Towards a design process for micro-based computer-aided drafting systems

by Jeffery Schantz

Bachelor of Fine Arts
Massachusetts College of Art
Boston, Massachusetts
1982

Submitted to the Department of Architecture in partial fulfillment of the requirements of the degree Master of Architecture at the Massachusetts Institute of Technology.

June 1985

© Jeffery Schantz 1985

The Author hereby grants to M.I.T permission to reproduce and distribute publicly copies of this thesis document in whole or in part.

Signature of Author

Jeffery Schantz
Department of Architecture May 8, 1985

Certified by

Imre Halasz
Professor of Architecture
Thesis Supervisor

Accepted by

Julie Meekery
Chairman
Department Committee for Graduate Students
TOWARDS A DESIGN PROCESS FOR MICRO-BASED
COMPUTER-AIDED DRAFTING SYSTEMS

by

Jeffery Schantz

Submitted to the Department of Architecture on May 8, 1985 in partial fulfillment of the requirements of the degree Master of Architecture.

ABSTRACT

Few things will change the architectural profession in the coming years more than the wholesale introduction of computers into the design process that has been made possible by the development of affordable micro-based CAD.

This thesis is a case study into how the current state of the art will affect the process of design. Based on this study, I hope to find out if they will make us better informed as designers or just more productive. How will they help us solve problems and create better environments?

Thesis Supervisor: Imre Halasz
Title: Professor of Architecture
ACKNOWLEDGEMENTS

Thanks to:

Amie for putting up with me
Imre for having faith in me
Steve for the means
Everyone in the computer lab
Rosie, Louie, Jennette, Norman, and Sondra
# Contents

## Part 1
- Introduction
- Assumptions
- Why micro-based CAAD
- A brief description of the software

## Part 2 - Definitions
- Libraries - architectural languages and building vocabulary
- Fragments - assembly and composition
- Rules - ways of ordering
- Consolidation - making buildings

## Part 3 - Documentation
- A computer based design process
- Creating libraries
- Description of the building system
- Rules
- Assembling fragments
- Consolidation
- Program description
- Design documentation
- Conclusions

## Part 4 - a changing environment
- Role of the computer
- Role of the designer

## Appendix
- Three proposals
- Bibliography
- Footnotes
Part 1
Introduction

It is apparent that the micro computer has arrived as a tool for the conception, refinement, and representation of design ideas in general and architecture in particular. I became curious as to how these tools would affect the way designers work and think about the process of design. I was also curious about the formal possibilities that might be revealed by such a process.

My original intention was to engage in a design process using a currently available micro-based software, document the process, then write about the experience. As a result of pursuing this, I've stumbled on a host of other issues.
I began with a lot of assumptions on how the process would happen. The first parts documents those assumptions and seeks to clarify my sometimes archaic language, the second part documents the process. The third part is result of working many hours on a device I essentially did not understand, which makes me sympathetic to others who do the same. The proposals are some thoughts on the state of micro CAD and the direction it might go. My intention in presenting them is to offer a designer's perception of a design system. I'm not a "computer person", in that I do not program or write code. In that way, I consider this to be a kind of "second generation" look at a computer-aided process.

It is inevitable that we will be affected by the advances yet to come. This is not to say that architects should become programmers, but it is imperative that we become informed users.
Assumptions

I began this study under the following assumptions:

1) Computer-aided drafting (CAD) differs from computer aided design (CAAD) in that CAD is a production tool, CAAD is a design tool that helps to solve problems. Currently available programs for micro-computers do not yet qualify as design tools.

2) Extending the role of the micro-computer from that of production assistant to a design assistant is desirable. Developing working methodologies for existing applications software is the most expedient way to extend the computer's role in the process.
3) The development of a process that takes advantage of the software's abilities as a drawing editor holds the greatest promise.

4) Any process involving currently available systems is library based. That is to say that the process revolves around the ability to access and assemble building elements from a database.

5) Because the process is library based, the key to designing in the drawing editor is the development of a "form language" or a "building vocabulary" of elements that can then be deployed in an additive fashion.

6) The assembly of this vocabulary requires that the designer adopt some "rules" that will guide him.

7) Further exploration of these systems by designers as well as programmers will help make a better tool.
Why Micro-based CAAD?

The explosion in the use of CAD programs has been impressive. It is conceivable that coming generations of architects will apprentice on computers instead of drafting tables. This has been made possible by the development of affordable, interactive graphics packages for PC's. Micro-based CAD is modular, and can be supplied by range of hardware and software options. The computers themselves are quickly becoming super-micros, and will continue to become ever more sophisticated. This in conjunction with an increased ability to integrate applications packages will soon give the micro-based system much of the capability of something larger. Eventually the price and capability of both the micro-based and larger stand alone systems will meet.

Another reason for focusing on micro-based systems is more people can
participate in their development. This means that those who use these machines have the opportunity to develop ideas that might otherwise not be developed. Micro computers are affordable, you need not be a large office to own one. As they become more widely used, designers and architects will have to start coming to terms with computers. Until then, we really haven't got a tool.
A Brief Description of the Software

The software used for this exploration was version 2.0 of AutoCad (tm) published by Autodesk, Inc., Sunnyvale, California. It is a menu-driven 2-dimensional, vector based, computer-aided drafting package. It is essentially a drawing editor that has the ability to edit drawing files the way a word processor would handle text. The editor stores and accesses the drawing files from a data base that can be thought of as a library. One can then edit, assemble, and manipulate any of the graphic information in your library. AutoCad allows a certain amount of freedom to "customize" its procedural structure to the preference of the user, by grouping commands into
"scripts" or routines, or by the "customization" of its menu structure. It also has the ability to send alphanumeric information to other programs, by assigning and creating attribute files or by converting a graphic file.

Conceptually, it allows you to do many of the things you would do at a drafting table and a bit more. Graphic information is organized onto layers, similar to using tracing paper. Layers allow for the assignment of colors, linetypes, and other graphic qualities that help to visualize.

Editing functions include the ability to transform or distort any given part by moving, copying, erasing, arrays (repetitive copying), and so on.

Drawing files are created with a series of drawing commands and modes that create lines, arcs, circles, etc. It is important to remember that this system in its' current configuration is really just a sophisticated way of creating, cutting and pasting graphic information. It does not offer a new way to perceive space, it offers a different environment in which design takes place.
Part 2
Definitions

The definitions stated here are an attempt to clarify the language and critical position that became the basis for design in this study. It is therefore necessary to present them before discussing the actual process.
Conceptually, a library is more than a data base. The library is a graphic representation of an architectural or formal "vocabulary". Because it is a kind of "language", it could be said to have symbols, syntax and grammar that form the "rules of composition".

Formulating a vocabulary requires that a building be described in terms of its' elements. It is the articulation and delineation of building elements that sets the framework for the creation of a vocabulary. The operative definition of elements in this study can
Sketches for a building vocabulary

be stated as such - Buildings consist of basically two "types" of elements or parts. The first type could be said to be those elements in a designer's architectural vocabulary that organize buildings. They are the persistent patterns of primary elements that tend to reappear in successive transformations because the patterns of the use of buildings has a subtle continuity that demands familiarity. These elements tend to be primary structural systems, have very long lives, and can be said to be unique to conditions relative to the particular site, or to particular normative or stylistic notions.
Sketches of vocabulary and components
The second "type" of part are components. These are the changeable elements of buildings, ones that wear out or get replaced as the technology advances. They are off-the-shelf parts available from a manufacturer's catalogue or repetitive parts that can be mass produced for buildings large enough to achieve an economy of scale. They are secondary, infill, and closure systems, such as stud partitions, building skins, doors and windows.
Fragments - assembly and composition

Fragments

Fragments are an extension of the vocabulary notion to the middle scale. They become the vehicle for the evaluation of design solutions, allowing the designer to generate, test, and compare alternative possibilities of his vocabulary.

They are patterns and sub-patterns that can be expressed by assembling a vocabulary into "sketches" that suggest larger buildings, but by themselves are not. The concept is appropriate in metaphorical as well as practical terms.

Metaphorically, fragments represent shards of experience. They are pieces in our memory that are selectively recalled, appearing architecturally as incomplete yet familiar places. They become an appropriate vehicle in this environment because they act as a bridge between experience and machine memory. They allow the "assembly of experience" in a manner similar to the assembly of elements in the editor.
There are many ways and sources from which to generate the rules of assembly. By definition in this study, rules are the syntax and grammar of the language. Certain relationships amongst elements are deemed "good" because they meet certain visual, spatial, or architectural criteria. By developing rules, what is added is a means of articulating the relationship amongst elements. They are a means to order, being both criteria and method. What then becomes the issue is a) What is the source of the criteria? and b) What is the method by which they are satisfied?
Consolidation - making buildings

The purpose of consolidation is to try and incorporate as many good (ie: appropriate) ideas as possible into a chosen scheme (1).

The consolidation of design ideas into buildings involves the assembly and elaboration of fragments to an appropriate level of detail. In that way, it becomes an extension of the notions of fragments and building vocabulary, happening at a point when enough possibilities have been explored to satisfy curiosity about the building vocabulary.

Consolidation presents the opportunity to recycle a lot of assumptions and decisions. The process of recycling is one of incorporation and transformation in a progressive fashion.
Part 3
A micro-based computer-aided design process

The process I engaged in happened in five overlapping "phases":

1) the definition of a design strategy that applies at all levels;

2) articulating a vocabulary and building a library;

3) evaluating a vocabulary by the assembly of library parts into fragments,

4) consolidating the fragments into buildings, and;

5) recycling of assumptions about decisions.

Development of a micro-based process required that I familiarize myself conceptually with the drawing editor. Adopting this "protocol" as a method and a resource kept it from becoming a hinderance to the natural flow of thought while designing in the editor. Towards that end, I have found:

1) To a large degree, the process occurred in an additive fashion. Because a lot of what occurs is the selection of parts from a data base, it is very natural to work in an additive way.

2) Since any process involving
Currently available systems are library based, until a library is produced, understood, and made operational, the machine offers little advantage.

3) Although the graphics editors of various CAD systems may differ in size, capability, and speed, conceptually, they all tend to handle information in the same way. The system of layers, menus, libraries, and editing functions seems to have evolved universally. As such, many of the basic concepts should apply to other systems.

4) As a result, the issue then becomes: a) what is an appropriate vocabulary and, b) how to overcome the uncomfortable feeling brought on by an unnatural interface and design following a natural train of thought.

As I began to work with the computer, it became apparent that sometimes were best done on the machine and some off. Most of the conceptual work took place off the computer, while most of the experimenting and testing occurred on it. This became a kind of dialogue that led to many of the decisions I made about what to use and how to use it.
Creating libraries

The selection of a vocabulary can take many forms, depending on how a design strategy is defined. For example, if one is working at a high level of abstraction and little in the way of preconceptions, it is possible to work with a vocabulary of generic shapes that allow ultimate freedom. These shapes could be used to represent building footprints, section cuts, or views in elevation. The shapes can be distorted, rotated, or deployed according to the designers' preference. On the other hand, one could have a highly defined strategy that pre-determines a building system, construction type, style and therefore a vocabulary. It is therefore possible to have libraries suitable for different scenarios.

A library is made operational by giving it a structure and putting in a menu format. There are three general rules I have found helpful when creating libraries:

1) keep the building elements graphically simple. This will keep the drawings from becoming memory intensive, and will allow for further elaboration.

2) create all the elements in plan,
Library structure for a building vocabulary

section and elevation. This will make it easy to jump from view to view.

3) try to loosely define the structure of the libraries in advance of customizing menus. It will simply make things easier to find.

While my assumptions about types of elements in a building vocabulary proved useful in articulating a library, most buildings are actually a little bit of both. One could treat a building as totally unique or as totally off the shelf. The catch being that the former is usually too expensive to build and the latter is too boring to be considered interesting architecture. The library had to include both. Some parts of the vocabulary are unique to the designer, represented in some normative formal ideas and some parts are off-the-shelf stuff of catalogues.
The building system

The building system used for this exploration is a steel frame, with masonry panel and light metal panel skin. The parts of the system were represented as drawing files and delineated into categories of elements:

1) Construction lines

2) Planar elements (treated as composites of materials and representative of a vocabulary of elements of my own preference).

3) A frame system

4) Components (also of my own design, but find their reference in available products).
Mostly due to the need for consistency of dimension in an additive process, I adopted a proportional modular as an ordering system. It seemed appropriate for two reasons. First, because it offered an alternative to the grid provided by the software and, second because this software does not have "a" standard unit of measurement, it has four. It knows one unit that it calls an inch or a millimeter.

By using a proportion as a rule, it was easy to keep the dimensions of the building system consistent from its' creation to its' deployment. The proportions used were the golden mean and the square root of two. This approach allowed me to:

1) distort any of the elements in my library along its X and Y axis in a predictable fashion. If I had a part or a fragment of a given size, I could achieve another compatible size by multiplying or dividing by a known factor. Since all the numbers in the series are known, the "next size" was always available;

2) use regulating lines as a means of ordering. I then could then menu 3 basic configurations, distorting them as needed, creating a range of sizes. This provided a means of achieving dimensional variety, while allowing a visual continuity.
Construction or regulating lines

Planar elements
Rules

In terms of defining my own "rules" or syntax, three design methods and sources of criteria have been helpful in formulating a process of working in the drawing editor. The sources of criteria on which I based the assembly of elements were:

1) criteria that is form dependant, which deals mostly with visual issues;

2) criteria that is building system or construction method specific, which deals with issues of structure and connections, and are inherent in the library elements;

3) criteria that is use dependant, and deals mostly with spatial and territorial issues.

These criteria became the modifiers of a process based on three methodologies:

1) design by selection - outright iteration or imitation;

2) design by borrowing - "selective" selection and transformation of an iterative idea or "pattern matching";

3) design by invention - new variations or "original" ideas.

The computer made it possible to test many variations and combinations of these methods and criteria, by helping me clarify in my own mind the degree to which these "rules" are applied to design decisions that were mostly of a visual or architectural nature.
When working in the drawing editor, the process becomes in some ways iterative and in all ways cumulative. Decisions should be orchestrated because the software remembers the sequence of input and assignment to layers. What this means is that undesirable moves made early on are sure to come back to haunt you.

As a result, there are times when it is more appropriate to select or borrow, then it is to invent. For instance, the creation and assembly of a library differs from the process of selection that occurs when “using” it. Composing a component into an elevation is more a process of “borrowing” or “pattern matching”, that is then transformed.

Pattern matching becomes an appropriate way of designing in this
environment because "it is less a prescription for the perfectly designed environment as it is a convenient format for formulating design concepts" (2). Extending the capabilities of a drawing editor into the developmental stages of the process requires that we have some way of dealing with highly defined elements in the library in a more conceptual fashion, a disparity that pattern matching tends to lessen.
Fragments

Fragments play a number of roles in this kind of process. They are not only a means of evaluating alternatives, they also become the generator of the complete design.

Conceptually, this was where most of the investigation of the "rules of assembly" took place. Since the fragments were less than buildings, it was possible to display a plan, section, and elevation on the screen at the same time. It was also the point where computer proved to be the most valuable. Because alternatives could be quickly and easily explored, it became efficient to evaluate many of them. Consolidation into larger ideas became very easy because I felt confident and well informed about the vocabulary I was using.

The notion of fragments also had a practical side to it. As the memory and speed of micro computers are limited, the assembly of a vocabulary into large buildings can create drawings that are memory intensive, making them slow to regenerate and precarious to save. By creating fragments, the buildings' final design can be consolidated off the machine or can then become part of the library, thereby building a vocabulary of "patterns", typical rooms (such as bathrooms or kitchens), and arrangements of typical rooms.

This interplay between element, fragment, and consolidation became the dialogue for design at all scales.
Pen Plots of fragments
Consolidation

For the aformentioned practical reasons, consolidation of a building designed on a micro-based system is best done off the machine. This problem can be addressed by the addition of memory to the system, but in this case was not an option. There are also some not-so-practical reasons for doing the final stages off the machine. The softwares' practical limitations are in many ways its' artistic limitations.

I found it usefull to be free of the library at this point. Since it is highly memory intensive to depict textual or material qualities in the drawing editor, the fragments remain graphically simple. For myself, this was more a benefit than a hinderance. Keeping the fragments graphically simple allowed elaboration to occur at comfortable point in the process, allowing me to enhance the building studies by traditional drawing techniques.

One of the most important aspects of this phase was the design of the "connective tissue" between the fragments. These areas are the "glue" that holds a design together and are difficult to conceive without having
fragments in front of you. As a result, the study of this "connective tissue" also took place off the computer.

Consolidation also became the means of reworking decisions made in the fragmentary notions. Many of the fragments were designed in isolation from each other, so consolidating was really a means of testing how or if the fragments came together.

Another compelling argument for bringing the design together off the machine is sense of scale. If one is working on a large building or at the site size, it is easy to lose a sense of scale because the image on the monitor is not displayed to scale. By plotting fragments at an architectural scale, consolidation can take place with a better sense of size.

Plan sketch that led to final design
Program description

Since the focus of this study is the process, the program is somehow secondary. The program is for a synagouge located on a rural site in Dover, Mass. It calls for a sanctuary for 350 people, offices, school rooms, a function room, and retreat residences. The site is on a ridge approximately 380 feet in elevation with a commanding view of Boston.

The program was chosen for its size and complexity. A small building is quite easily accomplished on AutoCad. Larger, more involved structures are not. Part of the intention here was to deliberately test the software’s limits.
Plan drawing made by consolidating fragments
Plan drawing after design of “connective tissue” and elaboration
Section at sanctuary as drawn by plotter
Same section rendered by hand
Hand drawn axonometric
Conclusions

The experience of developing a design process for a micro-based CAD system has led me to three conclusions about the current state of these systems, and how they affect the process.

The most obvious conclusion that needs to be stated is the change in thinking that occurs when using a CAD system. The observation is that thinking about how to design a building becomes highly defined and compartmentalized. This is further reinforced by the structured nature of the interface between the designer and the computer.
On the design process

The logical correlation between the conceptual structure of CAD and ideas of building vocabulary and form languages is striking. Having not participated in the actual design of the software, I can’t say if that was an intention. But it certainly appears to be a symptom. The format for accessing and editing graphic information in this environment is largely responsible for this. To that end, I think it is a good thing.

In a lot of ways, one’s ability to take full advantage of this kind of tool is predicated on the delineation of buildings into elements. This fact holds the strongest indication of how the computer will shape the design process in general. As computer-aided techniques become institutionalized, suppliers will have incentive to format products and information about products in a similar fashion.

On micro-based systems

The kind of thinking that we engage in as designers is not as chaotic as we’d like to believe, but it is also not as rational as the computer demands. As a result, a progressive approach towards developing a better micro-based CAAD environment would look at the entire system and it’s interface. Which means that a micro-based systems’ greatest strength is potentially its’ greatest
weakness. With all the various functions being performed by programs from many sources, coordination into an integrated system requires the interface of many packages, with the discovery of routines that will make the system user friendly.

Micro computers will continue to improve and it is my guess that they will become the tool of choice for small architectural offices. In the six-months or so it took to investigate and write this thesis, a multitude of new products has arrived on the scene, indicating an active and growing interest in improving the micro-based design environment.
Part 4
Changing Roles

With the widespread use of computers in the design process, the role of the professional will change. In many ways, a computer becomes like another partner in the process. And like any member of a design team, it will do some things better than others. What those roles are and will become is still evolving. The scope of this study was to take an in-depth look at one of many roles the computer currently plays, production assistant.
Role of the Computer

In order to fully integrate the computer into a process of design, it is helpful to look at the roles the machine currently plays, how those roles affect the allocation of resources, and how it will redefine the role of the designer. A computer-aided design problem can be characterized as consisting of the following (3):

"- a data structure consisting of variables describing the relevant properties of potential solutions,

- a set of operators which may be applied to variables to change the state of a data structure,

- a set of constraints and/or objectives which define...a set of acceptable solutions, and,

- a solution generation procedure which produces a set of potential solutions for consideration."

indicating that a computer may be used for any combination of the following purposes:

"To store and retrieve data...the computer is used as a data processing machine..."
To automatically generate solutions to well defined problems by executing a program which operates upon a data structure...

-To test potential solutions for...consideration...by executing a program that operates on a mathematical model of a proposed design solution..."

An ideal system would to some extent, play all three of these roles.

Currently, most micro-based CAD software concentrates on the most labor intensive parts of the process. Surveys conducted by the AIA (1950, 1970) confirm that nearly 25% of the total time expended on a given project is devoted to the production of working drawings, as compared with about 15% for design. The later survey also noted between 20% - 40% of total cost and effort was reportedly devoted to work by outside engineering consultants. A British study conducted by the Department of Environment (1974), investigating the allocation of man-hrs showed a similar pattern (4).

Yet the ability of an office to achieve the economies made possible by computers is not as simple as just implementing automation. Until the introduction of very sophisticated machines, the process of design may have to be modified somewhat to realize the full potential. Concurrently, one must ask at what price to the quality of design?
This question is presented to introduce the major focus of this study. If the use of the computer can be extended into the range of the design schematics and development phases of a buildings design, it will enhance the process in two ways:

1) it will make the transition from design development to construction documents easier, eliminate redundancy, and help to realize further economy; and

2) since the production phase will not be quite as labor intensive, more time can be spent in the design development phases.

Poorly considered decisions that are made in the earliest stages of the process often end up becoming the most costly miscalculations in the later stages. As such, time gained from the production of construction documents is well spent on design. Hopefully, this is an observation that will prevail and the quality of design should not suffer with the coming of automation. It is this potential increase in productivity that is the strongest incentive to develop a working technique that integrates design development as much as possible with current capabilities of the machine.

The current state of micro-based CAD is such that its strongest role is that of a data processor. Its role as a generator and tester of solutions could quickly become a reality as we enter the era of concurrent operating systems and "expert" software. As such, micro-based CAAD should be thought of in terms of an
integrated system. This study, however, was limited to the process as it relates to working in the graphics editor.

Role of the Designer

As the possible roles computer-aided design systems play increases, the role of the designer will change. For now, computers work nicely for well-defined problems, repetitive tasks, and problems to which typical patterns or patterns of solutions can found. How much of this rubs off on the designer remains to be seen.

What then becomes an issue is: How do we maintain a sense of richness and place in a design environment that could be anywhere, at any time? In the limited course of this study, I have found my ability to deal with that question lies in three sources:

1) the designer him/herself (of course);
2) the library or "vocabulary" of elements;
3) and the development of a design process that can achieve interesting environments from a relatively small vocabulary of elements.

Since the machine could not possibly have the range of experience as its human counterparts, we can't expect to
utilize its memory in the associative ways a designer might. As a result, an overwhelming aspect of design on this current generation of micros is design by selection. A sensitive and selective eye is an asset at all levels of this process. Since the amount of memory is limited, one must be selective about the vocabulary that fills it, just as one must be discerning when choosing an element from that source. Ideally, a sensitive operation would have a number of "appropriate" vocabularies on hand that could be deployed as the situation demanded. Variation in the relationship of the elements then becomes the key to richness.

Architecture is defined by its process. As the process of working with these machines informs the making of them, limitations will turn into advantages. Which starts to point to an evolving role of the designer.

As computers become more institutionalized, the new roles that will emerge will be that of "designer of design systems" and "maintainer of design systems" (5). The first role will be important in that it requires not only a sound and general knowledge in architecture, but a competence in computer sciences and/or applications. This is a good reason why the micro-based systems offer the greatest opportunity to architects. Micros are affordable, accessible, and relatively easy to become fluent on. The ability to integrate roles professionally and technologically will have a revolutionary effect on the practice,
procedure, and quality of our working environment.
Appendix
Three Proposals

The experience of working on these machines has made me appreciate the value of a well thought out interface. These three proposals are observations on some shortcomings of the current working environment, and some thoughts on future capabilities. They are proposals concerning aspects of:

- integration of applications software,
- the man-machine interface
- design intelligence or knowledge.
Integration of application programs

As stated earlier, an integrated micro-based CAAD system consist of more than a drawing editor. It consists of a number of applications packages that share information. In order to build a system that edits, analyses, tests and interprets data, you need at least four programs—a drawing editor, a data base, a computational model in the form of either an "expert" system or spreadsheet, and a text editor. The programs then must be "linked" together, which sometimes requires the writing of routines or at least an intimate knowledge of an operating system. This means that an architects office must acquire and configure these systems themselves.

Another draw back of this arrangement is the discontinuity of the thought process that is brought on by the separation of fact from artifact ie: in order to "test" a solution generated in a drawing editor, you must leave the editor completely and run the appropriate program in a serial fashion. While the coming of concurrent operating systems will help in this respect, they won't help sort things.

In keeping with the modular approach of the micros, it would be useful to have a "tool kit" of "smart" routines that are designed specifically to aid with the exchange of non-graphic data in a CAAD environment. It would help in dealing with one of the paradoxes of an automated design environment—too much information. Designers should concentrate
on design, not information flow. This tool kit would allow the configuration of "filters" that have the ability to:

1) Run constantly as a background operation, updating descriptions,

2) Automatically "route" data to the appropriate file or program,

3) Automatically format it for that application,

4) Allow graphic information to be updated from changes made in a data file.

5) Have the ability to determine the execution sequence of applications programs.

In other words, the "tool kit" would provide routines that make information generated in the drawing editor "reciprocal" with other programs. I believe this will make the interface much easier and the concept of an integrated design assistant more acceptable for micros for three reasons:

1) It would make it possible to have a data base as a background operation that is constantly updated, which would be an enormous improvement on a conceptual level.

2) Most architects are not programmers and cannot afford to be one. This "tool kit" would pre-package routines, allowing users to simply say what, when, where and how by simply installing the module. It would complement an operating system in that it would provide more sophisticated and interactive routines that are designed to support specific program.
configurations.

3) As stated earlier, the coming of automation almost requires the creation of a new role for someone responsible for the maintenance of an office CAAD system. Small offices usually cannot afford the manpower, let alone the cash. It would be this person's tool kit, making the definition and maintenance of a system much easier.

Man-machine interface

Beyond issues that are related to the software of this design environment, are a host of issues relating to how we physically interact with the computer. I have two real problems with currently available hardware.

The first problem is created by the separation of the drawing surface from the viewing area. While that kind of responsive drawing is a desirable technique for life drawing, it is not for design drawing.

The second problem is the lack of a sense of scale that is created by the dimensionless representation of monitor screens. While the instantaneous jump in scale is one of the nicest features of these systems, present display options are simply not adequate. Drawing with a light pen on larger monitors is one way of addressing this, but it is still a long way from ideal. A much more natural interface could be accomplished by combining the viewing screen with the digitizing surface in a large, flat horizontal format that resembles a desk.
The pointing surface would be large, say 72" x 56". The video image would be rear projected on to a touch sensitive device that is activated with cordless pen that can accommodate a variety of media. The tablet software would be calibrated to actual architectural scales, allowing zooming, drawing, and viewing that gives a sense of size. The tablet would also include limited sketch recognition for things such as plumbing and electrical symbols.

In this environment, a designer could draw on paper on his desk at the same time he is encoding a drawing file. It would make it easier to integrate CAAD workstations into the office environment by making them physically easier to deal with.

Design Intelligence

The coming generation of CAAD research will be in areas of design knowledge, method, areas of expertise. In terms of micro-based systems, this indicates that areas of expertise be distilled into "component" programs that can be easily plugged in to a given system. An "expert system" that is meant to aid in the synthesis of design ideas will require the identification and encoding of "design primitives" that become part of a body of design knowledge used in test and generate procedures.

What it means to have design knowledge is not easy to pin down because many issues become inter-related in the designer's mind. What this short
description proposes is a way of encoding a basic knowledge about how to understand spatial relationships. It is based on using notions of territory as a means of synthesizing spatial configurations of graphic "objects".

In the "Hidden Dimension", Edward Hall relates our spatial sense to our territorial needs and how that affects society. In the same way as people, formal elements of buildings have the same kind of public/private "zones". It is this sense of territory that allows a person to understand the relationship between himself and other people.

This "territoriality" also forms the basis for our understanding of the relationship between space, the built environment, and each other. The assumptions in presenting this are:

1) it is a way of describing the spatial aspects of architecture at the most basic level of the interaction between people and their environments.

2) it is a way of describing spaces between objects as a range of dimensions, not absolute dimensions;

3) it describes space as created by form, making the two integral, and,

4) it is applicable at all scales of design.

Zoned Objects

Suppose all the formal objects in a design vocabulary had these characteristics. Objects would be assigned a series of concentric zones. The zones could then be assigned a value and a series of concentric boundaries, making it possible to
recognize when they overlap. The degree of overlap would then be "associated" with a verbal or numerical description of the space created between the two objects. A message would then be passed on to either the user or a routine designed to act on that statement. This would allow the computer to "understand" space in a way similar to the way a person does — as a series of overlapping relationships that are determined by a range of dimensions and not absolute values.

A scenario of how such an idea would work might read like this: Each object would be attributed three zones. The innermost zone being assigned the value "intimate", the second "private", and the third "public". For now, the actual dimensions shall remain arbitrary. When

---

Zoned object

Two objects showing overlapping territories

Pattern of territories created by objects
the two innermost zones of two objects overlap, the space between could be described as intimate, or the two objects could be "understood" to be connected. The machine could then either pass a message or simply connect the two. If zone 1 and 2 overlapped, the space could be described as private, and so on. By "reading" a pattern of overlaps, the machine now has a framework for the understanding of the spatial relationships implied or created by the formal elements of the building. The encoding of the relationships might read something like this:

If zone1=zone1, then connect;
If zone1=zone2, but not 1 or 3, space is intimate;
If zone1=zone3, but not 2 or 1, space is private;
If zone2=zone2, but not 1 or 3, space is semi-private;
If zone2=zone3, but not 1 or 3, space is semi-public;
If zone3=zone3, but not 1 or 2, space is public.
As the spaces are generated by the overlap of territories, a type of use could be assigned to the spaces. As the uses are assigned, the design can be displayed as a pattern of uses. The uses could then be manipulated to achieve "proper" adjacency. This way, the building could be "understood" at more than one level at a time, which is the way talented designers tend to understand their work. The machine could be "taught" to understand these relationships by reading the "patterns" generated by three classes of entities—zoned objects (formal elements), spaces (use types), and values (actual dimensions between zoned objects). Each of these classes of entities could be deliniated into types, so that they could understood as patterns or manipulated as groups or individuals.

The application of this method would be a useful "design primitive" computer-aided design system that posesses some intelligence about how we understand the built world.
Bibliography

Alexander, Christopher - "The Timeless Way of Building"
1979, Oxford University Press, New York

Alexander, Christopher - "A Pattern Language"
1977, Oxford University Press, New York

John Wiley & Son, New York

Handel, Steven - "A computer-aided participatory design system", 1975 M.Arch thesis M.I.T.

Laseau, Paul - "Graphic Thinking for Architects and Designers", 1980
Van Nostrand Reinhold, New York

Mitchell, William C. - "Computer - Aided Architectural Design"
1977, Van Nostrand Reinhold, New York

Various Authors - Daidalos 5 (Berlin Architectural Journal), "The First Sketch"
Sept. 15, 1982
Bertelsman Facchzeitschriften GmbH, Berlin
Footnotes

1) Laseau - "Graphic Thinking for Architects and Designers"
2) Ibid.
3) Mitchell - "Computer Aided Architectural Design"
4) Ibid.
5) Ibid.