Rectangular Meshes: Their Uses and Control in Computer-Produced Architectural Schemes

Gonzalo Velez, Architect, Caracas, Venezuela

A procedure is described to obtain computer oriented solutions to functional organization problems in architectural design. This procedure considers both a satisfactory functional scheme and one that satisfies the dimensional constraints.

KEY WORDS AND PHRASES: computer-aided design, space planning, architecture
CR CATEGORIES: 3.23,3.26,3.41

FOREWORD

This work is concerned with the effects rather than the causes of the actions brought about in tampering with the hidden mechanisms which control the formation and adaptation of geometrical patterns.

For that reason, it shall not go into topological considerations about which, on the other hand, there has been a number of excellent works published lately (1,2,3).

SCOPE OF THE WORK

The following work is a compendium of a number of ideas, both personal and derived through three years of on and off research, along with colleague Lindolfo Grimaldi.

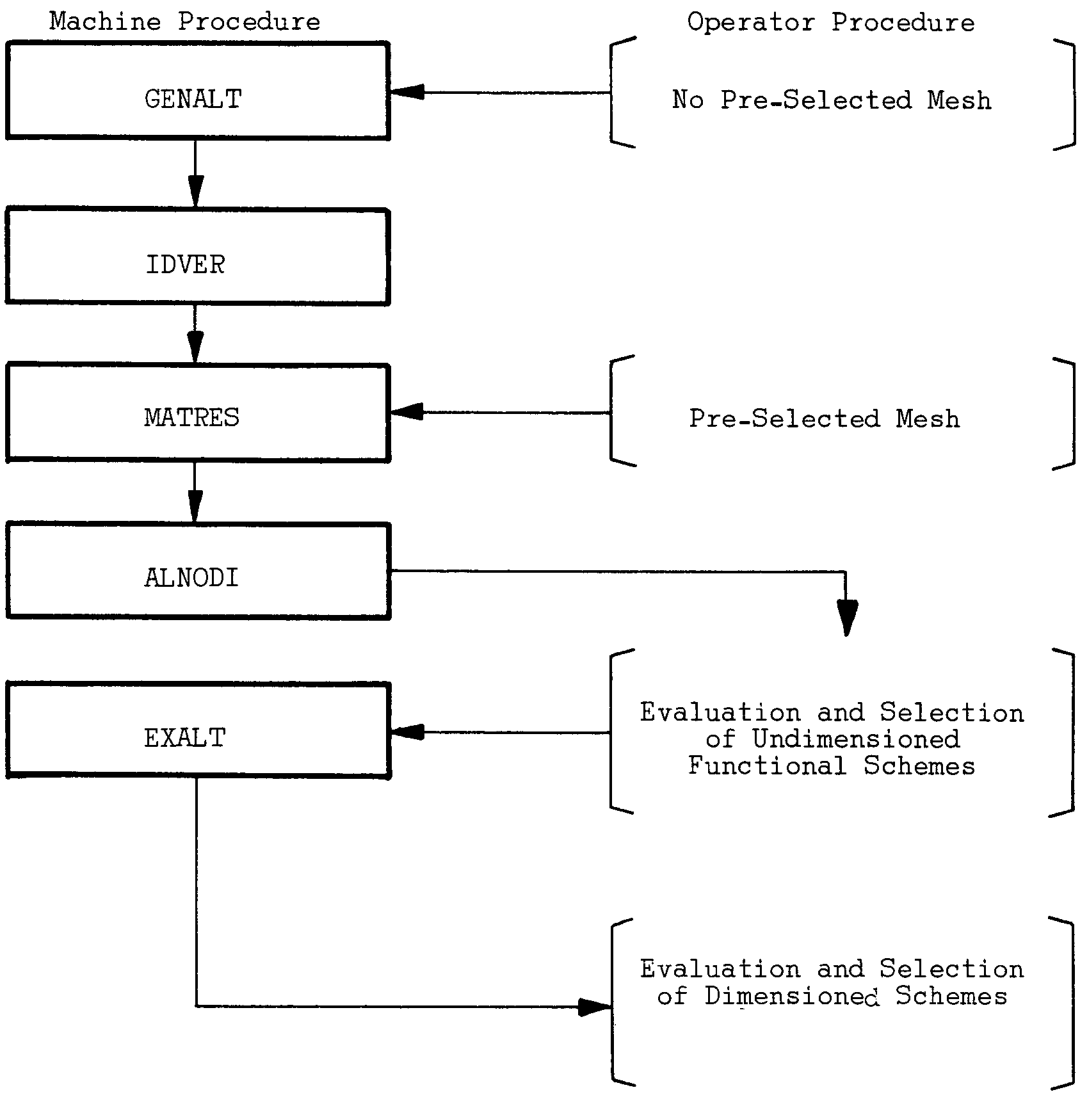
It constitutes a part of a more extensive experimental work being carried out both at the Faculty of Architecture and Urban Planning, Universidad Central de Venezuela (4), and at government's low cost housing organization Banco Obrero (5).

The aim of this paper is implicit in its title: To provide ways for the generation and control of meshes as an aid to the designer.

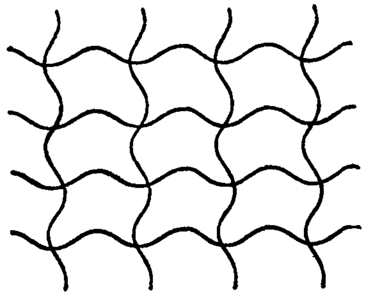
The procedure described in the following pages contemplates the generation and evaluation of meshes from a functional viewpoint, through the aid of five routines:

1. Routine GENALT - Generation of meshes.
2. Routine IDVER - Identification of meshes.
3. Routine MATRES - Analysis of feasibility of functional lodgement.
4. Routine ALNODI - Generation of undimensioned functional alternatives.
5. Routine EXALT - Attaining of satisfactory dimensional requirements for the schemes obtained from ALNODI, whenever possible.

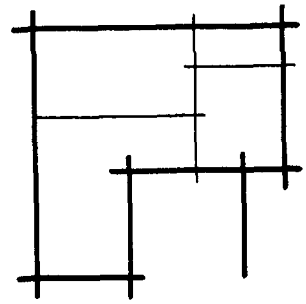
All five subroutines have been written in basic Fortran IV language. The first four are already operating although still in experimental fashion. Routine EXALT is actually in the "debugging" stage.



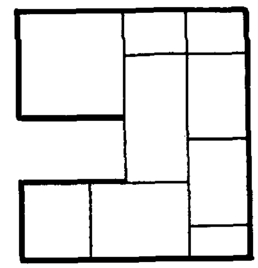
SYSTEM DESCRIPTION



MESH



RECTANGULAR MESH



RECTANGULAR MESH
DESIGN ORIENTED

RECTANGULAR MESHES AND BASIC COMPONENTS

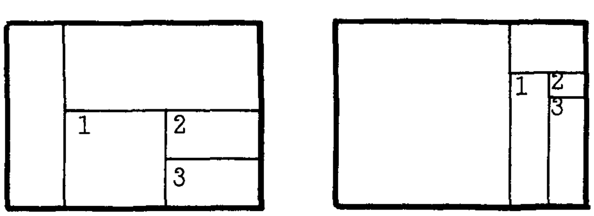
General Concepts

The procedure followed here, to fulfill the aim of obtaining computer oriented solutions to functional organization problems in architectural design, consists in dividing, much after the fashion the architect proceeds traditionally, the process in two cycles.

In the first cycle a satisfactory functional scheme is achieved while in the second cycle the scheme is adjusted to conform to dimensional requirements.

The underlying, unseen, but intuitively dealt with, factor controlling the feasibility of both cycles is the behavior of the chosen geometric patterns which governs the plan, i.e., its capability to adapt to the particular problem for which it has been required. In other words, the application of a rigid pattern to a plan will produce, most probably, poor or rejectable solutions.

It has been sought to express in a better semantic way the meaning of geometric pattern through the use of the word "mesh", because of the tangibility of its physical nature. This mesh must be thought of as capable of deforming in an elastic fashion, without the loss of its original relational characteristics.



IDENTICAL MESHES
(CONCEPT OF ELASTICITY)

Aims

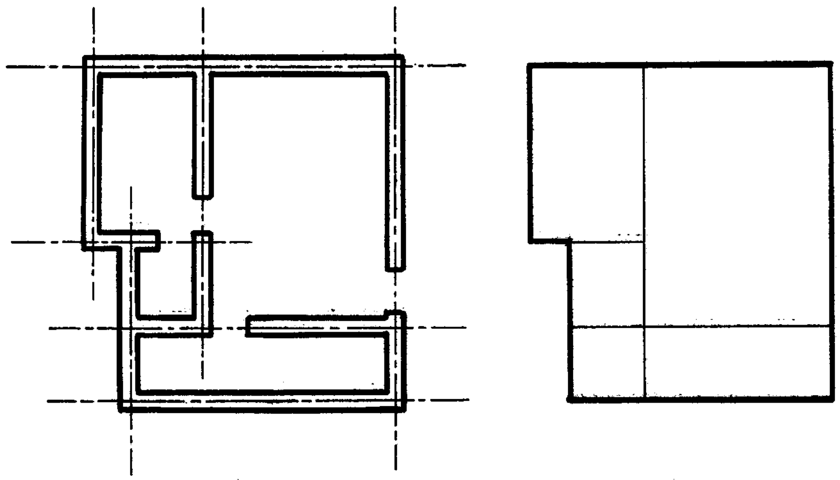
This work aims to show a way to go about solving both cycles of the process although emphasizing the first, in which meshes are created and afterward explored to see if they allow successful lodging of the functions as imposed by the restrictions of the design problem to be solved.

Thus, this work will be concerned mainly with the generation and handling of rectangular meshes, being most commonly used from a constructional viewpoint and the easiest to manage in computer graphic representation. Also, for that reason the accompanying printing procedure can be carried out with the help of a line-printer, it being the simplest and most common output device. No doubt, however, other possibilities could be explored and developed according to the peripheral units at hand.

Definitions

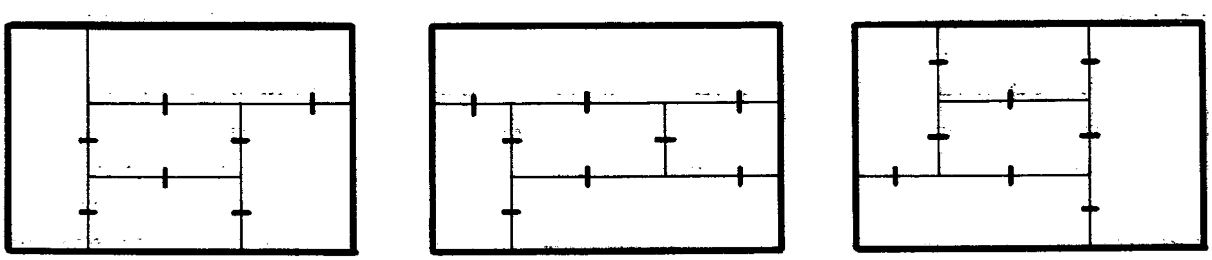
In order to gain a better understanding of the concepts used here, several definitions must be stated. It shall be assumed that:

1. A <u>mesh</u> is the underlying geometrical web that conforms an architectural plan.
2. A <u>component area</u> is a linearly bound portion of the mesh.
3. A <u>partition</u> is any boundary of a component area.
4. A <u>computer matrix</u> is a matrix where each element is associated to a memory position within a computer.
5. A <u>geometric matrix</u> is a computer matrix where a shape is described by its elements, such a shape possessing a size of any given dimension and being totally in-

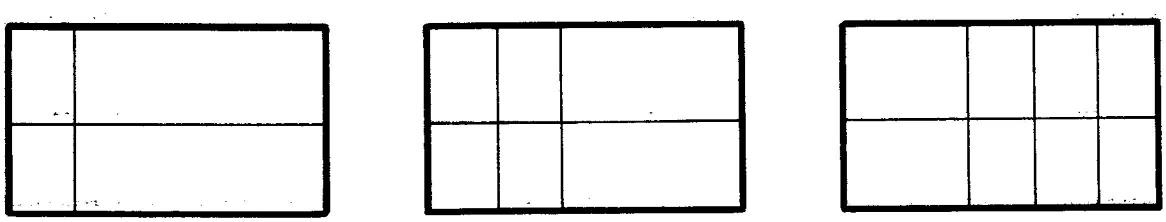


Scheme Mesh

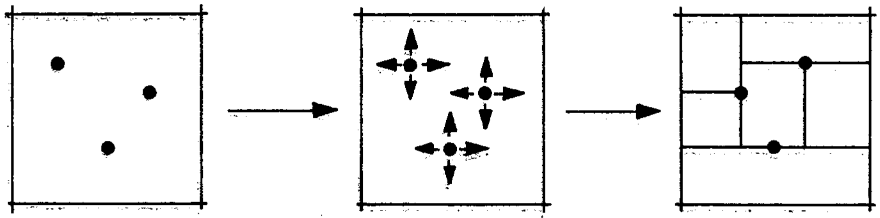
MESH AS RELATED TO SCHEME



MESH "FAMILY" ACCORDING TO INTERNAL RELATIONSHIPS



MESH "FAMILY" ACCORDING TO GENERATIVE PATTERN



STEPS TOWARD THE FORMATION OF A MESH

dependent of the size of the computer matrix.

6. An envelope is the constricting boundary that defines the outward edges of a mesh.
7. A link is the point where two partitions meet.

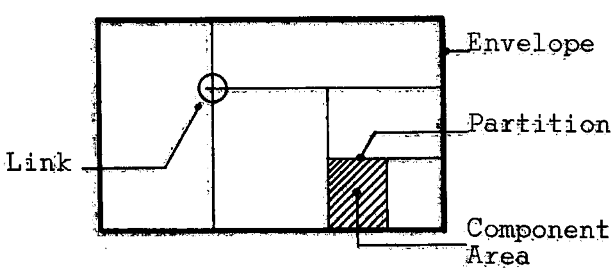
Basic Components

Mesh generation may be thought of as being originated by the partitions drawn from internal points to the edge of the boundaries that surround them.

Accordingly, the complexity of the mesh increases with a) the number of existing points, b) the number of drawn partitions, c) the sequence in which the partitions are drawn, and d) the relative position of the points.

It can be seen at once that the number of different meshes that can be generated through the combinations of the factors listed, increase enormously with the addition of each new point, hence the proposed use of pseudo-random number generators to probe into the distinct "families" of meshes that potentially satisfy the problem under consideration.

We shall consider as basic components: links (points), partitions (lines) and envelope or total boundary of the mesh. Assuming the criterion that each point may be considered a "link" of two or more partitions we may then proceed to analyze the particular variations which arise whenever a rectangular area is generated.



BASIC COMPONENTS OF A MESH

There are actually eleven basic ways of joining partions in the process of generating component rectangular areas. The combinations of these will generate all the possible types of rectangular meshes.

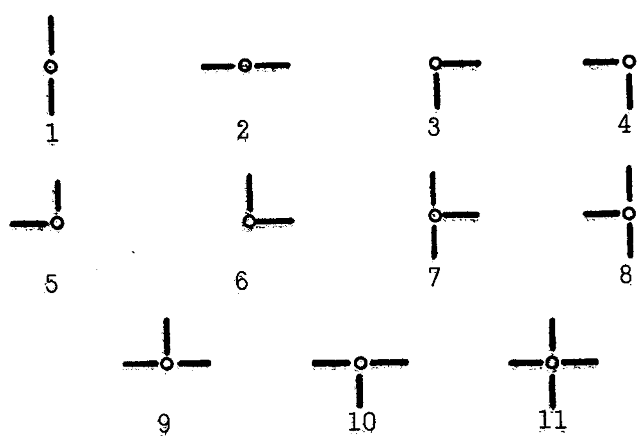
Modules - Basic Types

For the purpose of mesh generation, we shall assume that each basic grouping of partitions is contained within a square which in turn we shall call a "module"; each module is the same size as the rest; of course, there shall be eleven basic modules.

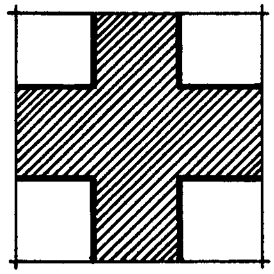
These modules, randomly selected, or otherwise, shall be stored in a geometric matrix where each element corresponds to the number by which the type of module chosen is to be identified. The modules registered in the matrix as a whole, define the mesh.

Module Identification

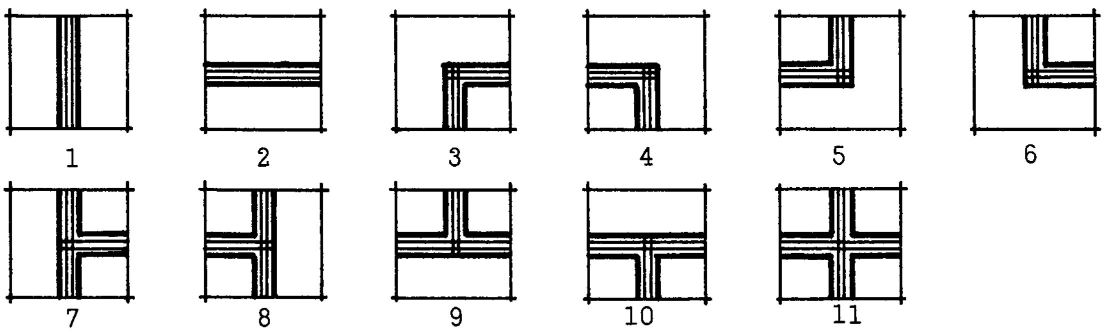
As stated previously, each basic module shall respond to a certain number assigned arbitrarily and by means of which it can be identified at any point of the process.



THE ELEVEN BASIC WAYS TO LINK RECTANGULARLY



Basic Module



THE ELEVEN BASIC MODULES

MESH GENERATION THROUGH MODULE COMBINATION

Process Description

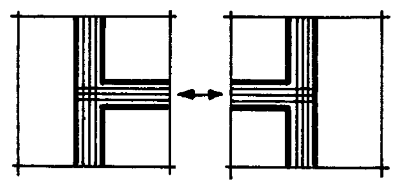
Implementation. In order to generate a mesh, it is essential to provide a pseudo-random number generator which shall feed the matrix containing the modules which will in due time integrate a mesh.

Initial Stage. By means of pseudo-random processes, a matrix shall be filled with numbers identifying modules; they will be, initially, in a totally disordered way, the next step being their rearrangement.

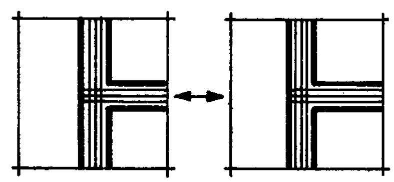
Continuity and Compatibility Criteria. The rearrangement of modules will be based fundamentally on two important criteria:

A. In order to produce architecturally oriented meshes, a principle of continuity shall be observed; that is, we shall consider only these cases where the fabric of the mesh being generated is of continuous nature and discard all discontinuous alternatives.

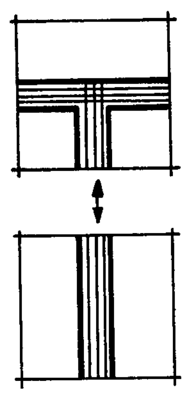
B. To attain continuity, restrictions shall be laid that between two given modules which are adjacent either horizontally or vertically in the matrix, there shall be an agreement in the way their partitions meet; such a continuity agreement shall be named "compatibility".



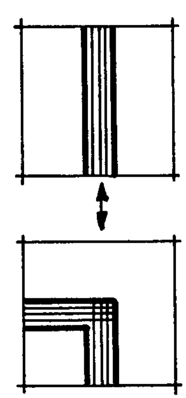
HORIZONTAL CONTINUITY



HORIZONTAL DISCONTINUITY



VERTICAL CONTINUITY

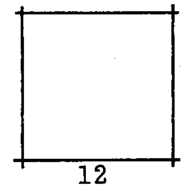


VERTICAL DISCONTINUITY

<u>Rearrangement Stage</u>. The possibly ways in which modules can or cannot be compatible are registered into matrices - one for horizontal and one for vertical restrictions. Thus, the task of rearrangement will consist of the simple procedure of going through the matrix containing the identity of modules and analyzing each module as regards its vertical and horizontal compatibility; if it is found non-compatible in any direction it shall be changed by another basic module and so on, until it reaches a state of total agreement.

Every so often, a situation may arise where no total agreement may be reached. This will be solved with the use of a twelfth module, containing "blank" space; that is, a module where no partitions will be generated. The use of this module shall be restricted to these special situations since its indiscriminate use could bring about a dispersal of area, which, being of no dimensional interest at this stage, would only create limitations to the horizontal extension of the mesh, it being limited by the width of the line printer paper.

THE "BLANK" MODULE

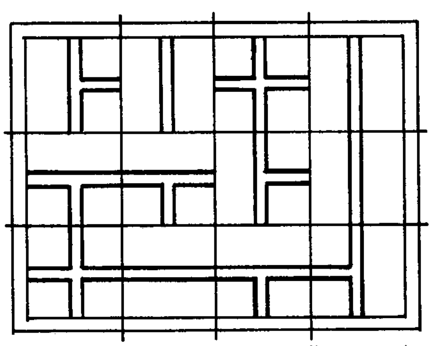


12

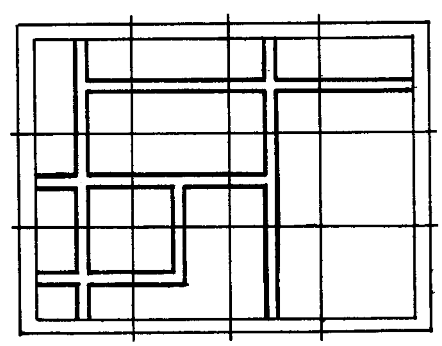
According to this process, a continuous mesh is always attainable, deriving from the original random-produced elements and after undergoing a number of transformations to comply with compatibility restrictions.

Routine GENALT attains the objectives described.

Although in the initial program the different basic modules go into the computer memory by means of an alphanumerical array and are used to print the solutions achieved, it has been found, surprisingly enough, that this is actually not necessary and that <u>it is possible to produce meshes through restrictions alone</u>. This leads to the appealing surmise that it may be possible to generate meshes based directly upon functional restrictions.



Initial Mesh Random Generated



Rearranged Mesh

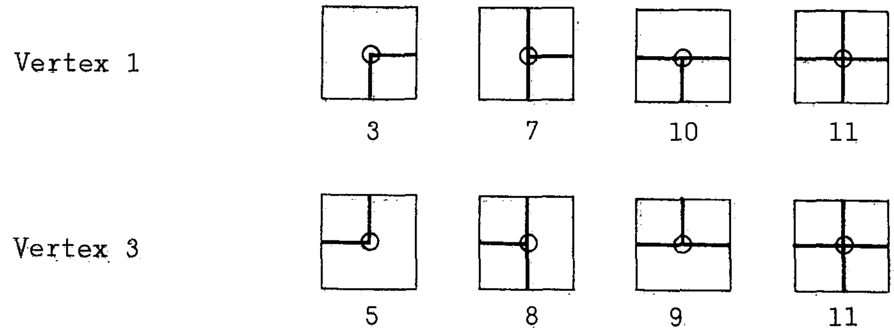
7	1	11	1
10	10	7	1
11	2	10	8

Initial Matrix

7	2	11	2
11	10	8	12
11	5	1	12

Rearranged Matrix

(The elements of each matrix correspond with the types of basic modules.)



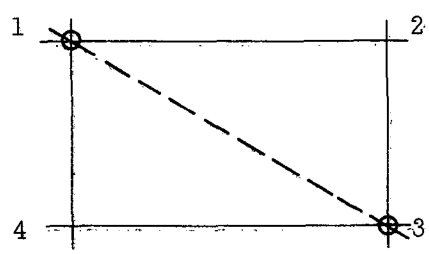
BASIC MODULES CAPABLE OF GENERATING VERTEX 1 AND/OR VERTEX 3

THE IDENTIFICATION AND EVALUATION OF MESHES AS APPLIED TO SPECIFIC ARCHITECTURAL PROBLEMS

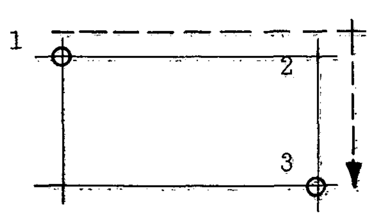
Mesh Identification

Once a mesh has been generated, the following step consists of identifying its coordinates prior to its evaluation, rejection, or admittal from a functional viewpoint.

This will be accomplished by the process of going once more through the matrix that contains the newly arranged modules, and determining the coordinates of the component areas. It will only be of interest for the purpose of the procedure to determine those vertices set at the main diagonal of each component area.



MAIN DIAGONAL



SEARCH FOR VERTEX 1 AND VERTEX 3

Not all modules contain a vertex; this property is limited to only those which lodge an angular union of partitions; in this case, the vertex shall be the apex where the partitions meet. For a good graphic presentation, we shall assume that a module is conformed of 9 x 9 parts, the vertex will always be found at 5,5, if existing. This shall, of course, be referred to a general rectangular coordinate system.

If we consider all four vertices of a component area, the searching strategy will be to identify the first and third vertices and, once this is achieved, to register their coordinates. All this can be attained through a rectangular, file-column search. The basic concept to be considered here is that certain modules are apt to contain the first vertex while others are apt to contain the third vertex.

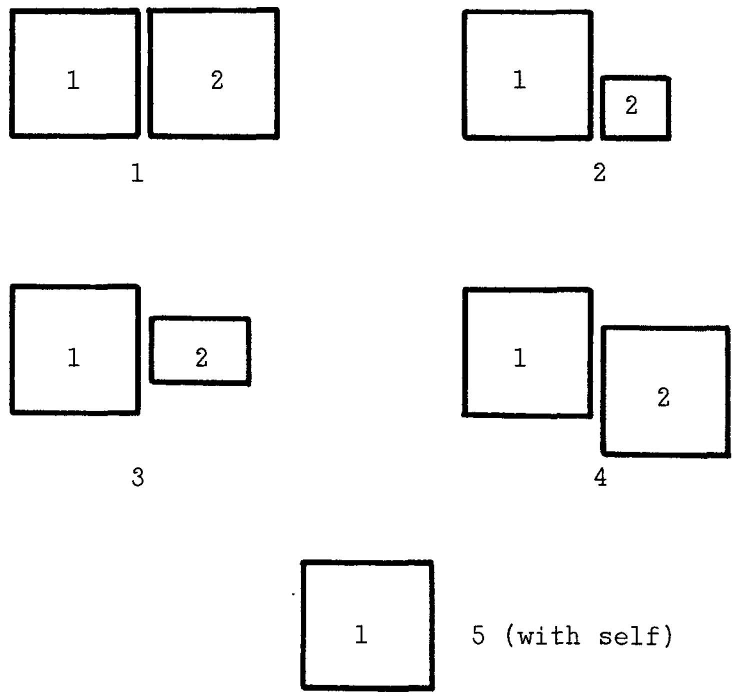
Routine IDVER carries out this task.

Mesh Evaluation

Next, the potential evaluation of the mesh is attained by estimating the number of internal relations (relations to adjacent components) existing for each component area. There are five basic types of existing relationships between components.

The task in this case is to go over the coordinates that identify the mesh and seek out the existing relationships between component areas.

Routine MATRES evaluates the mesh.



THE FIVE BASIC TYPES OF DIRECT RELATIONSHIPS BETWEEN TWO COMPONENT AREAS

Determination of Functional Feasibility

At this point, the mesh is evaluated as regards its capability of lodging required functional relationships. This is attained first by verifying which components can potentially lodge each given function; next, by producing all alternatives that will give a correct functional lodging; and finally, by verifying if the alternative is valid as regards the requirement for adjacent relationships. That is, if the relations among functions can be met satisfactorily by the relations among component areas of the mesh.

Routine ALNODI solves this part. There shall be a number of solutions or none according to the possibilities of the mesh to cope with the particular problem.

Mesh Dimensioning

Once a satisfactory functional solution has been attained, a final process of mesh dimensioning must be realized.

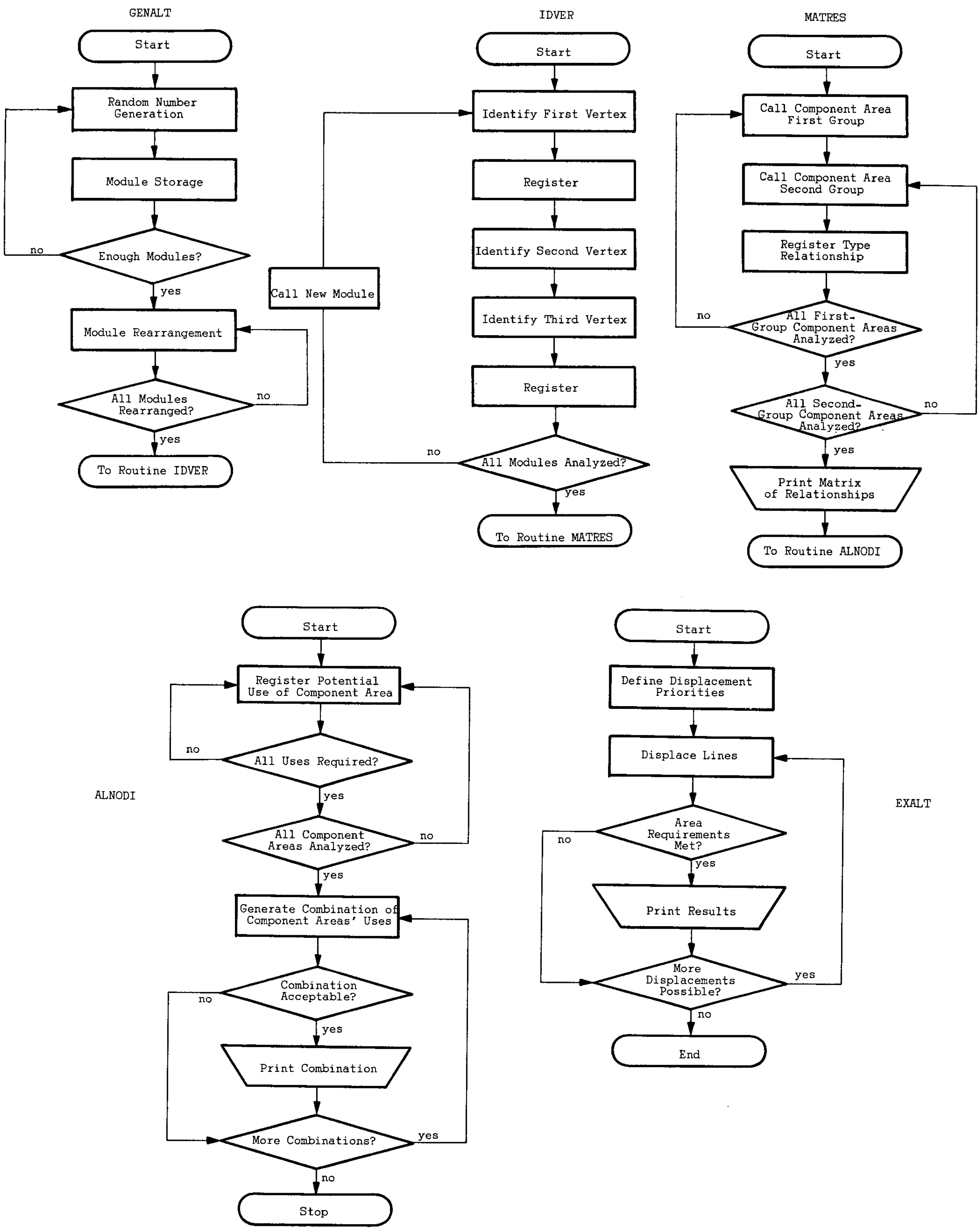
The method presented here is that of decomposing the mesh into vertical and horizontal lines (doing away at this stage with the adopted concept of "module") and resorting to a new modular concept in which the mesh is supposed to assume modular dimensions based upon 70 x 70 cms, 80 x 80 cms, or any other module identified with traditional architectural design.

The mesh is then (resorting to the principle of elasticity) "pushed back" into the upper corner of the envelope, until a minimum physical separation among components is attained and then each vertical or horizontal line begins to move forward after a given order.

At each move, the deforming mesh is questioned in regards to its capability for lodging dimensionally acceptable functions; the functions remain assigned to a given space and do not change their position throughout the length of the process.

There may be a great deal of solutions produced according to the flexibility of the mesh and these in turn should be refined by the setting of new, more sophisticated restrictions.

Routine EXALT carries out this task described above.



MACRO DIAGRAMS OF SYSTEM ROUTINES

CONCLUSION

The system intends to be of help to the designer in search for unhindered solutions, by providing him with a way of breaking out of the limits imposed by personal restrictions and knowledge (6).

It can help him, thus, to gain a broader view of the field of possibilities available for the search of a design solution. It could also, with further refinements imposed, provide a quick way of dimensioning a designer's own schemes and adjusting them until satisfactory solutions are achieved.

REFERENCES

1. Oystein Ore, Theory and Application of Graphics, 1963, Random House, New York
2. Rafael Leoz de La Fuente, Redes y Ritmos Espaciales, 1969, Editorial Blume, Barcelona
3. Jean Cousin, Organisation Topologique de l'Espace Architectural, 1970, Les Presses de l'Universite de Montreal
4. L. Grimaldi and G. Velez, La computacion electronica aplicada a la Arquitectura, I, II, Revista PUNTO, Nos. 38, 40, 1969-1970
5. G. Velez, Los programs PARC: Resultados parciales de una investigacion, (internal paper) Banco Obrero. D.D.AI, 1970
6. E. V. Krick, Methods Engineering, 1962, John Wiley and Sons, New York