; it was written a few months ago by Kuipers (Kuipers,1975a,4.4) and proposes ' An hypothesis driven Recognition system for the blocks world'. The completeness as well as the simplicity of the example illustrate quite remarkably most of the basic issues embedded in the frame theory (if a frame theory exists). We will use Kuipers's example extensively as an invaluable aid in the exposition and introduction to the somehow complete way of representing knowledge.

Scene analysis, applied to blocks world domains, has been largely studied (see Part I). Such a program as Waltz's is working in a bottom-up way, from local evidences. The local constraints on the characteristics of an edge connecting two vertices and the addition of shadows considerations considerably reduce the possible interpretations of the nature of lines in a scene. The system works from local evidences to global interpretation of a scene. It collects separate and simple fragments of knowledge on a mini-world --the blocks world. It functions from scratch every time, building a particular knowledge on a particular scene in a sort of blind way which ignores its precedent experiences. In other words, it does not learn. The extension of this approach to more general vision problems is not clear. A eore general knowledge seems to be necessary for the comprehension of complicated scenes. Previously accumulated knowledge and experiences can be essential in the perception of a scene or in the understanding of a discourse. They can speed up the recognition process and from a computational point of view, they can keep some of the work to be performed, done in advance.

Kuipers's program relies on quite different assumptions. It works as an hypothesis-driven recognition process. What does it mean?

A frame is a mechanism for representing knowledge in the computer; especially, a mechanism to organize previously accumulated knowledge and to relate the immediate perception of a scene (or understanding of a story) to the already organized knowledge on that particular domain. In the case of Kuipers's blocks domain, the recognition process will be driven by hypothesis about the object being recognized; the hypothesis will decide which feature to examine and will be confirmed or abandoned depending on the data observed. In the case of contradictions between observations and predictions, one of the main issues will be how to select a new hypothesis.

We will, at first, try to establish the main properties underlying the frame representation. Then we will illustrate the precedent points through Kuipers's recognition program. The following is largely inspired by the two articles (Kuipers,1974a, 1975b,4.4).

A frame functions within a small domain of expertise. It contains the knowledge necessary about the description of an object in that domain (it could be a stereotyped situation such as: going to a birthday party or the different parallepipedes in Kuipers's example below). Attached to a particular domain, some information tells us what observations to make and makes them correspond to the global hypothesis on the domain.

Frames's theory tends to reject the two following explanations of the vision mechanisms:

- there is a global order imposed on the sensory inputs,
- scene analysis can be explained by simple addition of independent and punctual local evidences.

The order of the various sensory inputs we receive is at least partially imposed by what we have learned through experience. Global knowledge contained in a description is coming, in part, from internal representation and not only from the observations. We can cite Minsky 'in my theory; the analysis is based on many interractions between sensations and a hudge network of learned symbolic information'.

Instantiation is the process by which a frame creates a description from observations of an object in its domain. Part of the description is already obtained by selecting a frame. Instantiation is a matching process by which an expected chosen description (a proposed frame) is tested against the observed information.

The selection of the good frame to instantiate is part of the problem. If a piece of information is incompatible with the selected frame, it can be used in the selection of a replacement frame. Part of the already gathered information can be conserved indifferent frames.

If some features have not been observed, the frame can make some predictions regarding the nature of these features. Some 'default values' are given at the terminal of a frame. A value can be weakly bounded to a description (as in: ' John kicked the ball', the color and the size of the ball), or more strongly bounded, as in a line drawing of a cube, a hidden corner and three more faces can seriously be expected. The default values can guide the process of recognizing and instantiating a description by proposing which features to look for.

Changes can cause the perturbation of a frame and of the description it produces (moving around a cube or walking through a room (Minsky 1974)). When a description is partially changed, part of the ancient description can be saved in the network (different frames can share the same terminals).

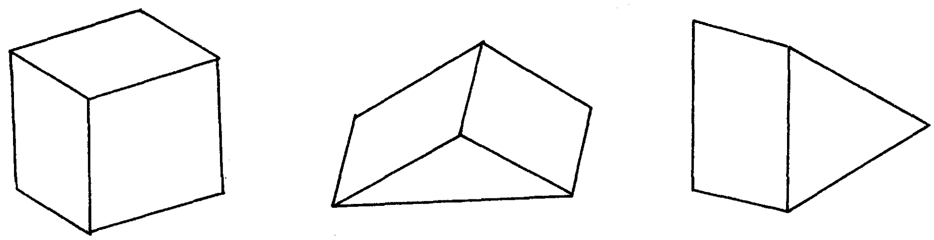
1 Example:

The intuitive example which follows was proposed by Kuipers as an answer to the question: in what are frames, which are a sort of explanation of how people organize their knowledge, of any help in the representation of such knowledge in a computer?

The program looks at the line-drawing representation of a single unoccluded block. It attempts to classify it either as a parallelepiped (with three visible faces) or a wedge which may have two or three visible faces, depending on its position (Fig. 1).

The data furnished the program from the 'sensory world' are a collection of edges and vertices. Each of them can deliver specific information about it and its immediate neighbors. A particular part of the visual scene can only be reached along a known edge from an already examined vertex.

- A vertex delivers its type (L, ⅄ , ↓), the edges which terminate at it and the size of the angles between pairs of edges (acute, right or obtuse); this is a circular search



parallelepiped
with three
visible faces

wedge
with three
visible faces

wedge
with two
visible faces

Fig. 1. The domain.

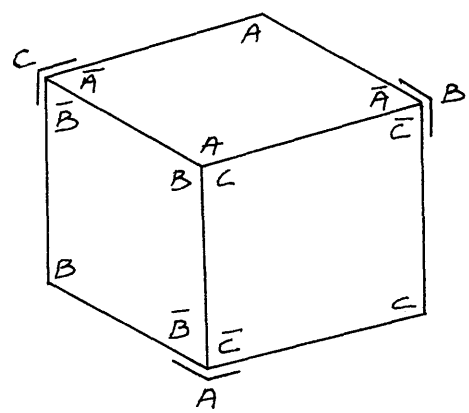


Fig. 2. Global angle relations in the parallelepiped
frame.

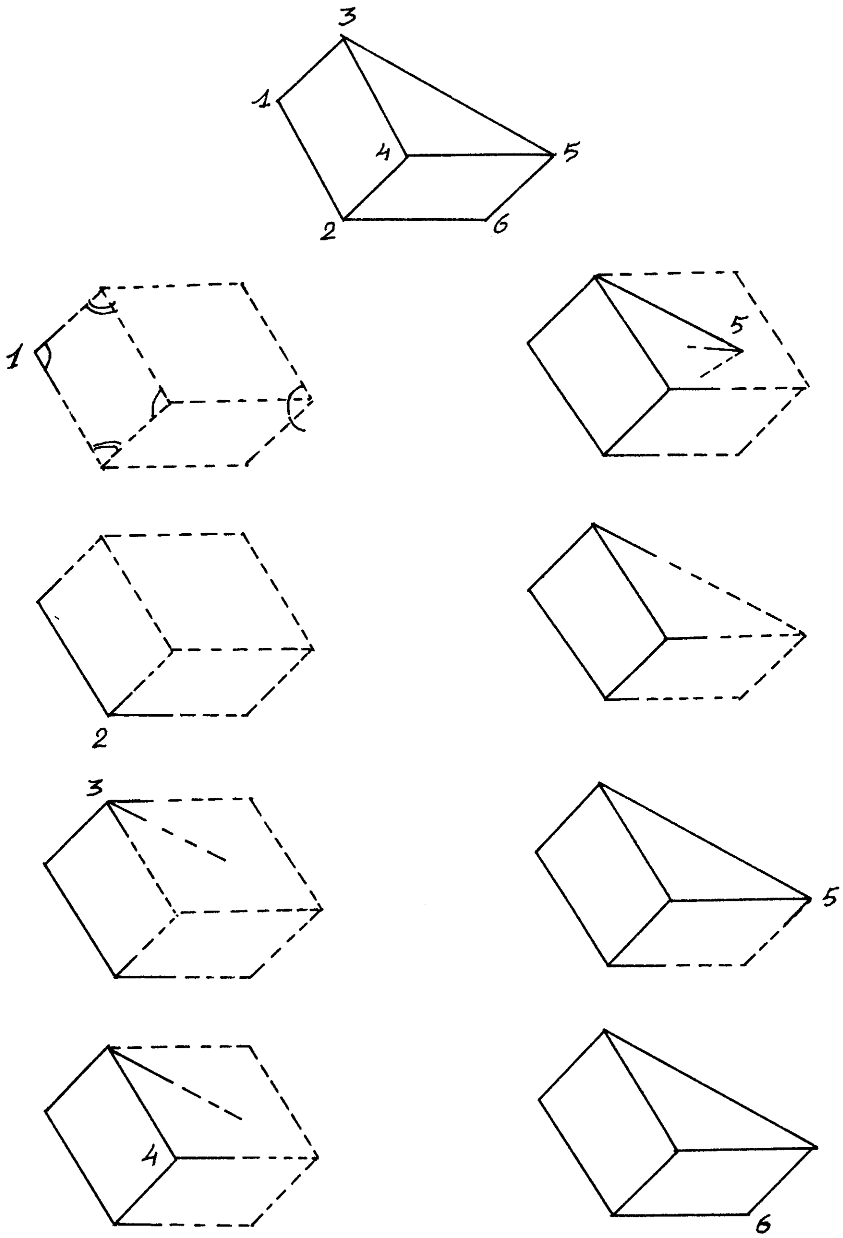


Fig. 3. Stages of the recognition scenario.

in the neighborhood of a vertex.

- An edge delivers its ' other vertex' upon being presented with one vertex. This corresponds to following an edge from one vertex to another.

It can be noticed that the internal representation which has been created is different from the sensory world, such information as a precise angle measurement has been reduced to 'acute, right or obtuse'; people discard many available precisions in the same manner.

Tg this local knowledge about the type of vertices and edges in a figure is added a second knowledge of the global relations among angles in different parts of the drawing (e.g.,: an observed angle measurement can allow the prediction of another measurement in a remote part) (Fig.2).

The description of a line-drawing reveals information not immediately apparent in the scene itself. It imposes a level of organization on the observed data (the statement of the observed object as a 'cube' involves some features which are apparent and some others which are not). A global relational structure is provided between the features (looking at one corner of the cube, one may ask, 'Where is the opposite corner?').

The recognition problem in this blocks world domain is the following: the recognizer has to select and instantiate the correct frame for the drawing. It uses its predictions to guide the recognition. If a conflict occurs between observed data and predictions, a complaint department associated with the frame can select a new frame, if necessary, and some of the previously collected observations can be saved.

What is the recognizer? It consists of three frames, one for each object in the domain. A frame is a program which examines the input data and constucts from that data a description of one of the three types of block. It is a description in that it imposes its global organization on the observed data. It can answer questions about not yet observed features of the scene based on its predictions; its predictions are based on its observations along with its assumptions about the type of object being observed. A frame is more than a description; it contains a strategy knowledge: i.e., which are the best observations to consider when it builds its description. If some inconsistencies appear between the data observed and its assumptions, it can select a new hypothesis and choose another frame.

A scenario of the recognition of a block-drawing will clarify the main characteristics of the recognition process. The object to be recognized is the three-face view wedge (Fig.3) (top-drawing, with the vertices numbered in the order in which they will be explored). The different phases of the recognition process are shown with observed data indicated in solid lines and hypothetical in dotted lines.

Vertex 1: The recognition process is started by giving the program an initial vertex, which is an L vertex. The initial hypothesis is that the figure is a paralellepiped (dotted lines). The simple angle measurement --and the paralellepiped-- hypothesis provides the size of the four indicated angles.

Vertex 2: The second vertex observed agrees completely with the hypothesis (arrow vertex and anticipated measurement for the left side angle of the arrow). The angle measurements provided by the arrow complete the specification of the expected values for all the angles in the figure (the global angle relation allows this prediction).

Vertex 3: This is an arrow vertex (as predicted by the current hypothesis). At this point, the program can't see the angle, which is small and prevents the figure from being a parallelepiped. Its angle resolution is not able to notice the error and the angle specialist accepts the information as consistent.

Vertex 4: The fork-vertex corresponds to the parallelepiped hypothesis. A complete parallelogram face has been explored.

Vertex 5: The L vertex specialist observes an unexpected type of vertex (an arrow instead of an L). The parallelepiped hypothesis breaks down; a transition to the three-face view of the wedge is operated. The correspondance between the cube and the wedge frame allows retention of some of the previously collected data. The selected transformation is executed.

Vertex 6: The remaining vertex confirms the hypothesis of the three-face wedge frame. The frame is fully instantiated.

2 The Recognition Process:

A frame is build around a hypothetical description. It consists of a number of active programs called 'specialists'. These programs interact by sending messages to each other. Each vertex is represented by a specialist in one of the vertex type L, fork and arrow. A vertex specialist has pointers to each of the edges terminating at it.

An edge is represented by a specialist with pointers to its two vertices. This network of specialists connected with pointers represent the topological connectivity of the object.

An initial correspondance established between observations and hypothesis constitutes a prediction of all the vertex type and their connections through the figure. This prediction cannot be changed by incoming data but can be refuted, the frame being replaced by another (example above).

Edges, faces and block, as a whole, are also represented by specialists sending messages to each other.

The recognition process works by selecting and evaluating gbservations with respect to the predictions made by the current frame hypothesis. The program works by sending observations to corresponding specialists for evaluation; predictions and additional data are communicated between specialists.

The recognition strategy can be summarized as follows: an initial hypothesis has been chosen (cube for ex.), the blocks specialist will cycle through its faces telling which, in turn, to select; the faces specialist will cycle through its edges, telling each, in turn, to select the observation. An edge specialist scans from one end of the real edge (already observed) to the other; the newly observed vertex is sent to its corresponding vertex specialist. If the observed vertex is not of the right type, the vertex specialist sends a complaint to the complaint department. If the observed vertex is of the right type, it sends the observed edge and angle measurements to Their respective specialists. Af observed angle measurement is compared to the prediction of the angle specialist, if they are not consistent a complaint is sent to the complaint department.

The complaint department in the frame receives complaints about violated expectations from the vertex and angle specialists.. The complaint department selects the appropriate action: with each anomaly a transition is specified to another frame. We simply give an example of the complaint department behavior in the case of a complaint from a vertex:

Vertex specialist:

expected arrow, got L ⟶ two-face wedge

expected L, got arrow → three-face wedge.

The complaint department is the important part of the recognition scheme; it decides what to do when the predictions are wrong.

With the selection of an appropriate new frame, the delicate point is how to perform the transition between the previous frame and the new one, which prevents the system to function in a blind manner ignoring the previously collected data. The system exploits the similarities between the line-drawings selected under a first hypothesis and the new one. In the case of the transition from a parallepipede to the three-face view of the wedge, only one parallelogram face has to be changed to a triangular one and the angle prediction has to be adjusted. Data already observed are transferred to the corresponding new specialist, the parallelogram face remains valid in the new frame. In the case of the transition to the two-face wedge, the structure of the description is more seriously disturbed (e.g., a face which has two neighbors, now has only one; vertices which expected to be arrow will now be L); there is still an important saving in observations to be investigated, but not as much program structure can be shared between the two-face wedge and the parallelogram as in the above example.

Following the exposition of the preceding example, Kuipers proposed some conclusions which it seems important to summarize.

The representation of the hypothesis is divided, in the blocks domain, between local features (vertex types), fixed global relations (connectivity between vertices and edges represented by the network of neighbor pointers) and predictive global relations (angle specialists which represent local relations among the angle measurements). The distinction between global and local features is made easy by the nature of the domain --blocks world-- where the features are easily separable. That could not be true in other domains.

The manipulation of the hypothesis is performed through the use of 'modules', each having a specific role:

- a module to select the next observation to consider;
- a module to evaluate the observation against its predictions;
- a module to serve as a complaint department (what to do in the case of an anomaly?);
- a module to perform the transition to a new frame.

When selecting the next information, the module has to decide which potential observation would be the most useful at each point in the recognition process. It sends the data to the appropriate specialist. In a not so trivial domain as the blocks-world, several questions could be asked: which observations are the most productive at that time to refine a hypothesis, which parts of the description are more useful than others? In some cases, the importance of these factors can be decided at once; in other cases, they can be re-evaluated depending on the situation.

When evaluating the observations, the frame checks an observation against its hypothesis; this evaluation is performed on local features, through the vertex and angle specialists, which check the consistency of the observed data with their expectation. If the observation is inconsistent with the hypothesis, the specialist sends a description of the problem to the complaint department.

The complaint department is given a description of the current complaint. It has to select a new hypothesis. In the example above, with most of the anomalies. The frame which should replace the current one is specified. The complaint department can also represent the frame range of variation. Certain complaints can be disregarded and certain excuses can be accepted under some circumstances (an 'excuse' allows certain discrepancies between an ideal and its description, such as: like a chair, except in size. It could be a toy chair).

The transition procedures are strongly dependent on the structure of the description. How much can be saved in replacing gne frame with another? The hierarchical structure of the description saves a large sub-structure (as a parallelogram face) which is the same in the two descriptions. In the transition to the three-frame wedge, only a few parts of the top-level description need to be changed. In the transition to the two-face wedge, since the higher structures of the two descriptions are quite different, less of the old description can be preserved.

3 Conclusion:

First, frames are a method of representation of cnowledge. With any representation, the first question is the selection of the relevant knowledge, obviously depending on the domain in which you are working. However, the problem of the choice of the relevant frames, in the case of scene recognition, for example, is not always evident; also, within a specific area, depending on the particular task involved, different types of representation can be chosen. Several frames can express different aspects of a specific domain. Each representation tends to be optimized for the data we are expecting to encounter. The notion of context dependency is one of the basic issues of such a representation; your description will be built in a way which makes it behave differently in different contexts.

Another important aspect of the frame representation is the ability to handle partial or incomplete knowledge. It is more than a static description of something; it tells you what to do with particular links and relations between features.

Part of the interest of such a representation relies on the fact that it works as a verification process of a pressupposed knowledge rather than as a discovery process. This gain of time and of prestructured data organization can be seriously perturbed if the knowledge is strongly .

However, one of the main interests of the frame representation is its debugging knowledge: what to do if the chosen frame does not fit the reality and how to save and transfer the knowledge already collected to your new hypothesis.

As opposed to the use of more and more refined 'methods' to search trough a problem space, the frame theory tends to emphasize the problem of finding good representation:

> " The primary purpose in problem-solving should be better to understand the problem space, to find representations within which the problems are easier to solve. The purpose of search is to get information for this reformulation, not-- as is usually assumed-- to find solutions; once the space is adequately understood, solutions to the problems will more easily be found." (Minsky 1974).

We have decided to complete this analysis by presenting a theory in which flexibility and ability to deal with fuzzy and incomplete knowledge is somewhat opposed to the problem solving approach. This direction of inquiry, through its psychological belief, which seems reasonable, as well as through its formalization exigencies, appears as one of the most promising A.I. preoccupations. Its possible utilization within the field of our interest --automatized methods and architecture-- is certainly not clear and easy. We have briefly thought of such a utilization in the recognition of plans and, as a further use, in the generation of plans in a sort of Golstein's program attitude

(Golstein 1974): from typical nets on specific rooms a matching process could decide on the nature of a room. It would be wise to start with very simple relationships such as: 'needs sunlight in the morning','has to be near the kitchen', 'must have a window'.. Some differences (or discrepancies) will be allowed with a frame of a room. Depending on the importance of the observed difference, a frame can be rejected.

We could imagine a system which could recognize types of rooms in a proposed plan, which could tell us, for example: 'this room is the same as a bedroom, but twice as big as a usual bedroom...'.

To seriously think of the building of such a system could be the propos of another thesis. We wanted to terminate this analysis with a somewhat optimist attitude and simply propose a direction of thought which seems promising and which could permit escape from some of the computer aided design approach limits.

BIBLIOGRAPHY

1 GENERAL

Eastman, C. M., (editor), Spatial Synthesis in Computer-Aided Building Design. Applied Science Publishers. 1975.

March, L.; Steadman, P., The Geometry of Environment. Cambridge, Mass.: MIT Press, 1974.

Milne, M., " From Pencil Points to Computer Graphics" Progressive Architecture, 51, 168-177, June 1970.
Computer Graphics in Architecture and Design, (editor). New Haven: Yale School of Art and Architecture, 1968.

Mitchell, W. J., " Techniques of Automated Design in Architecture: A survey and Evaluation" UCLA, 1974.

Negroponte, N., Anthology, Computer Aids to Design and Architecture. Mason & Lipscomb, 1975.
The Architecture Machine. Cambridge, Mass,: MIT Press 1970.

Spillers, W. R., (editor) Basic Questions of Design Theory. New York: American Elsevier. 1974.

2 COMPUTER GRAPHICS

2.1 GENERAL:

Ahuja, D. V.; Coons, S. A., " Geometry for Construction and Display" IBM Systems Journal, 7, Nos. 3 and 4, 188-205. 1968.

Anonsen, Sheldon Lee " Interactive Computer Graphics in Architecture" Computers and Automation, 19, No. 8, 27-30 August 1970.

Architecture Machine Group. " Artificial Intelligence and Inference Making in Computer Aids to Design" Proposal to the National Science Foundation, 1974.

Bertrom, Dennart Axel An Approach to Computer Graphics in Planning. Stockholm: Statens Institut for Byggnadsfforskning.

Berhnoltz, A.; " Computer Graphic Displays in Architecture" Journal of the Society for Information Display, 3, No. 2, 52-55. March/April 1966.

Blasi; Cesare; Gabriella, " Visualizzazioni con il Computer" Argomenti e Immagini di Design, 4, 43-51, January-February 1972.

Brodey, W. M., " Experiments in Evolutionary Environmental Ecology" Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture, 1968.

Brodey, W. M., Johnson A. R., " A Visual Prosthesis that Looks" Association of Computing Machinery's Second Conference on Visual Prosthesis, Chicago. 1969.

Coons, S. A.,, " Computer Aided Design" Design Quarterly 66/67, 7-14 December 1966a.

" The Uses of Computers in Technology" Scientific American, 215, 177-188, September 1966b.

" Computer Graphics and Innovative Engineering Design" Datamation, 12, N.5, 32-36, May 1966c.

" Computer-Aided Design" Architecture and the Computer. Boston: First Boston Architectural Conference, 26-28. December 5, 1964.

" An Outline of the Requirements for Computer-Aided Design Systems" Cambridge, Mass.: Electronic Systems Laboratory Technical Memorandum 169, MIT. March 1963.

Davies, Ross " Computer Graphic Techniques in Planning" Planning Outlook, 8, 29-34. Spring 1970.

Fetter, W. A., Computer Graphics, Design Quarterly 66/67, 15-24 December 1966.

Computer Graphics in Communications. New York, Mc Graw-Hill. 1964a.

" Computer Graphics: Architecture and the Computer". Boston: First Boston Architectural Center Conference, 34-36, December 5, 1964b.

Giloi, W.; Gnatz, R.; Handler, W., " Gesellschaft fur Informatik" Berlin: Symposium on Computer Graphics 19-21 October 1971.

Gingerich, Jeffrey Z., " Computer Graphics Building Definition Systems" DMG-DRS Journal, 7, 2, 124. April-June 1973.

Graham, B., " Computer Graphics in Architectural Practice" Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture. 1968.

Greenberg, D. P., " Computer Graphics in Architecture"

Scientific American, May 1974.

Hendren, P., " Simulating Architectural Forms" Computer Graphics in Arhitecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture. 38-45. 1968.
" Experiments in Form, Using Computer Graphics" Santa Barbara: University of California 1968.
" A System for Dynamic Simulation, Using Computer Graphics" Stillwater: Oklahoma State Univesity 1967.

Herzog, B., " Computer Graphics for Designers" Emerging Concepts in Computer graphics, D. Secrest and J. Nievergelt (editors). New York: W. A. Benjamin, 189-230 1968.

Hornbuckle, G. D., " The Computer Graphics/Users Interface" IEEE Transactions on Human Factors, HFE-8, 1, 17, March 1967.

Licklider, J. C. R., " Man-Computer Communication" in Annual Review of Information Science and Technology, 3, 201, Cuadra C. A. (editor), Encyclopedia Britannica, Chigago, 1968.
Libraries of the Future. Cambridge, Mass.: The MIT Press. 1965a.
" Man-Computer Communication-- Introduction" Communication Processes, F. A. Geldard (editor). New York: Pergamon. 1965b.
" Problems in Man-Computer Communications" Communication Processes, F. A. Geldard (editor). New York: Pergamon 1965c.
"Man-Computer Symbiosis" Institute of Radio Engineers Transactions on Human Factors in Electronics, HFE-1, No. 1, 4-10 March 1960.

Miller, I. M., " Computer Graphics for Decision Making" Harvard Business Review, 47, 121-132 November-December 1969.

Murray, R. D. (editor) Computer Handling of Graphical Information . Washington, D. C.: Society of Photographic Science and Engineers. 1970.

Negroponte, N., " Idyosincratic Systems- Toward Personnal Computers and Understanding Context" Architecture Machine Group. MIT June 1975.
" Recent Advances in Sketch Recognition" Proceedings of the National Computer Conference, New York, June 4-6 1973.

Newman, W. M.; Sproull, F. R., Principles of Interactive Computer Graphics. Mc Graw-Hill, 1973.

Parslow, R. D.; Prowse, R. W.; Elliot Green , R., (editors) Computer Graphics. London: Plenum Press. 1969.

Phillips, R. L., " Computer Graphics in Urban and

Environmental Systems" Proceedinds of the IEEE, May 1974, Vol.62, No.4.

Proceedings of the IEEE. Computer graphics. April 1974. Vol.62, No.4.

Sider, R. A., et al. Computer Graphics. New York: American Management Association. 1966.

Sutherland, I. E., "Structure in Drawings". "The Hidden-Surface Problem " Anthology, Computer Aids to Design and Architecture, N. Negroponte (editor). Mason & Lipscomb, 1975.
" Computer Displays" Scientific Americain, June 1970.
" Ten Unsolved Problems in Computer Graphics" Datamation, 12, 5, 22, May 1966.

Talbot, P. A.; Carr, J. W.; Coulter, R. R.; Hwang, R. C., Animator: An On-Line Two-Dimensonal Film Animation System April 1971.

Van Dam, A., " Computer Driven Displays and their Use in Man/Machine Interaction" Advances in Computers 7, Alt, F. Z; Rubinoff, M., (editors), Academic Press, 1966.

2.2 CONFERENCE PROCEEDINGS:

Conference on Computer Graphics and Interactive Techniques. Abstracts. July 15-17 1974. University of Colorado, Boulder, Colorado.

Faimar, M.,; Nievergelt, J. (editors), Pertinent Concepts in Computer Graphics, Proceedings of the Second University of Illinois Conference on Computer Graphics, University of Illinois Press, 1969.

Gesellschaft fur Informatik, Symposium uber Computer graphics, Bericht Nr2. Berlin 19-21 October 1971.

I.R.I.A. Colloques 2 "Journees Graphiques" 1973.

Prince, M. D., Man-Computer Graphics for Computer-Aided Design. Institute of Electrical and Electronics Engineers, Proceedings, 54, No. 12, 1698-1708.

Special Workshop on Computer Graphics as Related to Engineering Design. July 12-13, 1973. University of Colorado, Boulder, Colorado.

Symposium on Interactive Computer Graphics. 26-27-28

October 1970. Delft, The Netherlands.

2.3 INPUT HARDWARE AND DISPLAY SYSTEMS:

Davis, M. R.; Ellis, T. O., " The Rand Tablet: A Man/Machine Graphical Communication Device" FJCC 1964, Spartan Books, Baltimore, Md., 325.

Entwisle, J., " An Image Processing Approach to Computer Graphics" Architecture Machine Group. Departement of Architecture, MIT.

Evans and Sutherland Computer Corporation " Line Drawing System Model 1: System Reference Manual" Salt Lake City, Utah 1971.

Girardi, M., " Computer Augmented Drafting System" Environmental Design: Research and Pracice, Proceedings of the EDRA3/ar 8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Keast, D. N., " Survey of Graphic Input Devices" Machine Design, 39, No. 18, 114-120 August 3, 1967.

Miller, S. W., " Display Requirements for Future Man-Machine Systems" IEEE Transactions on Electronic Devices ED-18, 9, 616, September 1971.

Negroponte, N., " Raster Scan Approaches to Computer Graphics" Architecture Machine Group, MIT 1975.

Slottow, H. G., " The Plasma Display Panel-Principles and Prospects" 1970 IEEE Conference on Display Devices, 57.

Stotz, R. " A New Display Terminal" Computer Design Magazine, April 1968.

" Man-Machine Console Facilities for Computer Aided Design" SJCC 1963, Spartan Books, Baltimore, Md., 323.

Stromberg-Carlson. SC 4060 Stored Program Recording System. October 1966.

SC 4020 Computer Recorder. Information Manual. August 1965.

Sutherland, I. E., " Three Dimentional Data Input by Tablet" Proceeding of the IEEE, May 1974, Vol.62, No.4.

" A Head-Mounted Three Dimentional Display" FJCC 1968, Thompson Books, Washington, D.C., p. 757.
" Computer Inputs and Outputs" Scientific American, September 1966.

Teixeira, J. F.; Sallen, R. P., " The Sylvania Tablet: A New Approach to Graphic Data Input" SJCC 1968, Thompson Books, Washington, D.C., 315.

2.4 GRAPHIC LANGUAGES:

Brun, J. M.; Theron, M., " EUCLID un Langage Geometrique Tridimentionel" Journees Graphiques IRIA 17 May 1972a.
" EUCLID ou Comment Definir, Manipuler et Dessiner des Objets Tridimentionnels" Architecture, Mouvement, Continuite. No 24, Janvier 1972b.

Ellis, T. O.; Heafner, J. F.; Sibley, W. L., " THe GRAIL Language and Operations" RAND Report RM-6001-ARPA, September 1969.
" The GRAIL Project: An Experiment in Man-Machine Communications" RAND Report RM-5999 ARPA, September 1969.

Feldman, J. A.; Rovner, P. D., " An Algol-based Associative Language" CACM, 12, 8, 439, August 1969.

Inverson, K. A Programming Language, Wiley, New York, 1962.

Newman, W. M., " A Graphical Language for Display Programming" Specialist Session of International Computer Graphics Symposium, Brunel University, Uxbridge, August, 1968.

Porter, W., " The Development of DISCOURSE: A Language for Computer Assisted City Design" Urban Studies Ph.D Thesis 1969 MIT.

Porter, W.; Lloyd, K.; Fleisher, A., " DISCOURSE: Computer Assisted City Design" Computer Graphics in Architecture and Design. M. Milne (editor). New Haven: Yale School of Art and Architecture, 1968.

Rovner, P. D.; Feldman, J. A., " The LEAP Language and Data Structure" Proceedings 1968 IFIP Congress, Morell, A.J.H., (editor) , North Holland Pub. Co., Amsterdam, 579, 1969.

2.5 HIDDEN-LINE AND HIDDEN-SURFACE ELIMINATION ALGORITHMS:

Archuleta, M., " Hidden Surface Line Drawing Algorithm" Computer Science Departement, University of Utah, June 1972, UTEC-CSc-72-121.

Bui-Tuong, P., " Illumination for Computer Generated Pictures" University of Utah, Communication of the ACM, June 1975, Vol 18, N6.

Encarnacao, J.L., " A Survey of and New Solutions to the Hidden-Line Problem" Proceeding of the Interactive Computer Graphics Conference, Delft, Holland, October 1970.

Evans, D.; Wylie, G.; Romeny, G.; Erdahl A., " Half-tone Perspective Drawing by Computer" American Federation of Information Processing Proceedings, Fall Joint Computer Conference, 31, 46-58 1967.

Galimberti, R., Montanari, U., " An Algorithm for Hidden-Line Elimination" Communications of the ACM, April 1969.

Gouraud, H. " Computer Display of Curved Surfaces" University of Utah, UTEC-CSC-71-113, June 1971 and IEEE, TC-20, p. 623.

Kuberts, B., Szabo, J., Gulieri, S., " The Perspective Representation of Functions of Two Variables" Journal of the ACM, April 1968.

Loutrel, P., " A Solution to the Hidden-Line Problem for Computer-Drawn Polyedra" IEEE Transaction on Computers, March 1970.

Parke, F. I., " Computer Generated Animation of Faces" Computer Science Departement, University of Utah, UTEC-CSc-72-120, June 1972.

Ricci, A., " An Algorithm for the Removal of Hidden-Lines in 3-D Scenes" Comitato Nazionale Energia Nucleare Centrodi Calcolo, Bologia, Italy, November 1970.

Roberts, L. G., " Machine Perception of Three Dimentional Solids" in Optical and Electro-Optical Information Processing, Tipper et al (editor). MIT Press 1965.

Sutherland, I. E.; Sproull, R. F.; Schumaker, R. A.; A Characterization of Ten Hidden-Surface Algorithms, ONR Report.

Warnock, J. E., " A Hidden-Surface Algorithm for Computer Generated Half-tone Pictures" TR 4-15, Computer Science

Departement , University of Utah, 1969.

Watkins, G. S., " A Real-Time Visible Surface Algorithm" Computer Science Departement, University of Utah, UTECH-CSc-70-101, June 1970.

2.6 COMPUTER PROGRAMS:

Bonsteel, D. L., " The Suzzcello Quad: A Computed Graphics Simulation of Sequential Experience" University of Washington Architecture/Development Series 1969.

Coons, S. A., " Surfaces for Computer-Aided Design of Space Forms" Cambridge, Mass.: Project MAC, MIT. June 1967.

Dudnik, E. E., SYMAP User's Reference Manual for Synagraphic Computer Mapping" Report No. 71-1. December 1971. Departement of Architecture. University of Illnois, Chicago, Illinois.

Hodes, I., " Machine Processing of Line Drawings" Lexington, Mass.: MIT Lincoln Laboratory, Group Report 54G0028. March 1961.

Johnson, T.E., " SKETCHPAD III: A Computer Program for Drawing in 3-Dimensions" MIT Electronic Systems Laboratory, ESL-TM-173, June 1963. Also in SJCC 1963, Spartan Books, Baltimore, Md., 347.

Kamnitzer, P.; Hoffman, A., " INTUVAL: An Interactive Computer Graphic Aid for Design and Decision Making in Urban Planning" EDRA Two, Proceedings of the Second Annual Environmental Design Research Association Conference, Pittsburgh, Penn., John Archea and Charles Eastman (editors), 383-390 October 1970.

Kilgour, A. C., " Computer Graphics Applied to Computer-Aided Design" Computer Bulletin, 15, 7, 18, January 1971.

Maxwell, P. C., " The Perception and Description of Line Drawings by Computer" Computer Graphics and Image Processing, 1, No. 1, 31-45 April 1972.

Mehring, H. E., (editor) Interactive Graphics in Data Processing IBM Systems Journal, 7, 3-4 1968.

Negroponte, N., " HUNCH-- An Experiment in Sketch Recognition" Environmental Design: Research and Practice, Proceedings of the EDRA 3/ar 8 Conference, Los Angeles. January 1972.

" HUNCH-- An Experiment in Sketch Recognition" Berlin: Gesellschaft fur Informatik. October 1971.

" URBAN5-- A Machine that Discusses Urban Design" Emerging Methods in Environmental Design and Planning, Gary T. Moore (editor). Cambridge, Mass.: The MIT Press 1970.

" URBAN5: An Experimental Urban Design Partner" Computer Graphics in Architecture and Design, M. Milne (editor). New Haven: Yale School of Art and Architecture, 1968.

Negroponte, N.; Groisser, L. B., " URBAN5: A Machine that Discusses Urban Design" Emerging Methods in Environmental Design and Planning, G.T. Moore (editor). Cambridge Mass.: MIT Press 1970.

" Machine Vision of Models of the Physical Environment" Cambridge, Mass.: MIT, Departement of Architecture, Proposal to the National Science Fundation. 1969.

" URBAN5 " Ekistics, 24, No. 142, 289-291. September 1967a.

" URBAN5: An On-line Urban Design Partner" IBM Report, 320-2012. Cambrige, Mass. June 1967b.

Negroponte, N.; Groisser, L. B.; Taggart, J., " HUNCH: An Experiment in Sketch Recognition" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA 3/ar 8 Conference, University o California at Los Angeles, 22.1 January 1972.

Newman, W. M.," Graphics Systems for Computer-Aided Design" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at los Angeles, 22-6 January 1972.

" Display Procedures" Communications of the ACM, 14, No 10, 651-660 October 1971.

Sutherland, I. E., " SKETCHPAD-- A Man-Machine Graphical Communication System" AFIPS Conference Proceedings, Spring Joint Computer Conference, 23, 329-346. 1963.

" SKETCHPAD: A Man-Machine Graphical Communication System" MIT Lincoln Laboratory TR 296, May 1965.Abridged Version in SJCC 1963, Spartan Books, Baltimore, Md.,329.

Wehrli, R.; Smith, E. F.; Smith, M. J., " ARCAID- The Architect's Computer Graphics AID" Computer Science Departement, University of Utah, UTEC-CSc-70-102, June 1970.

3 SPACE ALLOCATION TECHNIQUES

3.1 GENERAL:

Ainsworth, W. A., " On the Efficiency of Learning Machines " Institute of Electrical and Electronics Efgineers Transactions, Systems Science and Cybernetics, SSC-3, 77-85, November 1967.

Alexander, C., The Timeless Way of Building (in press).
" What Was Design Methodology, Daddy?" Architectural Design. December 1971a.
" What Was Design Methodology, Daddy?" DMG Newsletter, 5, 3 March 1971b.
" Systems Generating Systems" Systemat, 1, 1-11, 1967.
" The Question of Computers in Design" Landscape, 14, No. 3, 6-8. Spring 1965.
Notes on the Synthesis of Form. Cambridge, Mass. Harvard Businesss Press. 1964a.
" A Much Asked Question about Computers and Design" Architecture and the Computer. Boston: First Boston Architectural Conference, 52-56. December 5 1964b.

Anderson, S. " Problem Solving and Problem Worrying" Bloomfield, Mich.: Cranbrook Academy. American Institute of Architects Teachers Seminar, June 1966.

Auger, B., The Architect and the Computer. New york: Preager, 1972.

Barnett, J., " Computer-Aided Building Design; Where Do We Go From Here?" Architectural Record, 141, 219-220. April 1967.
" Computer Revolution: How Does It Affect Architecture?" Architectural Record, 140, 168-170. July 1966.
" Will the Computer Change the Practice of Architecture?" Architecture Record, 137, 143-150, January 1965.

Benthall, J., Science and Technology in Art Today. London: Thames and Hudson.1972.

Berkeley, E. " Computers for Design and a Design for the Computer" The Architectural Forum, 128, No.2, 60-65, March 1968.

Bernholtz, A., " Some Thoughts on Computers" Role Playing and Design Connection, 5, Nos. 2 and 3, 88-91. Winter-Spring 1968.
" Design and the Computer" Toronto, Canada: Southam Business Publications 1965.

Bernholtz, A.; Bierstone, E., " Computer-Augmented Design" Design Quarterly 66/67, 41-52. December 1966a.
" Computer-Augmented Design" Design and Planning 2, M. Krampen and P. Seitz (editors). New York: Hastings house, 41-52 1966b.

Branham, R. " Case for Computerized Design" Industrial Design, 17, 22-29, March 1970.

Broadbent, G., Design in Architecture. New York: Wiley 1973.

Brodey, W. M., " Unlearning the Obsolescent" Architectural Design, 39, 483-484. September 1969.
" The Clock Manifesto" Annals of the New York Academy of Science, 138, 895-899 1967a.
" Soft-Architecture---The Design of Intelligent Environments" Landscape, 17, No. 1, 8-12, Autumn 1967b.
" Unlearning the Obsolescent" Institute of Electrical and Electronics Engineers, Systems, Science and Cybernetics Conference 1966.

Brodey, W. M.; Lindgren, N., "Human Enhancement: Beyond the Machine Age" Institute of Electrical and Electronics Engineers, Spectrum, 5, No. 2, 79-93 February 1968.
" Human Enhancement Through Evolutionary Technology" Institute of Electrical and Electronics Engineers Spectrum, 4, No. 9, 87-97 September 1967.

Burnette, C., " Towards a Theory of Technical Description for Architecture" Proceedings of the First Environmental Design Research Association Conference, H. Sanoff and S. Cohn (editors), Chapel Hill, 1970.

Campion, D., Computers in Architectural Design. London: Elsevier, 1968.

Casalaina, V.; Rittel, H., " Morphologies of Floor Plans" Paper for the Conference on Computer-Aided Building Design 1967.

Cogswell, A. R., " Housing, the Computer and the Architectural Process" Law and Contemporary Problems, 32, No. 2, 274-285. Spring 1967.

Courtieux, G., " Recents Developpements dans les Applications des Ordinateurs a la Construction Civile, a l'Architecture et a l'Urbanisme" Techniques et Architectures, 33, No.4, 44-48 May 1971.
" Vous n'y Pouvez Rien, Petite Histoire de l'Architecte et du Robot" Architecture d'Aujourd'hui, 145, 9-13 September 1969.

" Recents Developpements dans les Applications des Ordinateurs a la Construction Civile, a l'Architecture et a l'Urbanisme" l'Ingenieur Constructeur, 47-57, July-August 1968.

Cousin, J., " Topological Organization of Architectural Space" Architectural Design, Vol XL, October 1970.

Croos, N., " Impact of Computers on the Architectural Design Process" The Architect's Journal, 623, March 22, 1972.

Duprez, G., " Perception de l'Espace par l'Informatique" Bruxelles, Neuf, No. 29, pp. 76-83, January-February 1971.

Eastman, C. M., " Requirements for Man-Machine Collaboration in Design" Environmental Design Research, Wolfgang F. E. Preiser (editor). Proceedings of the Fourth International EDRA Conference. Stroudsburg, Pennsylvania: Dowden, Hutchinson and Ross, Inc. April 1973.

" Adaptive-Conditional Architecture" Design Participation, Nigel Cross (editor). London: Academy Editions, 51-57 1972a.

" Requirements for Men-Machine Collaboration in Design" Pittsburgh: Carnegie-Mellon University, Institute of Physical Planning, Research Report No.35, November 1972b.

" Adaptive Conditional Achitecture" Pittsburgh: Carnegie-Mellon University, Institute gf Physical Planning, Research Report No.24, November 1971.

Fair, G. R.,; Flowerdew, A. D. J.; Munro, W. G.; Rowley, D., " Note on the Computer as an Aid to the Architect" Computer Journal, 9, No.1, 16-20, May 1966.

Farbstein, J., " Definition and Description of Activity" Journal of Architectural Research. Vol. 3, N.1. January 1974.

Frew, R. S.; Ragade, R. K.; Roe, P. H., " The Animals of Architecture" Environmental Design: Research and Practice, Proceedings of the EDRA3/ar 8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Friedman, Y., Realisable Utopias (in press) Society=Environment. Brussels: C.E.A. 1972.

" The Flatwriter: Choice by Computer" Progressive Architecture, March 1971.

Gavin, J. M., " The Social Impact of Information Systems" Computers and Automation, 18, 8, 16-18, July 1969.

Guttridge, B.; Wainwright, J. R., Computers in Architectural Practice, Halstead Press, New York, 1973.

Harper, G. N., (editor), Computer Applications in

Architecture and Engineering, Mc Graw-Hill, New York, April 1970.

Harris, S. A., " Style in Architecture" Computer Studies in the Humanities and Verbal Behavior, 2, 201-212, December 1969.

Hershberger, R. G., " Toward a Set of Semantic Scales to Measure the Meaning of Architectural Environment" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at Los Angeles, 6.4 January 1972

Herzberg, J. G., " Architects Open Dialogue" New York Times, VIII, 1:7 April 21, 1968.

Hormann, A. M., " Machine-Aided Evaluation of Alternative Designs" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at Los Angeles, 22.2 January 1972.
" A Man-Machine Synergistic Approach to Planning and Creative Problem Solving". Int. J. Man-Machine Studies, 3, 167-184, 241-267 1971.

Ingersoll, J., " Computer House: A House to Match Our Age" House Beautiful, 113, 20, 45-55 February 1971.

Jencks, C " Toward the Year 2000" A. A. Quarterly, 1, 56-60. 1971.

Kamnitzer, P. " Computer Aid to Design " Architectural Design, 39, 507-508 September 1969.

Krauss, R.; Myer, J. R., " Design: A Case History" Emerging Methods in Environmental Design and Planning, G. T. Moore (editor). Cambridge, Mass.: MIT Press 1970.

La Montra, " La Sfida Elettronica-Realta e Prospettive dell'uso del Computer in Architecture" Architettura, 15, 558, December 1969.

Lapied, F., " Informatica e Architettura" Architettura 16, 756-759 March 1971.

Lindgren, N., " Art and Technology, Steps Toward a New Synergism" Institute of Electrical and Electronics Engineers Spectrum, 6, No. 4, 59-68. April 1969.
" Human factors in Engineering-- Part II" Institute of Electrical and Electronics Engineers Spectrum, 3, No. 4, 62-72. April 1966a.
" Human Factors in Engineering-- Part I" Institute of Electrical and Electronics Engineers Spectrum, 3, No. 3, 132-139. March 1966b.

" Machine Recognition of Human Language-- Part II" Institute of Elecrical and Electronics Engineers Spectrum, 2, No.3, 45-59. April 1965a.

" Machine Recognition of Human Language-- Part I Institute of Electrical and Electronics Engineers Spectrum, 2, No. 3, 114-136. March 1965b.

Lindheim, R., " Computers and Architecture" Landscape, 14, No. 3, 8-11 Spring 1965.

Ludwig, M. E., " Architecture in the McLunan Age" American Institute of Architects Journal, 48, No. 1, 37-38 August 1967.

" Prejudice and the Computer" American Institute of Architects Journal, 46, No. 1, 70 Jul 1966.

McCulloch, W. S., Embodiment of Mind. Cambridge, Mass: The MIT Press 1965.

Manheim, M. L., " Problem-Solving Processes in Planning and Design" Design Quarterly 66/67, 31-40 , December 1966a.

" Role of the Computer in the Design Process" Building Research, 3, No.2, 13-17, March-April 1966b.

Mann, R. W.; Coons, S. A., Computer-Aided Design. Mc Graw-Hill Yearbook of Science and Technology. New York: Mc Graw-Hill, 1-9 1965.

Meir, R. L.; Duke, R. D., " Electronic Computer: A new Tool for the Artist" Arts Canada, 24, 20-21 February 1967.

" Conversation with a Computer" Datamation, 13, No.. 1, 57-58 January 1967.

Mitchell, W. J., " Vitruvius Computatus" Forthcoming Proceedings of the Cambridge Conference on Models in Architecture, September 1973.

Moran, T. P., "(Artificial, Intelligent) Architecture Computers in Design". Architectural Record, March 1971.

Mumford, L., The Myth of the Machine. New York: Hartcourt, Brace & World 1967.

" Utopia, The City and the Machine" Daedalus, 94, No. 2, 271-292 Spring 1965.

Mumford, E.; Ward, T. B.; Bateford, B. T., " Computers: Planning for People" Manchester, England: Manchester Business School 1968.

Myer, T. H.; Krauss, R. I., " Architectural Practice and the Computer Revolution" Architectural Design, 37, No.4, 193-195 April 1967.

Negroponte, N., Soft Architecture Machines. Cambridge, Mass.: MIT Press. 1975.
" Limits to the Embodiment of Basic Design Theory" Proceedings of Symposium on Basic Design Theory, W. Spillers (editor). New York City, May 1974.
" Meaning as the Basis for Complexity in Architecture" Architectural Design, 42, 11, 679-681. November 1972a.
" Aspects of Living in an Architecture Machine" Design Participation, Nigel Cross (editor). London: Academy Editions, 63-67. 1972b.
" The Architecture Machine". Werk August 1971a.
" The Architecture Machine". Architecture and Urbanisme, January-August, 1971b.
" The Semantics of Architecture Machines" Techniques et Architecture, May 1971c.
" The Architecture Machine-- A Mean in Teaching and Research" Institute for Electrical and Electronics Engineers Digest, March 1971d.
" The Semantics of Architecture Machines" Architectural Forum, October 1970a.
" The Semantics of Architecture Machines" Architectural Design (G.B.) September 1970b.
" Architecture Machine" Architectural Design, 39, 510 September 1969a.
" Towards an Humanism Through Machines " Architectural Design, 39, 511-512, Reprinted from Technology Review, 71, No. 6, April 1969, Copyright 1969.. September 1969b.
" Towards an Humanism Through Machines" Technology Review, 71, No. 6, 2-11 April 1969c.
" Towards a Theory of Architecture Machines" American Institute of Architects Journal, 51, No. 3, 71-74, March 1969d.

Negroponte N.; Groisser, L., " Semantica della Macchine per l' Architettura" Parametro, 10, 44-50 1972.
" Environmental Humanism Through Robots" Proceedings of the First Annual Environmental Design Research Association Conference, H. Sanoff and S. Cohn (editors). Raleigh: Design Research Laboratory, North Calorina State University, 1969.

Notes Methodologiques en Architecture et en Urbanismes. 2. Allocation Spatiale. Novembre 1973, Centre M.M.I., Institut de l'Environnement. Paris.

Pask, G., " Learning Strategies, Memories, and Individuals" Cybernetics, Artificial Intelligence, and Ecology, H. W. Robinson and D. E. Knight (editors). New York: Spartan Books, 42-63 1972.
" Architectural Relevance of Cybernetics" Architectural Design, 39: 494-496 September 1969.

" The Conception of a Shape and the Evolution of a Design" Conference on Design Methods, J. C. Jones and D. G. Thornley (editors). Oxford: Pergamon Press, 153-168 1963.

Propst, R., " The Action Office", Human Factors, August 1966.

Quintrand, P., " Considerations Generales sur Informatique et Architecture" Techniques & Architecture, Serie 33e, No. 4, 66-68 (Special) May 1971.

Raman, P. G., " Form, Model and Design Synthesis" Proceedings of the Design Activity International Conference, 1-15 August 1973.

Reitman, W. R., Cognition and Thought, Wiley, New York, 1974.

Rittel, H. W. J., " On the Planning Crisis: Systems Analysis of the First and Second Generation" Bedrifts of Konomen, No 8 pp390-396. 1972.

Rusch, C. W., " On Respgnsive Environments" DMG-DSR Journal, Design Research and Methods, 6, No.1, 14-16 January-March 1972.

Siders, R. A. " Computer-Aided Design" Institute of Electrical and Electronics Engineers Spectrum, 4, No. 11, 84-92. November 1967.

Spillers, W. R., " On the Use of Examples in Adaptative Systems" Cybernetics, Artificial Intelligence and Ecology, H. W. Robinson and D. E. Knight (editors). New York: Spartan Books, 217-223 1972.

" Artificial Intelligence and Structural Design" Journal of the Structural Division Proceedings of the American Society of Civil Engineering, 92, ST6, 491-497 December 1966.

Taylor, F. E., " The Essence of C.A,D." Data Processing, 12, 138-142, March-April 1970.

Teague, L. C., " Research in Computer Applications to Architecture" Computer Applications in Architecture and Engineering, G. N. Harper (editor), Mc Graw-Hill, New York 1968.

Teicholz, E. D., " Architecture and the Computer" Architectural Forum, Vol. 129, pp. 58-61, September 1968.

Tzonis, A.; Salama, O., "Problems of Judgement in Programmatic Analysis in Architecture: The Synthesis of Partial Evaluations" DMG.DRS Journal. Vol.8 No.3, Jul-Sep 1974.

Selected Exercices in Programmatic Analysis in Architecture. Harvard University, Graduate School of Design, 1972.

Weinzapfel, G., " It Might Work, but Will it Help?" Proceedings of the Design Activity International Conference, 1-16 August 1973.

Wellesley-Miller, S., " Self-Organizing Environments" Design Participation, Nigel Cross (editor). London: Academy Editions, 58-62 1972a.

" Control Aspects of Pneumatic Structures" Proceedings of the International Symposium on Pneumatic Structures, Delft, The Netherlands 1972b.

Willoughby, T.; Paterson, W.; Drummond, G, " Computer Aided Architectural Planning" Operational Research Quarterly, Vol 21 No 1, p. 91-98.

Zadeh, L. A., " Outline of a New Approach to the Analysis of Complex Systems and Decision Processes" Memo N. ERL-M342, July 1972. Electronics Research Laboratory, College of Engeneering. University of California, Berkeley.

" Fuzzy Sets", Information and Control, Vol.8, 1965.

Aguilar, R. J., " Architectural Optimization under Conditions of Uncertainty: Stochastic Models" University of Kentucky, Proceedings of the Kentucky Workshop on Computer Applications to Environmental Design, M. Kennedy (editor), 174-178, 1971.

Aguilar, R. J.; Hand, J.E., " A Generalized Linear Model for Optimization of Architectural Planning" American Federation of Information Processing Proceedings, Spring Joint Computer Conference, 32, 81-88 1968.

Analyse des Donnees en Architecture et en Urbanisme. Centre MMI, Institut de l'Environnement, Paris 1972.

Barnett, J., " Computerized Cost Estimating" Architectural Record, V.141, 219-220, March 1967.

Bernholtz, A. " Quantification of the Qualitative Design Process: Planning and Design by Computer" IFIP Lubliana- August 1971.

Botta, H.; Dussud, R.," Cout de la Construction et Ordonnancement" Gamsau: Marseilles, 12 January 1972.

Boudier, J. P.; Charalambides, S.; Fourcade, A. M.; Lafue, G., " Analyse de Programmes d' Allocation Spatiale". Allocation Spatiale. Centre M.M.I., Institut de l'Environnement, Paris, Novembre 1973.

Breuer, M. A., " Recents Developments in the Automated Design and Analysis of Digital Systems" Proceedings of the IEEE Vol. 60 N.1 January, 1972.

Brotchie, J. " A General Planning Model" Management Science, Vol.16, No. 3, 1969.

Burberry, P. " Layout Techniques Linear Programming Simulation Data" The Architects' Journal May 1970. pp. 1203-1216.

Camarero, E. G., " A Mathematical Model for the Automatic Generation of Behaviors" London: The British Computer Society. University of York, International Conference on Computers in Architecture, 72-79, September 20-22 1971.

Campion, D., " Computer-Aided Preparation, Retrieval from and Updating of Architectural Schedules" London: The British Computer Society. University of York, International

Conference on Computers in Architecture, 295-302. September 20-22, 1972.

Cooper, L., " Heuristic Methods for Location-Allocation Problems" SIAM Review (Society for Industrial and Applied Mathematics) January Vol.6, pp. 37-53 A5 1964.
" Location-Allocation Problems" Operations Research 11, No 3, 1963.

Dupagne, A. " Application d'une Methode de Localisation Automatique" Bruxelles, Neuf, No 29, January-February 1971.

Eastman, C. M.," Heuristic Algorithms for Automated Space Planning" IFIP, Lubliana, pp. 27-39, August 1971a.
" Heuristic Algorithms for Automated Space Planning" Proceedings: Eighth Annual Design Automation Workshop, June 1971b.
" Heuristic Algorithms for Automated Space Planning" Second International Joint Conference on Artificial Intelligence. London: Proceedings of the British Computer Society, 27-40, 1971c.
" Representations for Space Planning" CACM, Vol. 13, No.4, April 1970.
" Search Strategies for Space Planning" EIRA 1: Proceedings of the First Environmental Design Research Association Conference 1969.

Eastman, C. M.; Yessics, C. I., " An Efficient Algorithm for Finding the Union, Intersection and Difference of Spatial Domains" Departement of Computer Science, Carnegie Mellon University, September 1972.

Eastman, C. M.; Schwartz, M., " Methods for Treating Variable Shaped Objects in Computer Aided Design" Proceedings of the Design Activity International Conference, 2-8 August 1973.

Edwards, H.; Gillet, B., Hale, M " Modular Allocation Technique" (MAT). Management Science 4, Vol. 17, No.3, November 1970.

England, R. " Planning Complex Building Systems" Nature Vol.229, January 22, 1971.

Elms, D. G., " A Comprehensive Computer-Aided Building Design System" Cambridge, Mass.: MIT, Departement of Civil Engineering, Structures Publications No. 343, June 1972.

Feldt, A. A., " Operational Gaming in Planning and Architecture" Prepared for Presentation at the American Institute of Architects-Researchers Conference, Gatlinburg, Tenn. October 25, 1967.

Fenves, S. J., " Computer Use in Building Design" Building Research, 3, No. 2, 10-12, March-April 1966.

Forrester, J. W., World Dynamics. Cambridge, Mass.: Wright-Allen. 1971a.

" Testimony for the Hearings Before the Ad Hoc Subcommittee on Urban Growth of the Committee on Banking and Currency, House of Representatives, Ninety-First Congress, Second Session on Industrial Location Policy. Washington, D. C.: U.S. Governement Printing Office, 208-209 1971b.

" Systems Analysis as a Tool for Urban Planning" For the Symposium: The Engineer and the City. Washington, D.C.: National Academy of Engineering. October 22-23 1969a.

Urban Dynamics. Cambridge, Mass.: MIT Press. 1969b.

Industrial Dynamics. Cambridge, Mass.: MIT Press. 1961.

Freeman, P.; Newell, A., " A Model for Functional Reasoning in design" Second International Joint Conference on Artificial Intelligence, London: Proceedings of the British Computer Society, 621-633 , 1971.

Fromboluti, C., " The Site Plan Idea Generation System" Proceedings of the Kentucky Worshop on Computer Applications to Environmental Design, M. Kennedy (editor), 1970.

Gavett, I. W.; Plyter, N. V., " The Optimal Assignments of Facilities to Locations by Branch and Bound" Operations Research, Vol. 1, No.2, March-April 1966.

Gero, J. S., " Computer and Design in a Constrained Environment" Sydney, Australia: University of Sydney, Departement of Architectural Science, Computer Report CR11 1970.

Gilbert, E. N.; Pollak, H. O., " Steiner Minimal Trees" Siam J. App. Math Vol.16, No1, pp.1-29, 1968.

Grason, J., " An Appraoch to Computerized Space Planning Using Graph Theory" Proceedings of SHARE-ACM-IEEE, Design Automation Workshop, 170-179, 1971.

Guittet, J., " Sur Une Classe de Problemes d' Affectation" Metra pp. 543-552.

Handler, A. B., Systems Approach to Architecture. New York: American Elsevier 1970.

Hanan, M; Kurtzberg, J., " Force Vector Placement Techniques" IBM Report RC 2843. April 1970.

Harris, B., " On Case of Computer Optimization Relative to Design Methods" Architectural Design, June 1970.

Hau, L.; Wildblood, M. J., " A Simple Optimizing Model of Site Location" Architectural Design, May 1971.

Haviland, D. S., " The Activity Space: A Least Common Denominator for Architectural Programming" Paper Presented at the 1967, A.I.A. Architect-Researchers' Conference.

Hawkes, D.; Stibbs, R., " An Environmental Model: Computer Representation" Architectural Design, May 1971a.
" The Environmental Evaluation of Building 1: Mathematical Model" Working Paper 15, Center for Land Use and Built Form Studies, University of Cambridge, December 1971b.

Heliot, Hublin, " Recherche d'une Methode de Localisation" Architecture d'Aujourd'hui, Octobre 1969.

Hiller, F.S.; Connors, M. M., " Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities" Management Science, Vol.13, No.1, September 1966.

Holloway, D. R., " Some Comparative Measurements for Low-Rise Medium Density Housing" Ekistics 183, February 1971.

Idelsohn, J, M., " Ten Ways to Find the Optimum" Control Engineering, June 1964.

Kaufman-Diamond, S., " On Evaluation of Man-Computer Problems Working Systems" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at Los Angeles, 22-4 January 1972.

Kmetzo, I. " Building Automation" Progressive Architecture, 51, 110, October 1070.

Koopman, T. C.; Beckman, M., " Assignment Problems and the Location of Economic Activities" Econometrica, Vol.25, Nol, January 1957.

LaVine, G., " Computer Aided Solution of Functional Relationships for Architectural Design" Urbana, Illinois: University of Illinois, Coordinated Science Laboratory, Report R-492, October 1970.

Lambert, F., " The Traveling Salesman Problem" Paris: Cahiers du Centre d' Etudes de Recherche Operationelle, 2, 180-191, 1960.

Lawler, E.; Wood, D. " Branch and Bound Methods: A Survey" Operations Research, Vol.14, 1969.

Levin, P. H. " The Use of Graphs to Decide the Optimum Layout of Buildings" The Architect's Information Library, 140, No. 15, 809-815, October 1964.

Lew, P; Brown, P. M., " Evaluation and Modification of Craft for an Architectural Methodology" Emerging Methods in Environmental Design and Planning, G. T. Moore (editor), MIT Press, Cambridge, Mass: 1970.

Lin, S. " Computer Solution of the Traveling Salesman Problem" Bell System Technical Journal Vol.44, pp.2245-2269, 1969.

Loberman, H; Weinberger, " Formal Procedures for Connecting Terminals with a Minimum Total Wire Lenght" Journal ACM Vol. 4, pp.428-437, October 1957.

Lo Dato, V. A., " The Minimization of Distance in Placement Algorithms" The Computer Journal Vol.14, No3, pp.260-262, August 1971.

Loginou, V. N., " The Application of Electronic Computers to Planning" Mekhaniz I Automatiz, Prov1A-VA9, 32-35 1963.

March, L.; Steadman, P., " Electrical Networks and Mosaics of Rectangles" Architectural Design, May 1971a.
" A Set Theoric Description of Built Form" Architectural Design, May 1971b.

Markus, T., " A Doughnut Model of the Environment and its Design" Design Participation, N. Cross (editor). London: Academic Editions, 84-91 1972.

Maroy, J. P., " Description d' Espaces pour les Problemes d'Allocation Spatiale" Allocation Spatiale, Centre MMI, Institut de l'Environnement Paris, November 1973.
" Application de l'Analyse des Donnees a l'Architecture des Batiments Hospitaliers" Techniques et Architectures May 1971.

Meir, R. L., " The Metropolis as a Transactions-Maximizing System" Daedalus, 97, No. 4, 1292-1314, Fall 1968.

Meir, R. L.; Duke, R. D., " Gaming Simulation for Urban Planning" American Institute of Planners Journal, 32, No.1, 3-17, January 1966.

Miller, W. R., " Computer-Aided Space Planning"

Proceedings of SHARE-ACM-IEEE, Design Automation Workshop, 28-34, June 1970.

Mitchell, W. J., " Simple Form Generation Procedures" London: University of York, International Conference on Computers in Architecture, 144-156 September 1972.
" Computer-Aided Spatial Synthesis" 5th Annual Urban Symposium, Papers on the Application of Computers to the Problems of Urban Society. New York: Association for Computing Machinery, 101-121 1970.

Moran, T., " A Model of Multi-Lingual Designer" Emerging Methods in Environmental Design and Planning, G. T. Moore (editor) MIT Press, Proceedings of the Design Methods Group, First International Joint Conference: 1970.

Newell, A, " Heuristic Programming: Ill-Structured Problems" Progress in Operation Research, Vol.III, J. Aronofsky (ed), Prentice-Hall, New Jersey. 1969.

Noble, J.; Turner, J., " Evaluation, Housing Layouts by Computer" Architect's Journal, 315-316. February 10 1971.

Nugent, C. E; Vollman, T. E., " An Experimental Comparison of Facilities to Locations" Operations Research, January-February 1968.

Paterson, J., " Aids to Improve the Designer's Control of Cost and Performance of Buildings and to Speed the Pre-Contract and Post-Contract Processes" Environmental Design: Research and Practice, Proceedings of the EDRA3/ar 8 Conference, W. J. Mitchell (ed), Los Angeles, 1972.

Pereira, L. " Interactive Dimensional Layout Schemes from Adjacency Graphs" Proceedings of the Design Activity International Conference, 2-19 August 1973.

Purcell, P.A.; Wood, J., " Analysis for Computer-Aided Architectural Design" London: The British Computer Society. Ufiversity of York, International Conference on Computers in Architecture, 191-209, September 1972.

Rawlinson, C.; Doidge, C., " Dynamic Space Allocation" Architectural Research and Teaching, April 1971.

Resch, R. D., " The Topological Design of Sculptural and Architectural Systems" Proceedings of the National Computer Conference, AFIPS, 1973.

Roberts, R. D., " Compucon: Computer Aided Information System for Component Construction" Environmental Design:

Research and Practice, Proceedings of the EDRA3/Ar8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Schneider, B. J., " Solving Urban Location Problems Human Intuition Versus Computer" Journal of the American Institute of Planners, March 1971.

Scott, A. J., " Combinatorial Programming, Spatial Analysis and Planning" London: Methuen, 44-45 1971.

Seppanen, J.; Moore, J.M., " Facilities Planning with Graph Theory" Management Science, Vol.17, No4, December 1970.

Shotwell, T. K., " Information Flow in an Industrial Research Laboratory- A Case Study" IEEE Transaction on Engineering Management, February 1971.

Steiniz, C; Rogers, P., " A System Analysis Model for Urbanization and Change: Experiment in Interdisciplinary Education" Cambridge, Mass.: Graduate School of Design, Harvard University, 1968.

Studer, R. A., " The Dynamic of Behavior-Contingent Physical Systems" Ekistics, March 1969.

Teague, L. C., " Network Models of Configurations of Rectangular Parallelepipeds" Emerging Methods in Environmental Design and Planning, G. T. Moore (editor). Cambridge, Mass.: MIT Press, 1970.

Teague, L. C.; Lavette C. J.; Davis C. F., " III Information Systems for Architectural Programming" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at Los Angeles, 19.3 January 1972.

Tenenbaum, J. M.; et al., " A Laboratory for Hand-Eye Research" Information Processing 71, North-Holland Publishing Company, Amsterdam, 1972.

Van Emden, M. H., " Hierarchical Decomposition of Complexity" Machine Intelligence 5, B. Meltzer, D. Michie (editors). New York: American Elsevier, 361-380, 1970.

Vollmann, T. E.; Buffa, E.S., " The Facilities Layout Problem in Perspective" Management Science, Vol.12, No10 June 1966.

Wesolowsky, G.O.; Love, R. F., " The Optimal Location of New Facilities Using Rectangular Distances" Operations Research pp. 124-130, 1970.

Wheeling, R. F., " Heuristic Search-Structured Problems" Progress in Operations Research Vol III, J. Aronofsky (editor) Wiley 1969.

Willoughby, T.; " Evaluating Circulation Performance" Architectural Design, May 1971.

3.3 COMPUTER PROGRAMS:

Agraa, O. M., " Nuisance Restrictions in the Planning of Single Storey Layouts" Building Science, Vol.2, Pergamon Press, pp.291-302, 1968.

Allwood, R. J., " Genesys: A Policy of Software Investment" Proceedings: International Conference on Computers in Architecture, British Computer Society, London, 1972.

Archer, L. B., " Planning Acommodation for Hospitals and othe Transportation Problem Technique" Architect's Journal Vol. 138. pp. 139-142. A5.C2. 1963.

Armour, G. C.; Buffa, E. ., " A Heuristic Algorithm and Simulation Approach to the Relative Location of Facilities" Management Science, Vol. 9, No 2, January 1963.

Atkins, R. J.; Shriver, R. H., " New Approan to Facilities Location" Harvard Business Review, pp. 70-79. May-June, 1968.

Bax, M. F.; Kessenger, L. R.; Habraker, N. J., " Computer Aided Evaluation of a Structure for Housing" Proceedings: The Design Activity International Conference, Design Research Society and Design Methods Group, August 1973.

Bazjanac, V., " 30P- An Experiment in Architectural Design" Building Research and Practice (in press).

Bazzarella, L. et al., " Strutturazione Programmata di Spazi Elementari" Turin, Instituto di Elementi Costruttivi, Facolta di Architettura, Politecnico, 1971.

Beaumont, M. H., " Computer Aided Techniques for Synthesis of Layout and Form with Respect to Circulation" Unpublished Ph. D. Thesis, University of Bristol, 1967.

Berkeley, E., " RUMOR, the Random Generation and Evaluation of Plans" Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Cambriage, Mass.: Harvard University. 1969.

Bernholtz, A., " A Generalized Program for Transforming Relationship Values into Plan Layouts" Computer Graphics University Brunel. April 1971.
" LOKAT" Laboratory for Computer Graphics and Spatial Analysis, William Warntz (editor). Cambridge, Mass.: Harvard University, IV.4. March 1969.

Bijl, A., " Application of CAAD Research in Practice- A System for House Design" London: The British Computer Society. University of York, International Conference on Computers in Architecture, 286-294, September 20-22 1972.

Billon, R., " La Conception Automatique dans les Systemes Modulaires en Architecture" Gamsau, May 1971.

Blasi, C.; Galimberti, R.; Padovano, G.; Delotto, I., " A Program for Computer-Aided Urban Planning" Calcolo, 5, 229-249. April-June 1968.

Brooks, R. L.; Smith, C. A.; Stone, A. H. ; Tuite, W. T., " The Dissection of Rectangles into Squares". Duke Mathmatical Journal, Vol. 7. pp. 312-340. A5. 1940.

Buffa, E. S.; Armour, G. C.; Vollmann, T. E., "Allocating Facilities with CRAFT" Harvard Business Review. March-April 1964.

Chalmers, J., " The Development of CEDAR" Proceedings of the International Conference on Computers in Architecture, British Computer Society, September 1972.

Charalambides, S.; Lafue, G., " Une Methode Heuristique d'Aide a la Conception" Memoire, 3e Cqcle, Juin 1972. Institut de l'Environnement, Paris.

Eastman, C. M., " General Space Planner: A System of Computer-Aided Architectural Design: User Documentation", Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA 3/ar 8 Conference, University of California at Los Angeles, January 1972.

" GSP: a System for Computer Assisted Space Planning" 8th Design Automation Workshop. Atlantic City. 1971a.

(338) " GSP: a System for Computer Assisted Space Planning" Pittsburgh: Carnegie-Mellon University, Institute of Physical Planning, Research Report No. 23. June 1971b.

" LAYOUT: An Algorithm for Finding a Non-overlapping Arrangement of Two-dimensional Objects within a Bounded Space and Meeting a Variety of Constraints" Internal Working Paper, Departement of Computer Science, Carnegie-Mellon University, April 1968.

Ferrari, D.; Bracchi, G., " Cadep: A Language for Describing Two Dimensional Patterns" CACM January 1970.

Fullenwider, D. R., " SP-1--A Computerized Model for Store Planning" Environmental Design: Research and Practice, Proceedings of the EDRA3/AR 3 Conference, W. J. Mitchell (ed), Los Angeles 1972.

Fullenwider, D. A.; Reeder, C. E., " Implementation of a 'Space Planning' System in a Small-Scale Architecture Office" DMG-DRS, 7, 2, 124. April-June 1973.

Habraken, N. " SAR Design Method for Housing: Seven Years of Development in the Real Word" Proceedings: The Design Activity International Conference, Design Research Society and Design Methods Group, London, August 1973.

Hafez, E.; Agraa, O. M.; Whitehead, B., " Automation of Data Preparation in Computer Program for the Planning of Single Storey Layouts" Building Science, March 1967.

Hamer, J.; Milne, M. A.; Mitchell, W. J.; Robinson, J. H.; Teague, L., " Computer Aided Design System (CADS): Program Documentation, School of Architecture and Urban Planning, UCLA, March 1973.

Harper, G. H., " BOP--An Approach to Building Optimized" Proceedings, Association of Computing Machinery 23rd National Conference, 575-583. November 1968.

Herot, C. F., " Using Context in Sketch Recognition" Master Thesis, Electrical Engineering Departement, MIT May 1974.

Hoskins, E. M., " OXSYS: An Integrated Computer-Aided Building System for Oxford Method" Proceedings: International Conference on Computers in Architecture, British Computer Society, london, September 1972.

James, B., " Locating Human Activies" Computer Reports CR 15 and CR 19, Departement of Architectural Science, University of Sydney, 1972.

Johnson, T. E., " Space Arrangement" Architectural Design, 39, 509, September 1969.

Johnson, T. E.; Weinzapfel, G.; Perkins, J.; Ju, D. C.; Solo, T.; Morris, D., " IMAGE: An Interactive Graphics-Based Computer System for Multi-Constrained Spatial Synthesis", Departement of Architecture, MIT September 1970.

Johnson, T. E.; Weinzapfel, G., " Synthese des Espaces a l'aide de l'Ordinateur a Partir des Contraintes Geometriques", Industrial Forum, Vol. 3 Nol, October 1971.

Krawczyk, R.; Elliot, D., " Space Plan: A User-Oriented Package for the Evaluation of and the Generation of Spatial Inter-Relationships" DMG-DRS Journal, 7, No. 2, 124 April-June 1973.

Krejciric, M., " Computer Aided Plant Layout" Computer Aided Design, Autumn, GB, pp.7-19, 1969.
" Computer Aided Building Layout" IFIP North-Holland, Amsterdam pp. 1427-1430. 1968.

Lafue, G.; Charalambides, S., " PAVLOV: A Program Based on a Learning Method as an Aid to Architectural Design" Proceedings of the Design Activity Conference, London, 2-22 August 1973.

Lee, K; Stewart, C. D., " ARK2- An Implementable Computer Aided Design System" London: The British Computer Society, University of York, International Conference on Computers in Architecture, 251-266. September 1972.

Lee, R.B.; Moore, J.M., " CORELAP-Computerized Relationship Layout Planning", Journal of Industrial Engineering, vol. 18, No 3, March 1967.

Levin, P. H., " The Use of Graphs to Decide the Optimum Layout of Buildings" The Arhitect's Journal Information Library, October 1964.

Ligett, R. S., " Floor Plan Layout by Implicit Enumeration" Environmental Design: Research and Practice, Proceedings of the EDRA3/ar 8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Maroy, J. P., " Le Programme URBA", Departement Informatique Appliquee de l'IRIA. 1970.

Milne, M. A,, " CLUSTR: A Program for Structuring Design Problems" Proceedings: The Eighth Design Automation Workshop, Atlantic City, June 1971.

Mitchell, W. J., " The Automated Generation of Architectural Form" 8th Automn. Workshop pp. 193-207. 1971.
" A Computer-Aided Approach to Complex Building Layout Problems" Proceedings of the Second Environmental Design Research Association Conference, C. Eastman and J. Archea (editors), Pittsburgh, 1970.

Mitchell, W. J.; Dillon, R. L., " A Polyomino Assembly Procedure for Architectural Floor Planning" Environmental Design: Research and Practice: Proceedings of the EDA3/ar 8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Mohr, M., " A Computer Model of the Design Process that Uses the Concept of an Apartement Floor Plan to Solve Layout Problems" Cambridge, Mass.: MIT, Master's Thesis in Civil Engineering. 1972a.

" A Computer Model of the Design Process that Uses the Concept of an Apartement Floor Plan to Solve Layout Problems" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA 3/ar 8 Conference, University of California at Los Angeles, 23.6, January 1972b.

Moseley, L., " A Rational Design Theory for Planning Building Based on the Analysis and Solution of Circulation Problems" The Architect's Journal, pp. 525-537. September 1963.

Myer, T., " An Information System for Component Building" EDRA2: Proceedings of the Second Environmental Design Research Association Conference, J. Archea and C. M. Eastman (editors), Pittsburgh, 1970.

Nevill, G. E.; Growe, R. A., " Computer Participation in Conceptual Design" DMG-DRS Journal- Design Research and Methods. Vol 9, N3, Jul-Sep 1975.
" Computer Augmented Conceptual Design" Basic Questions of Design Theory, W. R. Spillers (editor), American Elsevier, 1974.

Newman, W. M., " An Experimental Program for Architectural Design" Computer Journal, Vol. 9, No.1, May 1966.

Oprero, B., " Description of Two Applications to a Mosaic Composition Technique as Employed in the Random Organization of Architectural Layouts " London: The British Computer Society, University of York, International Conference on Computers in Architecture, September 1972.

Ouprerie, J; Pietri, P.F., " Un Outil Informatique d'aide a la Conception CIMBLEZ " Paris, Gamsau Bull. Vol.1 No4, May 1971a.
" Le Programme CIMBLEZ" Architectes, pp, 12-15, February 1971b.

Pack, L; Reinermann, H.; Kiehne, R.; " Space Allocation and Layout" Management International Review, USA/AU, Tome 5. pp.24-35, 1966.

Paterson, J., " An Integrated C.A.D. System for an Architect's Departement" Computer Aided Design, I.P.C. Science and Technology Press, Surrey, 1973.

Pfefferkorn, C., " A Heuristic Problem Solving Design System for Equipment or Furniture Layout" CACM. Vol.18, No 5, May 1975.
" Computer Design of Equipement Layouts Using the Design Problem Solver" Computer Science Departement, Carnegie Mellon University. May 1971.

Purcell, P., " The Cedar System" Environmental Design: Research and Practice, Proceedings of the EDRA3/ar 8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Schnarsky, A. J., " Some Computer Aided Approaches to Housing" Proceedings of SHARE-ACM-IEEE, Design Automation Workshop, 57-67, 1971.

Schumacker, B., " URBAN COGO- A Geographic-Band Land Information System" Proceedings of Fall Joint Computer Conference, 619-630, 1970.

Seehof, J. M.; Evans, W. O., " Automated Layout Design Program", Journal of Industrial Engineering, December 1967.

Simmons, D. M., " One Dimentional Space Allocation: An Ordering Algorithm" 1969.

Souder, J. J., et al. " Planning for Hospital: A Systems Approach with Computer Aided Techniques" American Hospital Association, Chigago, 1964.

Spillers, W. R., " An Algorithm for Space Allocation " Proceedings: Fifth Annual ACM Urban Symposium, New York, August 1970.

Steadman, P., " Minimal Floor Plan Generation" Cambridge University, Center for Land Use and Built-Form Studies. 1971.
" The Automated Generation of Minimum Standard House Plans", EDRA2: Proceedings of the Second Environmental Design Research Association Conference, J. Archea and C. M. Eastman (editors), Pittsburgh, October 1970.

Stewart, C. D.; Lee, K., " Comprospace: Interactive Computer Graphics in the Real World" Environmental Design: Research and Practice, Proceedings of the EDRA3/ar 8 Conference, W. J. Mitchell (editor), Los Angeles, 1972.

Stewart, C. D&; Teicholz, E.; Lee, K., Computer Architecture Programs, Center for Environmenal Research, Boston 1970.

Teague, L. C.; Hershdorfer, A. M., " BUILD: An Integrated System for Building Design" Journal of the Structural Division, Proceedings of the American Society of Civil Engineers, Vol. 95, No. ST4, April 1968.

Teicholz, E., " GRASP: Generation of Random Site Plans" Unpublished Report, Harvard Graduate School of Design, 1969.

Th'ng, R.; Davies, M., " SPACES-An Integrated Suite of Computer Programs for Accommodation Scheduling, Layout Generation and Appraisal of Schools" Computer Aided Design Vol.7 No 2, April 1975.

Veit, F. S., " Analyse-Compose-Display" 8th Automn Workshop pp. 180-192. 1971.

Vollmann, T. E.; Buffa, E. S., " The Facilities Layout Problem in Perspective", Management Science, Vol. 12, No 10, June 1966.

Ward; Wesley, S.; Grant, D. P.; Chapman, A. J. " A PL/I Program for Architectural and Space Allocation" 5th Annual Urban Symposium, Papers on the Application of Computers to the Problems of Urban Society. New york: Association for Computing Machinery, 122-141, 1970.

Weinzapfel, G.; Johnson, T. E.; Perkins, J., " IMAGE: An Interactive Computer System for Multi-Constrained Spatial Synthesis" 8 th Automation Workshop, p, 101-108. 1971.

Whitehead, B.; Eldars, M. Z., " The Planning of Single-Storey Layouts" Building Science, Vol.1, pp.127-139, September 1965.
" An Approach to the Optimum Layout of Single-Storey Buildings ", The Architect's Journal, Vol 139, pp.1373-1380, Vol 139, p.1373-1380, June 1964.

Willoughby, T., " A Generative Approach to Computer Aided Planning: a Theoretical Proposal", Computer Aided Design, Autumn p. 23-37, 1970a.
" Computer-Aided Design of an University Campus" Architect's Journal, March 25, 1970b.

Windheim, L. S.; Negroponte, N.; Flanders, S., "Computer-Aided Hospital Design" Environmental Design: Research and Practice, William J. Mihchell (editor). Proceedings of the EDRA 3/ar 8 Conference, University of California at Los Angeles, 23.8 January 1972.

Yessios, C. I., " Site Planning with SIPLAN" Proceedings of the Design Activity International Conference, 2-9 August 1973.
" Modeling the Site Planning of Homogeneous Uses" Pittsburgh , Pa.: Carnegie-Mellon University, Institute of Physical Planning. Research Report No. 30 April 1972a.
" FOSPLAN: A Formal Space Planning Language" Environmental Design: Research and Practice, William J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at Los Angeles, 23.9, January 1972b.

4 ARTIFICIAL INTELLIGENCE

4.1 GENERAL:

Buchanan, B. G., " Review of Hubert Dreyfus' What Computers Can't Do: A Critique of Artificial Reason" Nov 1972. Computer Science Departement, School of Humanities and Sciences, Stanford University.

Dreyfus, H. L., What Computer Can't Do: A Critique of Artificial Intelligence. Harper and Row. New York, 1972.

Feigenbaum, E. A.; Feldman, J. (editors), Computers and Thought. Mc Graw-Hill. 1963.

Goldstein, I., Understanding Simple Pictures Programs. AI-TR-294, September 1974. Artificial Intelligence Laboratory, MIT.

Minsky, M., (editor), Semantic Information Processing. Cambridge, Mass.: MIT Press. 1968.

Minsky, M.; Papert, S., Artificial Intelligence Progress Report. January 1972. AI Memo No. 252.

Newell, A. ; Simon, H. A., Human Problem Solving. Prentice Hall, 1972.

Nilsson, N. J., Problem-Solving Methods in Artificial Intelligence. Mc Graw-Hill 1971.

Papert, S. " The Artificial Intelligence of Hubert L. Dreyfus: A Budget of Falacies" MIT Artificial Intelligence Memo No 154. January 1968.

Schank, R. C.; Colby, K. M. (editors), Computer Models gf Thought and Language. San Francisco: Freeman. 1973.

Simon, H. A., The Science of the Artificial. Cambridge, Mass.: MIT Press, 1969.

Slage, ., Artificial Intelligence: The Heuristic Programming Approach. Mac Graw-Hill 1971.

Winograd, T., Five Lectures on Artificial Intelligence. Stanford Artificial Intelligence Laboratory. Memo AIM-246. September 1974.

4.2 NATURAL LANGUAGE SYSTEMS:

4.2.1 Early AI Systems:

Bobrow, D., Natural Language Input for a Computer Problem Solving System, M. Minsky (editor), Semantic Information Processing , pp. 134-215, 1968.

Minsky, M.(ed.), Semantic Information Processing, MIT Press, 1968.

Raphael, B., SIR: A Computer Program for Semantic Information Retrieval, M. Minsky (editor) Semantic Information Processing, pp. 33-133, 1968

4.2.2 Second-Generation Systems:

MARGIE: Memory, Analysis, Response Generation, and Inference on English, Advance Papers, Third International Joint Conference on Artificial Intelligence, pp.225-261, 1973.

Winograd, T., Understanding Natural Language, Academic Press, New York, 1972.

Woods, W.; Kaplan, R. M.; Nash-Webber, B., The Lunar Sciences Natural Language Information System: Final Report, BBN Report 2378, Bolt Beranek and Newman, Cambridge, Mass., 1972.

4.2.3 Case Structures:

Fillmore, C., "The Case for Case" E. Bach and R. T. Harms (editors), Universals in Linguistic Theory, New York: Holt, Rinehart and Winston, pp. 1-68, 1968.

Simmons, R., Semantic Networks: Their Computation and Use for Understanding English Sentences, R. C. Schank, K. M. Colby (editors), Computer Models of Thought and Language, pp. 63-113, 1973.

4.2.4 Larger Scale Chuncks of Knowledge:

Abelson, R., The Stucture of Belief Systems, R. C. Schank, K. M. Colby (editors), Computer Model of Thought and Language, pp.287-340, 1973.

Chafe, W., First Technical Report, Contrastive Semantics Project, Dept. of Linguistic, University of California, Berkeley, 1972.

Charniak, E., Toward a Model of Children's Story Comprehension, AI TR-266, MIT Artificial Intelligence Laboratory, Cambridge Mass., 1972.

Schank, R. C., Identification of Conceptualizations Underlying Natural Language, R. C. Schank, K.M. Colby (editors), Computer Models of Thought and Language, pp. 187-248. 1973.

4.3 MACHINE VISION:

Clowes, M. B., " On Seeing Things" AI Journal, Spring 1971.

Clowes, M. B.; Mackworth, A. K.; Stanton, R. B., " Picture as a Problem Solving Process" Laboratory of Experimental Psychology. University of Sussex , England, June 1971.

Guzman, A., " Computer Recognition of Three-Dimentional objects in a Visual Scene" MAC Technical Report 59, Project MAC MIT, December 1968.

Hewitt, C., " Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot" A.I. Technical Report 258, Artificial Intelligence Laboratory, MIT, April 1972.

Huffman, D. A., " Impossible Objects as Nonsense Sentences" Machine Intelligence 6, Edimburgh University Press, 1970.

Minsky, M.; Papert, S., " Progress Report" A.I. Memo. 252, Artificial Intelligence Laboratory, MIT, January 1972.

Sussman, G.; Winograd, T.; Charniak, E., " MICRO-PLANNER Reference Manual" A.I. Memo 203A, Artificial Intelligence Laboratory, MIT, December 1971.

Waltz, D., Doctoral Dissertation and A.I. Technical Report in preparation, Artificial Intelligence Laboratory, MIT.

Winston, P., " Progress in Vision and Robotics" A. I. Technical Report 281, Artificial Intelligence Laboratory, MIT, May 1973.

" Learning Structural Descriptions from Exemples" A .I. Technical Report 231, Artificial Intelligence Laboratory, MIT, September 1970.

4.4 REPRESENTATION OF KNOWLEDGE:

Fahlman, S., "A Hypothesis-Frame for Recognition Problem", Working Paper 57, Artificial Intelligence Laboratory, MIT, December 1973.

Freiling, M. J., " Functions and Frames in the Learning of Structures" Working Paper 58, Artificial Intelligence Laboratory, MIT, December 1973.

Kuipers, B. J., " A Frame for Frames: Representing Knowledge for Recognition" A.I. Memo 322, Artificial Intelligence Laboratory, MIT, March 1975.
" An Hypothesis-Driven Recognition System for the Blocks World" Working Paper 63, Artificial Intelligence Laboratory, MIT, March 1974.

McLennan, M., " Hypothesis and Refutations as an Aid to Computer Vision" Work Performed as a Visitor to the MIT Artificial Intelligence Laboratory, 1975.

Minsky, M., " A Framework for Representing Knowledge" A.I. Memo 306, Artificial Intelligence Laboratory, MIT, June 1974.

Rubin, A. D., " Artificial Intelligence Approaches to Medical Diagnosis" Working Paper 65, Artificial Intelligence Laboratory, MIT, March 1974a.
Hypothesis Formation and Evaluation in Medical Diagnosis. AI-TR-316, Artificial Intelligence Laboratory, MIT, January 1974b.

Theoretical Issues in Natural Language Processing. An Interdisciplinary Workshop in Computational Linguistics, Psychology, Linguistics, Artificial Intelligence. 10-13 June 1975, Cambridge, Mass.

Wilks, Y., " Natural Language Understanding Systems Within the AI Paradigm. A Survey and Some Comparisons" Stanford Artificial Intelligence Laboratory, Memo AIM-237, December 1974.

Winograd, T., " Frame Representation and the Declarative/Procedural Controversy" Draft Version 2, April 25, 1974.

5 REVIEWS:

Bunselmeier, E., Computerized Location-Allocation: Bibliography, School of Architecture, Yale University, 1973.

DMG-DRS Journal, Design Research and Methods, Vol 8, No 62, 1974

Eastman, C. M., " Computer Applications to Architecture in the USA" Computer Aided Design, 6, No. 3, 119-124, July 1974.
" Logical Methods of Building Design: A Synthesis and Review" Institute of Physical Planning Research Report, No. 28, Carnegie-Mellon University, March 1972.
" Automated Space Planning and a Theory of Design: a Review" Pittsburgh: Carnegie-Mellon University, Institute of Physical Planning, Research Report No. 29 January 1972.

Gero, J. S., " The Application of Operations Research to Engineering Design: A Review", Computer Report CR7, Departement of Architectural Science, University of Sydney, 1970.

Hanan, M.; Kurtzberg, J., " A Review of the Placement and Quadratic Assignment Problem" IBM Research Report RC 3406, Yorktown Heights, New York, 1968.

Jones, C.," State of the Art" DMG Newsletter, 5, No. 10,2, October 1971. Design Methods . London: Wiley-Interscience 1970.

MCPBW, Bibliography on the Application of Computers in the Construction Industry, Revised Edition, HMSO, London, 1972.

Olsten, C. J., " A Summary of Architectural Involvement with Computers" Proceedings of SHARE-ACM-IEEE, Design Automation Workshop, 50-55 1971.

Pavageau, F., " Current Computer Applications in the Architectural Field in France" London: University of York, International Conference on Computer in Architecture, 346-353 September 1972.

6 CONFERENCE PROCEEDINGS:

Archea, J.; Eastman, C. M., (editors), Proceedings of the Second Environmental Design Research Association Conference, Pittsburgh, 1970.

Association for Computing Machinery, Proceedings of Annual Urban Symposia.

British Computer Society, Proceedings: International Conference on Computers in Architecture, September 1972.

Design Methods Group and Design Research Society, Proceedings: The Design Activity International Conference, London, August 1973.

Kennedy, M., (editor), Proceedings: Kentucky Workshop on Computer Applications to Environmental Design, University of Kentucky, 1970.

International Conference on Computers in Architecture, University of York. 20-22 September 1972, British Computer Society, London.

Mitchell, W. J., (editor), Environmental Design: Research and Practice, Proceedings of the EDRA3/ar 8 Conference, School of Architecture and Urban Planning, UCLA, 1972.

Moore, G. T., Emerging Methods in Environmental Design and Planning, MIT Press, Cambridge, Mass., 1970.

Preiser, W., (editor), Environmental Design Research, Proceedings of the Fourth International EDRA Conference. Stroudsburg, Penna.: Dowden, Hutchinson and Ross, Inc. April 1973.

Sanoff, H.; Cohn, S., (editors), Proceedings of the First Environmental Design Research Association Conference, Chapel Hill, North Carolina, 1969.

7 DATA STRUCTURE:

Clark, W. E., " The Organization of Data Stuctures" Computer Graphics in Architecture and Design, M. A. Milne, (editor), Yale University, School of Art and Architecture, 1969.

Eastman, C. M.; Fisher, D.; Lafue, G; Lividini, J.; Stoker, D.; Yessios, C., " An Outline of the Building Description System" Research Report No. 50, Institute of Physical Planning, Carnegie Mellon University, September 1974.

Eastman, C. M.; Lividini, J.; Stoker, D., " A Data Base for Designing Large Physical Systems" Research Report No. 53, September 1974. Institute of Physical Planning, Carnegie Mellon University.

Gray, J. C., " A Description of the I* Language" Proceedings: International Conference on Computers in Architecture" British Computer Society, September 1972.

" Fundamental Concepts in Computer Aided Architecture 1: Storage and Data Structure" Working Paper 49, Centre for Land Use and Built Form Studies, Cambridge University, December 1970.

"Compound Data Structures for Computer-Aided Design: A Survey" Proceedings of the 22nd Association for Cgmputing Machinery National Conference, 355-365, 1967.

Johnson, T. E., " A Mass Storage Relational Structure for Computer Graphics and other Arbitrary Data Storage" Cambridge, Mass.: Departement of Architecture, MIT,1967.

Knuth, D. E., The Art of Computer Programming, Chap. 2: Information Structures. Addison-Wesley.

Moran, T. P., " Structuring Three-Dimensional Space for Computer Manipulation" Departement of Computer science, Working Paper, Carnegie Mellon University, Pittsburgh, June 1968.

Page, .; Wilson, ., Information Representation and Manipulation in a Computer. Cambridge University Press, 1973.

Tou, J. T.; Wegner, P. (editors) Proceedings of a Symposium on Data Structures in Programming Languages. University of Florida, Gainesville February 25-27, 1971.

8 PARTICIPATION AND ARCHITECTURE:

Bock, P., " Responsive Democracy: Just Pick up a Phone and Vote!" 6th Annual Urban Symposium, Papers on the Application of Computers to the Problems of Urban Society. New York: Association for Computing Machinery, 43-57 1971.

Brewer, G. D., " Accomodating Increased Demands for Public Participation in Urban Renewal Decision Making" Santa Monica, Calif.: The Rand Corporation, p-4868 July 1972.

Draper, D., " Public Participation in Environmental Decision-Making" Waterloo, Canada: University of Waterloo, Geography Departement, 1973.

Friedman, Y., " Information Processes for Participatory Design" Design Participation, N. Cross (editor). London:

Academy Editions, 45-50, 1972.
" Flatwriter: Voice by Computer" Progressive Architecture, 52, 98-101, March 1971.

Johnson, A. R., " The Three Little Pigs Revisited" Collaborative Design in Community Development, Eleven Views P Batchelor, J.Pearce (editors). Raleigh, North Carolina: North Carolina State University, Student Publication of the School of Design, 20, No. 2, 173-186. 1971.
" Dialogue and the Exploration of Context: Properties of an Adequate Interface" Washington, D. C.: The American Society for Cybernetics, The Fourth Annual International Symposium, 3 October 1970.
" The Self-Organization Interface" Institute of Electrical and Electronics Engineers, Group on Man Machine Systems, September 8-12 1969.
" Self-Organizing Control in Prostnetics" Presented at the Third International Symposium on External Control of Human Extremies, August 25-30 1969.
" Organization, Perception, and Control in Living Systems" Industrial Management Review, 10, No. 2, 1-16. Winter 1969.
" A Structural, Preconscious Piaget: Heed Without Habit" Proceedings, National Electronics Conference, 23 1967.

Krampen, M. " Type Psychology and Representative Citizen Participation in Planning Projects" Proceedings of the Design Activity International Conference, 3-10 August 1970.

Lawhon, J.; Leander, M.; Matsas, N.; Reed, A., " A Playground: Participation in Design" DMG-DRS Journal, Vol.9 No.3. Jul-Sep 1975.

Maver, T., " Simulation and Solution Teams in Architectural Design" Design Participation, N. Cross (editor) London: Academy Editions, 79-83 1972.

May, J. V., " Citizen Participation: A Review of the Literature" Council of Planning Librarians, Exchange Bibliography, 210-211 August 1971.

Mitchell, W. J., " Experiments with Participation-Oriented computer Systems" Design Participation, Nigel Cross (editor). London: Academy Editions, 73-76 1972.

Webber, M. M., " Design Participation " DMG-DRS Journal, 7, No. 1, 60-64 January-March 1973.

9 LINGUISTIC APPROACH:

Alexander, C.; Hirshen, S.; Ishikawa, S.; Coffin, C.; Angel, S., Houses Generated by Patterns. Berkeley, Calif: Ceenter for Environmental Structure 1969.

Alexander, C.; Ishikawa, S.; Silverstein, M., A Pattern Language which Generates Multi-Service Centers. Berkeley, Calif.: Center for Environmental Structure 1968.

Breeding, K. J., PADEL: A Pattern Descriptive Language. Ohio State University AD 713594 June 1970.

Cho, Y., Application of Syntactic Methods to Picture Processing, Ph.D Dissertation, College of Engineering, Berkeley, Ca., 1973.

David, R. E., " Proposal for Diagramatic language for Design" Visible Language, VI Spring, 1972.

Gerzso, J. M., " The Hunting of the SNARQ and other Tales of Computer Languages for Architecture" DMG-DRS Journal Vol 8 No 4, 1975.

Kaneff, S. (editor), Picture Language Machine, Academic Press, London, 1970.

Miller, M.; Shaw, A., " Linguistic Methods in Picture Processing-- A Survey" Proceedings of the Fall Joint Computer Conference, 1968.

Rosenfeld, A., " Multidimensional Formal Language" Basic Questions of Design Theory, W. R. Spillers (editor), American Elsevier, 1974.

Shaw, A. C., " Parsing of Graph-Representable Pictures" Journal of the Association for Computing Machinery, Vol 17, No. 2, July 1970.

Yessios, C. E." Formal Languages for Site Planning" Spatial Synthesis in Computer-Aided Building Design, C. M. Eastman (editor) Applied Science Publishers, London. 1975.

Syntactic Structures and Procedures for Computable Site Planning, Unpublished Ph.D Dissertation, School of Urban and Public Affairs, Carnegie-Mellon University, 1973.

10 SIMULATION:

Bazjarac, V. , " Towards a New Breed of Simulation Models in Architectural Design" Proceedings of the Design Activity International Conference, August 2-24 1973.

Campion, D., " Design Simulation by Computer" Architectural Review, 140, 460. December 1966.

Evans, G. W.; Wallace, G. F.; Sutherland, G. L., Simulation Using Digital Computers. Englewood Cliffs, N.J.: Prentice-Hall 1967.

Parker, R. A., " The Simulation of Pedestrian Behavior in the Central Business District" Proceedings: Kentucky Workshop on Computer Applications to Environmental Design, M. Kennedy (editor), University of Kentucky, 1971.

Shubik, M.; Brewer, G. D., " Models, Simulations and Games---A Survey". Santa Monica, Calif.: The Rand Corporation, A Report Prepared for Advanced Research Projects Agency, May 1972.
" Methodological Advances in Political Gaming: The One-Person Computer Interactive, Quasi-Rigid Rule Game" Safta Monica, Calif.: The Rand Corporation, p-4733, November 1971.
" Systems Simulation and Gaming as an Approach to Understanding Organizations" Santa Monica, Calif.: The Rand Corporation, p-4664, June 1971.

11 BEHAVIORAL SCIENCE:

Brodey, W. M., " Information Exchange Modelled in the Time Domain" General Systems Theory and Psychiatry; W. Gray, F. A. Duhl; N. B. Rizzo (editors). Boston: Little, Brown 1969.
" Developmental Learning and the Education of the Child Born Blind" Etc., A Review of General Semantics, 23, 293-306, September 1965.
" Sound and Space" American Institute of Architects Journal, 42, No. 1, 58-60, July 1964.

Eastman, C. M.; Harper, J., " A Study of Proxemic Behavior: Toward a Predictive Model" Environment and Behavior, Vol. 3, No. 4, December 1971.

Gump, P. V., " The Behavior Setting: A Promising Unit for Environmental Designers" Landscape Architecture, January 1971.

Sanoff, H.; Sawney, M., " A Study of Users Attitudes Toward Their Residential Environment" Environmental Design: Research and Practice, W. J. Mitchell (editor). Proceedings of the EDRA3/ar 8 Conference, University of California at Los Angeles, 13.8 January 1972.

Stea, D., " Mediating Medium" American Institute of Architects Journal, 46, No. 6, 67-70, December 1966.
" Space, Territory and Human Movements" Landscape, 13-16 Autumn 1965.

12 ENGINEERING:

Computer Aided Design Center, Cambridge, Computer-Aided Drafting Systems, IPC Science and Technology Press, Surrey, 1973.

Furman, T. T., (editor) The Use of Computers in Engineering Design. New York: Van Nostrand Reinhold, 1972.

Krauss, R. I.; Myer, T. H., " Computer-Aided Cost-Estimating Techniques" Computer Applications in Architecture and Engineering, G. N. Harper, (editor), Mc Graw-Hill, New York, 1968.

Logcher, R. D., " ICES-STRUDL: An Integrated Approach to a Computer System for Structural Engineering" Emerging Methods in Environmental Design and Planning, G. T. Moore (editor), MIT Press, Cambridge, Mass., 1973.

13 PROBLEM SOLVING:

Dunker, K., " On Problem Solving" Psychological Monographs, Vol. 58, No. 5, 1945.

Eastman, C. M., " On the Analysis gf Intuitive Design Process" Emerging Methods in Environmental Design and Planning, G. T. Moore (editor). Cambridge, Mass.: MIT Press, 21-27 1970.
" Cognitive Processes and Ill-Defined Problems: A Case Study from Design" Proceedings of the International Joint Conference on Artificial Intelligence, 669-690, May 1969.

Foz, A., Some Observations of Designer Behavior in the Parti. Unpublished Thesis, MIT, Departement of Architecture,1972.

Simon, H. A., " The Theory of Problem Solving" Information Processing 71, North-Holland Publishing Company, Amsterdam 1972.

Thorndike, E. L., Human Learning, MIT Press, Cambridge, Mass., 1931.