ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

IN THIS ISSUE:

AMG COLOR RESEARCHERS VISIT FORT MONMOUTH by Robert Solomon

On June 23, Victor Tom and I visited with Dr. Elliot Schlam at the Ft. Monmouth Research Laboratory near Bradley Beach, New Jersey. The purpose of the visit was to see how the basic color research which will be conducted under the ARO contract could be applied directly to related problem areas at ARTADS and ECOM. First, I gave a one and a half hour talk to several research staff members on basic colorimetry, color coding, and the psychophysics of color vision. Included in the audience was Irv Reingold, Chief of the Beam, Plasma, and Display Technical area; Sid Littman (who works for Sil Pilosi at ARTADS); Joe Pucilowsky (who works both at ECOM and at ARTADS); and Munsey E. Crost (who has done considerable reading in the area of color perception).

Following the talk, Pucilowski and Littman presented three areas of current work where the hardware systems have already been nearly determined. One project involves the use of LED or plasma displays in conjunction with standard five color land maps to display dynamically target information and other strategic data. A Litton Data Systems module consisting of 32 x 64 picture elements (pels) is used as the building block in a system which projects information through the map from the rear. The pels consist of red and green LED's which can produce a red, yellow (each LED on at half power), or green dot. The 30 mil dots are on 45 mil centers and a meter square display is planned using 792,000 pels (390 modules).

Numerous human engineering, perception, and display problems were discussed in terms of evaluating the contrast, legibility, etc. of the various backlit lights through the different map inks. For example, the visibility of the green LED through green ink appears

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

AMG COLOR RESEARCHERS VISIT FORT MONMOUTH continued

to be rather poor. The LED's radiate at 1400 ft-lamberts and ambient lighting in the range of 10-25 ft-lamberts is required to view the map and yet not dilute the effect of the LED's. The system delivery time is 18 months and so ARTADS is mainly interested in setting up evaluation and testing experiments. Both Victor and I felt that the map overlay display system could be simulated to some extent on the 85.

A second method of dynamic map identification uses an orange plasma panel manufactured by Control Data Corporation (CDC). In this system, the map would go behind the display and the electrodes would appear as a stationary high spatial frequency mottle. The parallax problem is virtually eliminated by using a very thin rear plate. I informed those present, that any color analysis of such a system requires full spectral data on ambient lighting, phosphors, inks, and emission spectra. Pucilowski assured me that he would send out the relevant data quickly.

The other two projects involved the problems of obtaining fast hard transparent copy of the plasma overlay or LED rear projected symbols, and the standardization of the necessary graphics symbols, their encoding, storage, and transmission. In addition, Schlam is quite interested in interactive graphics and touch sensitive displays and I indicated that we will send him out the necessary background material on the work by Dick Bolt as a feeler for obtaining additional funding in this area.

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY

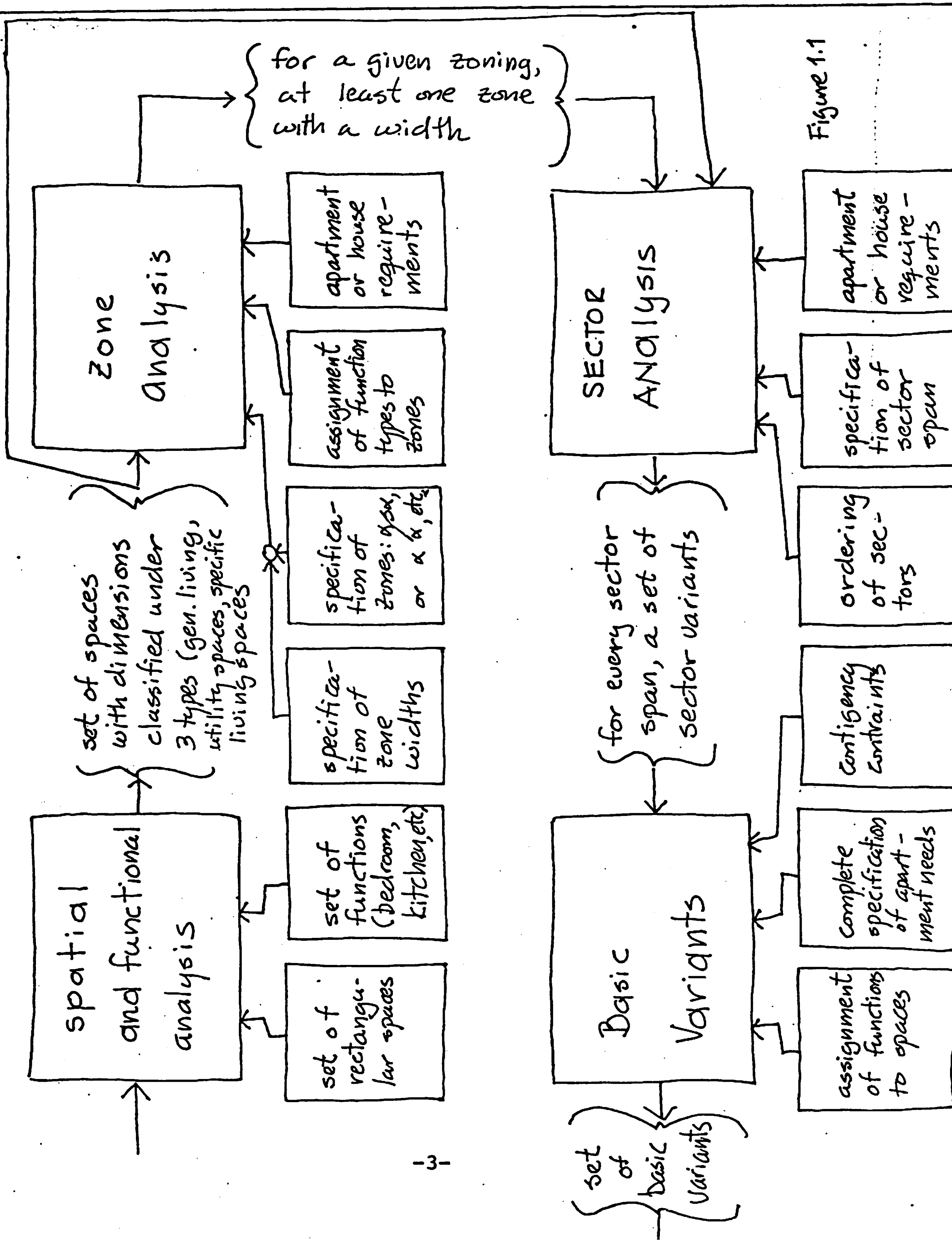
by Mike Gerzso

As outlined in the SAR methodology, the generation of basic variants is usually the stage described last. But in some situations the basic variants stage is the first area to be considered. This is certainly the case in writing a program that performs the analysis of the SAR methodology. To have some understanding of why this is so, we shall first consider the overall organization of the new SAR program. This could be interpreted as a conceptual model of the SAR methodology.

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976



ARCHITECTURE MACHINATIONS

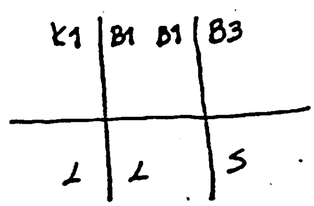
A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

(See Figure 1.1 on previous page.)

In the first stage, the dimensions and the possible functions of spaces are decided. In the second stage, the relation of these spaces to zones are analyzed. One of the purposes of this stage is to establish the minimum and maximum width dimensions of the zone. In the stage called sector analysis, the dimensions of the sector are laid down. The last stage to be carried out is that of the determination of the set of spatial/functional configurations called basic variants. These configurations should not be interpreted as floor plans. In the SAR methodology, a floor plan is defined as a subvariant which is a particular instance of a basic variant. Therefore, a basic variant is an "outline" of how spaces may fit together. In all the stages described above, the determination of dimensions and positions of spaces in zone sectors is subject to constraints such as the number of rooms desired, their relationship among each other and their relationship with the zones and sectors. (I recognize that this is a very schematic description of the method and one should see the publications on it.) A schematic floor plan (See Figure 1.2.) is used to represent one basic variant.



K1 = kitchen
B1 = single bedroom
B3 = master bedroom
L = living space
S = study

Figure 1.2

Several reasons exist for beginning with the basic variant generator: a) it is the "target" system, so to speak, to which all other systems provide input data and preliminary computation. That is, the basic variant generator requires its input data to be of a particular type and this type can only be made if we design the whole structure with this constraint in mind. b) the most complicated computations are carried out by the basic variant generator which means that the zone and sector stages could not be implemented if it were clear that the basic variants generator could not be implemented.

2.0 THE SEARCH SYSTEM: The basic variant generator is a depth-first

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

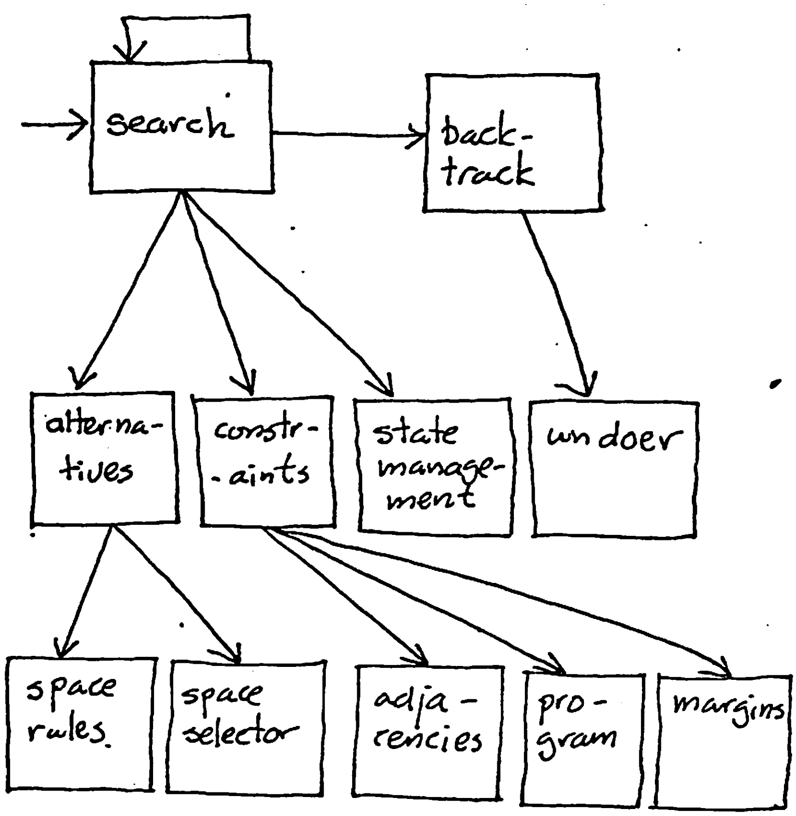


Figure 2.1

backtracking search program. Its general schematic organization is as follows (See Figure 2.1.).

At the top of the diagram are the programs that control the process of searching for all the basic variants. They consist of the search program itself and the backtracking program. The first one takes care of the generation of alternatives -- verifying them with respect to the constraints and updating the data base of positional relationships. The second program "undoes" what the first program generates in the case that the search program cannot find any alternatives. It is possible to visualize the performance of the search program with the aid of Figure 2.2. We see that the path of the search program always attempts to reach all the way to the bottom of the tree, There, it will backtrack and proceed to search down another path. The only difference with a bottom node and other nodes is that before it backtracks, the search program stores the solution -- a basic variant -- that it has found. The search program performs this operation recursively until it has gone through the entire tree.

The process in which the search for all basic variants is carried out

ARCHITECTURE MACHINATIONS
A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

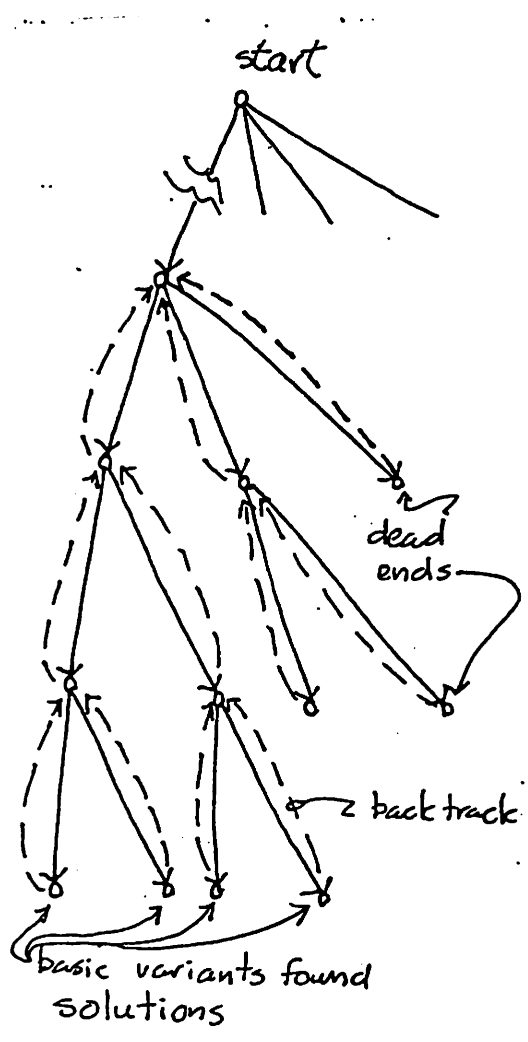


Figure 2.2

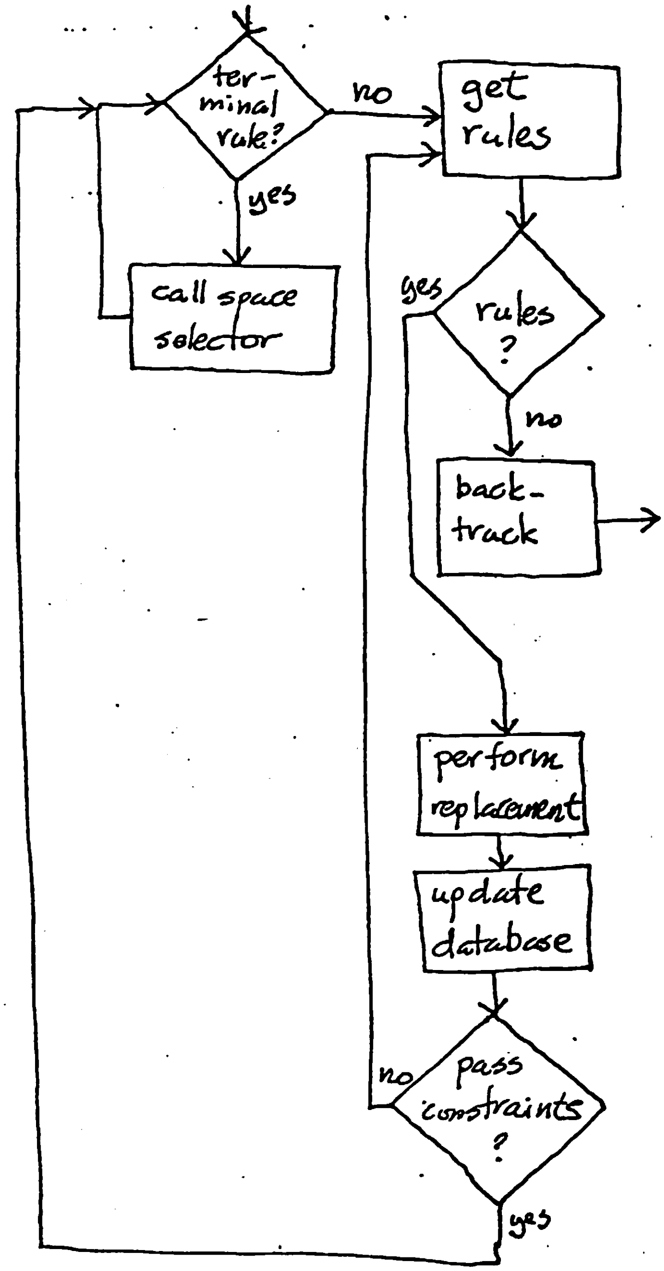


Figure 2.3

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

following the algorithm in Figure 2.3. First, the search program determines if there is an alternative available. If there is, then it updates the data base and checks for the constraints. If the alternative passes the constraints, then the search program proceeds to inquire what the alternatives are after that. If there are no more alternatives, then the search program calls the backtracker.

3.0 GENERATION OF ALTERNATIVES. The search program would not "know" what alternatives to consider if it were not for some mechanism that generated alternatives. In this system, there are two types of alternatives. If we consider an arbitrary portion of the search tree, the search program would "consider" either the first type of alternative or branch which is replacement rule and the second type which is a selector of spaces. (See Figure 3.0.)

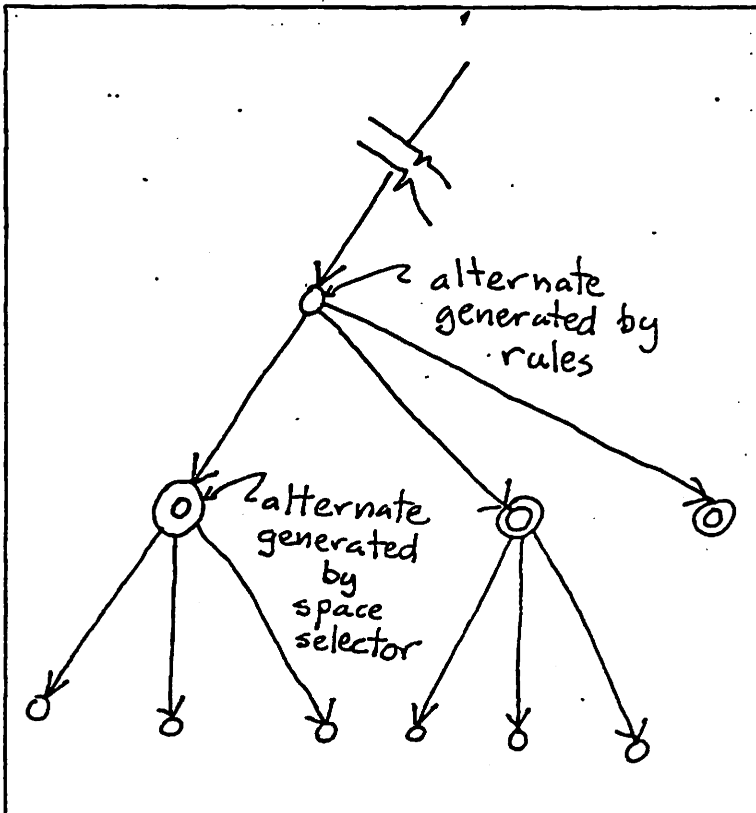


Figure 3.0

The entire generation of basic variants from the first stage of zone analysis through sector analysis and the basic variants stage itself is controlled by a set of rules. These rules guide the search program. The way in which the basic variant generator is programmed allows the freedom of redefining the rules since they are data. Therefore, different rules can be input for different projects. This is very desireable if the basic variant generator were to be used in research in which new ideas concerning zoning and sectoring can be proposed and evaluated. This means that any change in which

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

basic variants are to be generated does not imply that the whole system has to be reprogrammed.

It would take much more space to describe completely the characteristics of these rules than we have room for here. However, it is hoped that the following example is sufficiently explicit to provide some idea of the motivation behind the rules.

Suppose that at some point in the search tree a support has a zoning consisting of an alpha zone (alpha 1), a beta zone, and another alpha zone (alpha 2). (See Figure 3.1.) In each sector, there is a symbol

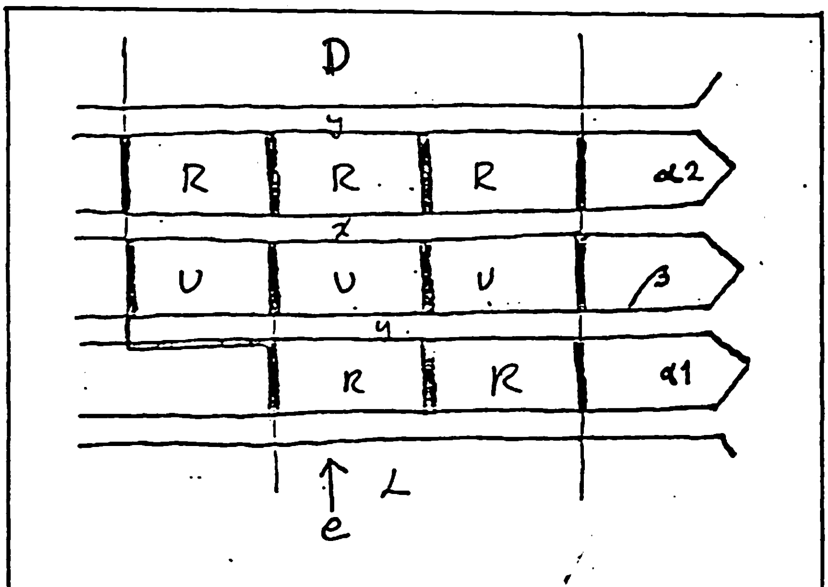
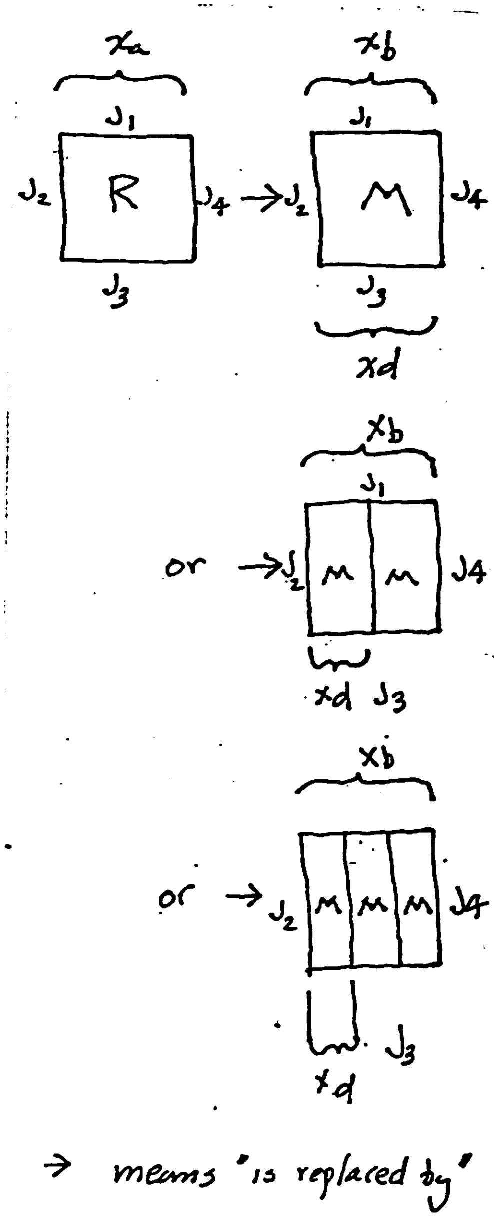


Figure 3.1

which represents the type of space. For example, in alpha 1, all the sectors are



(2)

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

of type "R". The left "R" type sector is special because it will contain the entrance to the apartment. At this point, we also presuppose that the space selector is pointing to the left-most "R" type sector in alpha 1.

The search program gets out of the list of rules that refer to "R" type spaces. It turns out that these rules are so-called sector division rules defined in the sector analysis stage of the program. The formula for this is (2) (See previous page.) and it "says" that the "R" type spaces can be divided into one or two or three "M" type spaces. The sector is now comprised of two "M" type spaces which is the alternative that the search program has selected. (See Figure 3.2.) The two other alternatives are not yet tried but will be when the search program backtracks to this point. It should be made clear that the operation of replacing an "R" type space with two "M" type spaces consists of one node in the search tree.

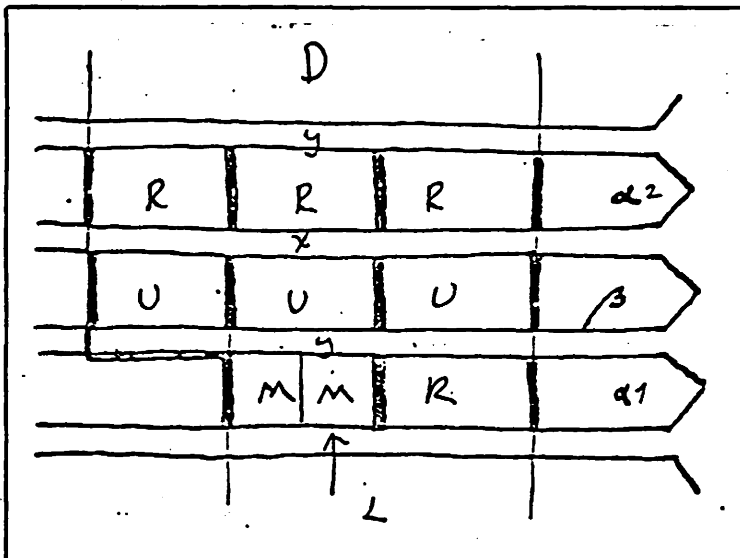


Figure 3.2

Now that a sector has been divided (and we know that this is a legitimate sector because the rules will only give us elements of the set of sector variants and nothing else), the space selector must decide which of the two spaces will be processed.

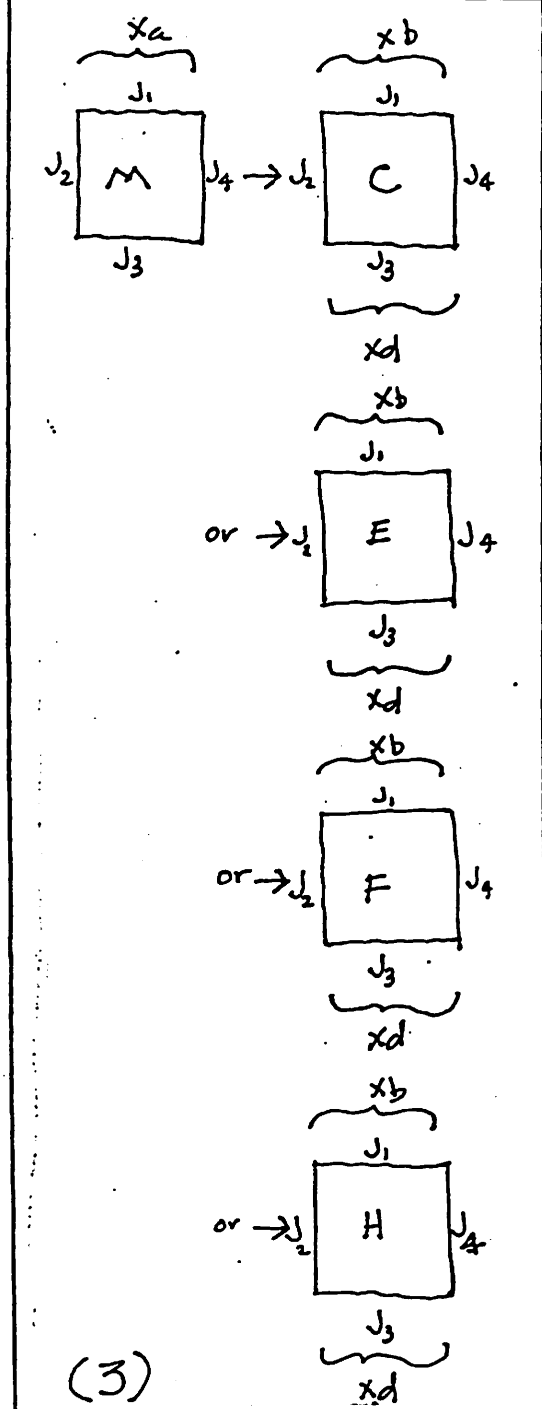
Suppose that the right "M" type space will be considered. The search program goes and looks for rules that concern "M" type spaces with the condition that the new space must be an entrance hall. The rule for this is formula (3) (See next page.). Here, the "M" type space can become a "C" type space in which only cooking rooms can be assigned,

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued



(3)

an "E" type space in which only eating spaces can be assigned, etc. None of these replacements are valid except for the "H" type space. (See Figure 3.3.)

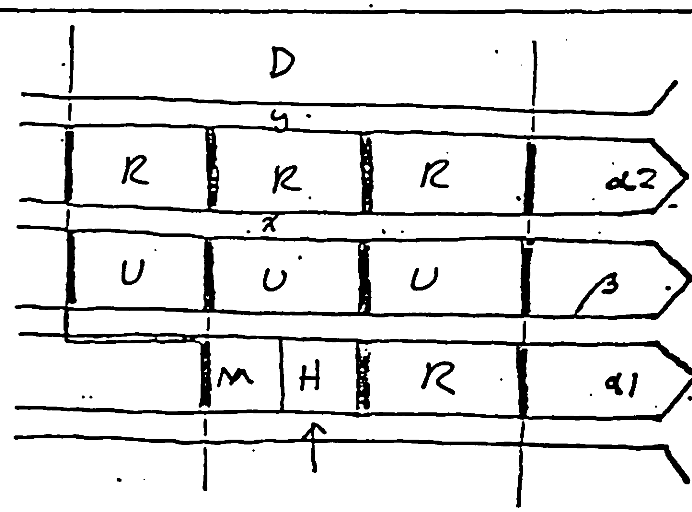
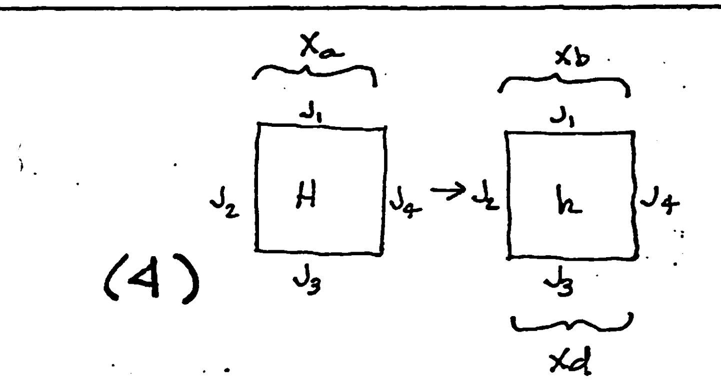


Figure 3.3

The search program queries the rules data base for a rule referring to "H" type spaces and finds one, (4) which



(4)

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

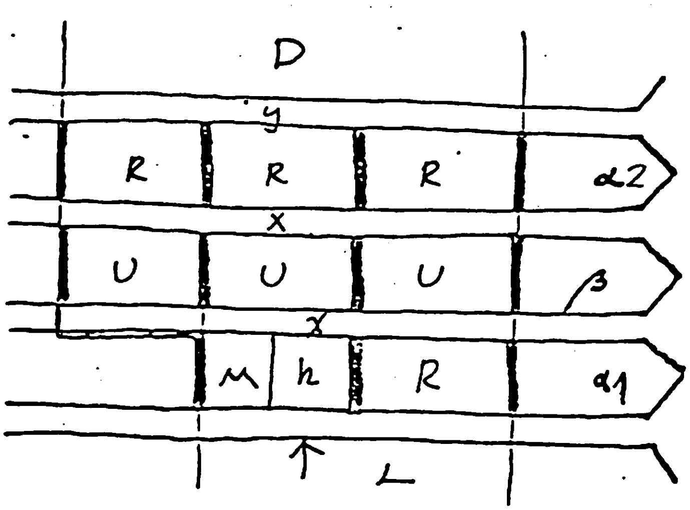


Figure 3.4

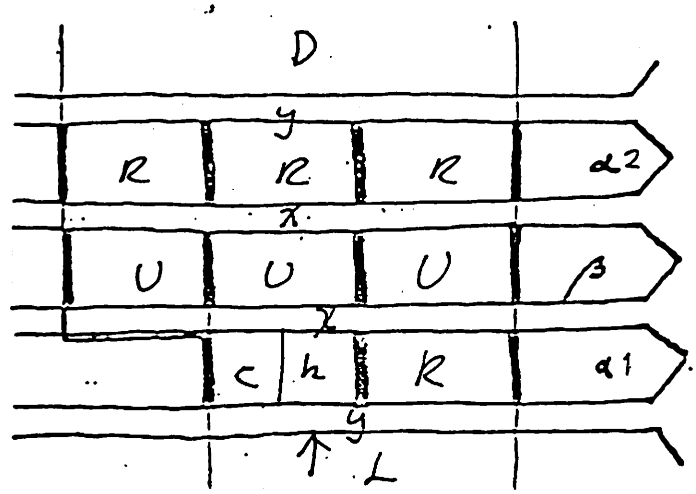


Figure 3.5

indicates that the "H" type space is now to become an "h" terminal type space. (See Figure 3.4.) In reaching a terminal rule, the search program triggers the space selector which will now point to the other "M" type space.

This is the second time the search program retrieves replacements for "M" type spaces. However, now it has to be more selective in the alternatives presented to it by the rules. It can only accept types of spaces that are not halls. In addition, because of other constraints, it must select either a "F", "C", or "E" type space. (These constraints are given by the user of the basic variant generator.) Let's say that the search program selects "C" type space (See Figure 3.5.). As in the previous cases, the search program will replace the "C" type space with other spaces until it has found a terminal space. (See Figure 3.6.) This particular terminal space is a kitchen that has a stove, a sink, and a set of cabinets. (See Figure 3.7.) The basic variant generator has no way to verify the internal spatial

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

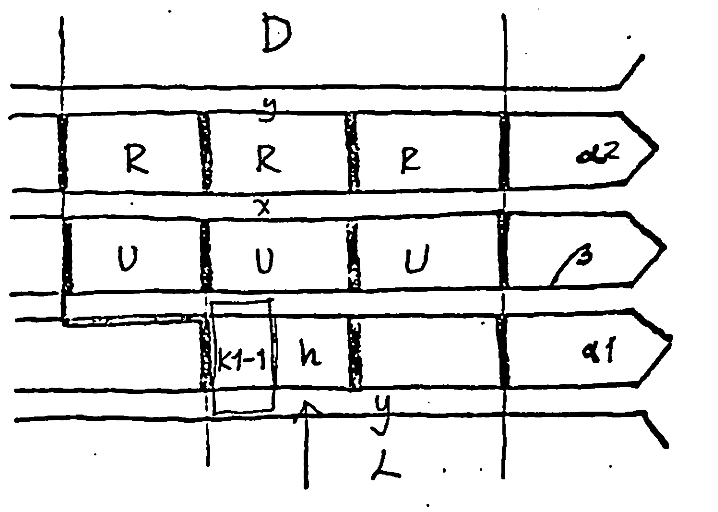


Figure 3.6

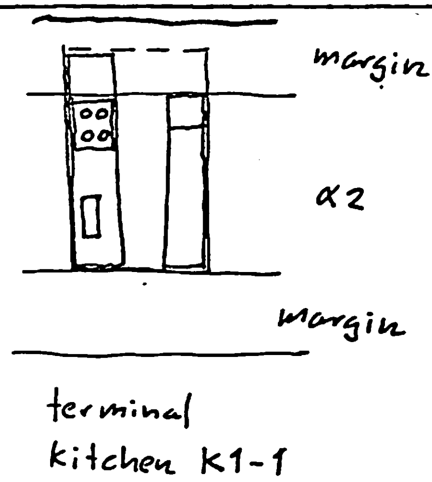


Figure 3.7

organization of this kitchen. However, the only aspect it must check is that if a space on the other side of the margin will interfere with the space taken by the kitchen.

4.0 THE COMPUTATION OF ADJACENCIES. The computation of adjacencies is done on an orthogonal basis. The representation of the "nearness" or "next-to-ness" of spaces has been chosen to be fuzzy sets. Given two spaces, A and B (See Figure 4.1.), the nearest parallel sides are computed. Such a computation of adjacency of these two spaces is carried out with respect to a side of space A and the closest parallel side of B. The "nearness" is described in terms of the following components: a) the "y" distance of the side of space "A" to the side of space "B", b) the "x" distance between these two spaces, c) the overlap between them, and d) the remainder.

Since the notion of what constitutes "nearness" on either one of the "x" or "y" axes can change from person to person and from project to project, each component can be defined according to the situation. The fuzzy set equations for each component are (5). Each one of these equations can be interpreted as a subjective evaluation of the

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

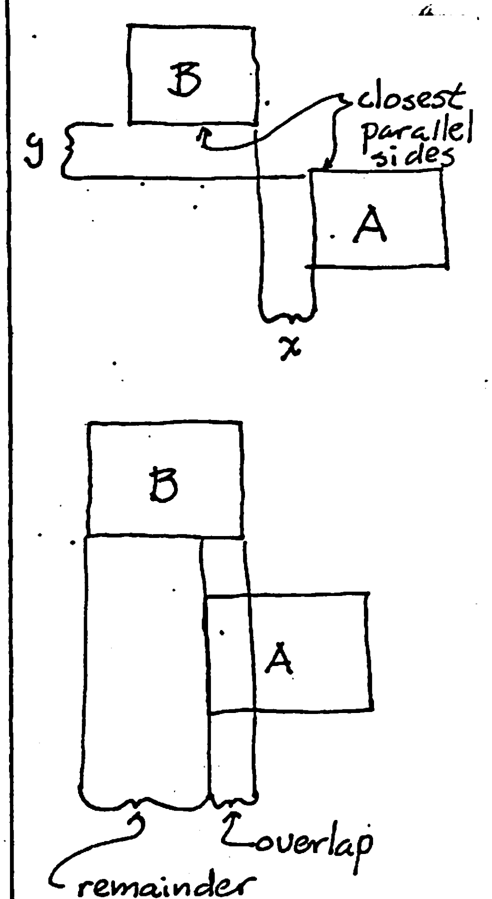


Figure 4.1

$$\underline{\text{near}_x} \triangleq \int_X \mu(x)/x$$

$$\underline{\text{near}_y} \triangleq \int_Y \mu(y)/y$$

$$\underline{\text{overlap}} \triangleq \int_L \mu(l)/l$$

$$\underline{\text{remainder}} \triangleq \int_K \mu(k)/k \qquad (5)$$

distance with respect to what one considers "near", "overlapping", and "remaining". It is interesting to note that the notion of establishing norms of the participating decision makers during the design process is particularly suited for using fuzzy sets. For example, in the case of "near" on the "x" axis, 0.0 centimeters can be considered adjacent to or very near and thus receive a ranking

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

THE COMPUTATION OF BASIC VARIANTS AS A TOOL FOR THE SAR METHODOLOGY continued

of $\mu(0.0) = 1.0$ while 6.00 meters may be considered not near and therefore be ranked as $\mu(6.0) = 0.01$. These evaluations are called membership values and they represent the degree of membership of the distance on the "x" axis to the set called "near on the x axis".

5.0 TYPES OF CONSTRAINTS. The prupose in computing the new updated positional relationships of each space with respect to another is to verify that the new space is within previously established constraints. These constraints are of the following categories: a) the number of spaces as required by the functional program of an apartment (1 kitchen, 2 single bedrooms, and the like), b) the constraints placed on the interrelationships of these spaces such as that the kitchen cannot be next to the master bedroom and the kitchen again must be next to the entrance of the house or apartment, c) constraints with respect to the zones and sectors which dictate that certain spaces must be in a certain physical place; this type of constraint is expressed in terms of stating that the bathroom, for example, must be in a particular sector which in turn is in a particular alpha-beta zone, d) the last type of constraint is what can be termed contingency constraints. These have the following form: in x sector in y zone there cannot be a space of k function if an r sector in an s zone has a space of q type function.

6.0 CONCLUSION. The implementation of the basic variants generator is the first step in the development of the new SAR system. It is also the most difficult and critical. At this time, most of the code has been written and is on the Architecture Machine in separate modules. It is possible that in the next few weeks we will have it running as an integrated system.

A JOURNEY TO PRIME by Seth Steinberg

This Friday, Mike Kazar and I took the turnpike out past route 128 to a gray slab concrete building which houses the Prime Computer Corporation. Prime manufactures the Prime 400 which can be characterized as the poor man's Multics. It all started over a week ago when I

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

A JOURNEY TO PRIME continued

received a call from Gary Morgenthaler from Tymshare. He was in Boston and interested in higher level programming languages for small computers. Tymshare has bought about 50 7/32's and may buy another 200 or so. They use the machines as network nodes on the Tymnet.

Anyway, I was introduced to a faction from Prime at a Hungarian restaurant in Brookline. And so, I was invited out to Prime to take a look around and have a few questions answered and a few asked. The original Prime 100 was a derivative of the Honeywell 316, a 16 bit minicomputer. Prime has a fair number of ex-Honeywell employees, and a number of ex-Multicians, which helps explain the Prime 400.

The Primes 100 and 200 were basically simple machines. The first jump occurred with the Prime 300 which was/is a paged machine, but still limits the user to a 64K word address space. About 2 years ago the Prime 400 was first conceived and has been gestating until recently. According to Russell Barbour, their PL/1 compiler team, the machine was designed to support a language like PL/1 in a reentrant, time-shared, paged, and segmented environment. Dynamic linking was designed into the hardware early.

Their current software takes little or no advantage of their astounding hardware and firmware support. Each user gets one big contiguous core chunk. Even the editor is not shared. This still doesn't stop them from running a 30 user time-sharing system. Their November software release will actually start to take advantage of the segmentation.

So what is so amazing about the Prime 400? On the surface it looks like a hopelessly kludgey sixteen bit machine with a thirty-two bit processor superimposed on it -- something like an underwater F104. This is just a surface impression. Most of the design was thought out, and most of the apparently strange features will have uses (or so I was informed). Once one is past the "upward compatible" problems of getting the machine capable of using 32 bit addresses, the interesting things start to happen.

The 32 bit address contains two bits of ring number. This leaves 30 bits of addressing data. Since few machines can hold a gigaword of memory on their main bus, a good chunk of this memory will be

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

A JOURNEY TO PRIME continued

virtual. Twelve of the 30 bits of address are segment number. At any instant the machine can know about four groups of segments. That is, the addressing scheme is based on 4 DTAR's, which are registers comparable to the DBR on the Honeywell 6180. To an operating system designer, one DBR is worth between 10 and 400,000 double precision floating point registers (with the hardware complex trig function option).

For those who have no idea of what I am babbling about and/or missed Dave Nadler's article on memory schemes in a previous A.M. article, I will explain. A Prime 400 address can be thought of as a Gaussian integer, or a complex integer, if preferred.

Basically, it has two parts. A segment number and a word number which must be turned into physical core addresses by some means. This is where the DTAR's come in. Effectively, they point at a table of 4096 segment descriptor pointers. In reality, each points at 1024 elements (or less), so the table can actually be sparse. The segment number is used to look up the segment descriptor in the table pointed to by the appropriate DTAR. The segment descriptor contains the page table (up to 64 entries) for that segment. The second part of the address, the offset, is used to find the page number which can be used with the page table to find the actual core address.

This all counds like an awful lot of hardware, but it fits on about 1.6 boards. This is because of firmware, or clever microcode which actually does the work. What really happens when an address is presented to the mapping unit is much simpler. First, the address is mashed around a bit and some relatively small magic number is obtained (much like a hash code) and used to look up some data in a small high speed memory made of RAM's. If the number in the RAM's matches the segment and page number in the address, then the remainder of the number of the RAM's and the page offset are passed on to a 1K word RAM which forms the cache of the Prime 400. This physical address is hashed again and something read out of the cache. If it matches the address being sought the data is present. Otherwise, it is a job for microcode.

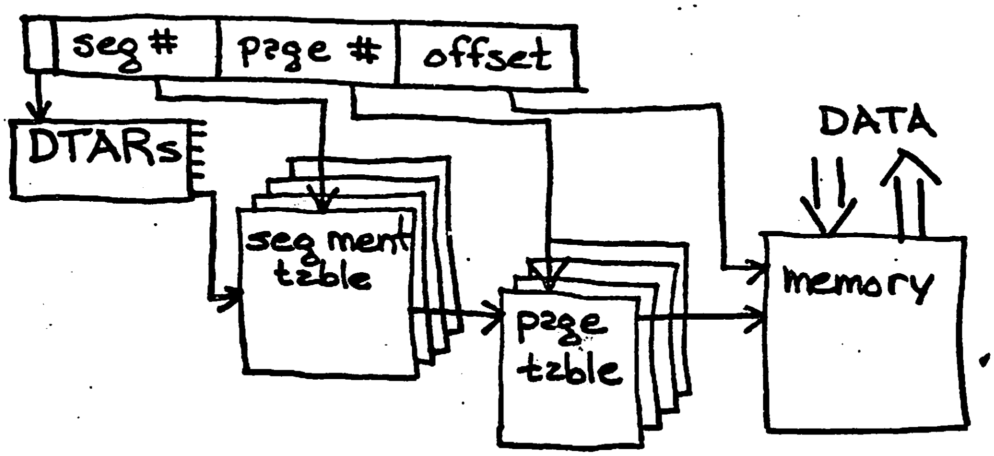
This sounds complicated, but is probably a lot simpler to build than the hardware needed to implement this scheme directly. Of

ARCHITECTURE MACHINATIONS
A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

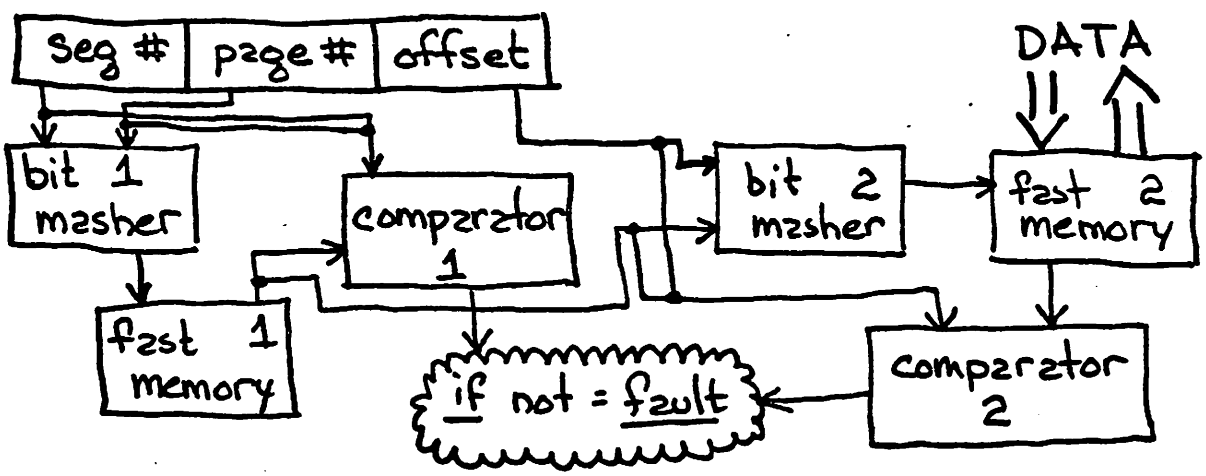
Vol. II., No. 26. June 27, 1976

A JOURNEY TO PRIME continued

What appears to happen



What happens



ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

A JOURNEY TO PRIME continued

course the most impressive thing about this scheme is the fact that the Prime 400 is not a whole lot more expensive than the Prime 300, and is an awful lot cheaper (30:1) than the Honeywell 6180, the only other machine with this kind of hardware.

Prime is working on a PL/1; although they have only been writing for a short time (in Fortran) they seem to be making some progress and would like to start writing in PL/1 as soon as they can. They are avoiding IBM's mistake and are keeping the number of people cooking th broth to a minimum (in the case one). I may as well note that their PL/1 writer also owns an orange Rabbit. Prime does have someone on the ANSI committee, and the committee is considering a PL/1 subset design. Freiburghaus has supposedly already submitted a briefly scrawled page of notes which proposes the DG subset as the standard. It leaves out all aggregate operations (as one might have expected), but his version also removed automatic and based initial. A large number of compromises will have to be made. Personally I would like them to do what they did to COBOL: break it into 4 levels, so a typical designer could either grow to the next or take things out to meet the next lower level. This will probably not be done.

In general, the visit was rather interesting. Mike Greata, a hardware type, and Russ Barbour, a software type answered most of my questions, although they were unable to release any documentation on the Prime 400 due to objections from their legal department.

An interesting side note. Among the in-house equipment there are a couple of Summagraphics tablets which are used for PC board layout design by the hardware team.

ODDS AND ENDS

The TSD is up and running! It is currently interfaced to the model 70, and is on the face of Imlac 3C. When all final tuning is done, it will be moved to the second screen of 3A, and the interface will reside in the new 7/32. See Steve Mann for software notes.

ARCHITECTURE MACHINATIONS

A weekly newsletter of the Architecture Machine Group, Department of Architecture, M.I.T., Room 9-518, Lee Nason, editor.

Vol. II., No. 26. June 27, 1976

ODDS AND ENDS continued

We have commenced building a Button interface so that you may interrupt the process you are running without using the front panel. This system will consist of a pair of buttons located at each MAGIC terminal, and will signal an interrupt to the file processor. We hope to have this whole thing done by the time MAGIC 5 is up. Specific notes will follow.

The 85 video system, and maybe the machine will be brought down this week (Wednesday onward) to correct some flaws in the display and the memory system. Meanwhile, there is the command OFF, in TOOLS. on the fixed heads. It will turn video off, and guarantee the integrity of your processor.

The processor communicator will be down from Wednesday until Friday for wiring modes (button box).

Video and maybe the model 85 will be down starting when Paul Pangaro goes (on or about July 6) for one week to rework the video and memory hardware. Most of the time, the model 85 will be available for users, but at least sometimes it will not be. If the scheduling gets to be a problem, I will move the moving head disks to either the 70 or the old 7/32.

DOCUMENTATION by Lee Nason

For those of you who don't already know, the new PLOD manual has been written and is in my possession. If you want a rough draft copy, leave a note for me (final version will not be out for quite a while).

QUOTE OF THE WEEK

"How long does it take you to swim an hour? 45 minutes?" -- O. T. Anderson -- in context.