COMPUTER-AIDED SPACE PLANNING

William R. Miller
Architectural Systems Engineer
Albert C. Martin and Associates
Planning/Architecture/Engineering
Los Angeles, California

Abstract

This paper describes the application of MATRAN-III, a computer program developed by the author, to planning the space for a branch bank in Southern California. MATRAN-III is used to identify clusters of elements within a set of interrelated elements. The program accepts any arbitrary listing of the elements and their proximal relations and creates an adjacency matrix. This matrix is block diagonalized yielding visually recognizable patterns which can be mapped into line-dot diagrams. These diagrams can then be presented to a designer providing him with a memory pattern from which we can begin laying out the geometric configuration of the space.

Introduction

Space planning is an architectural phrase used to describe the process of locating the functional spaces within a building facility. These spaces may be either rooms, as is the case with laying out building floor plans, or work stations as in the problem of office landscaping. In either case, the design process is essentially the same.

The typical non-computerized approach to space planning can be idealized as a three-fold process of information gathering, trial and error design and solution presentation. In actuality, many other things are included in the total design process, such as the emotional feel of a building, the flow of space, the transition between spaces, philosophical ideals, rhythm, pattern, texture, etc.

The information gathering portion of the process is referred to as the "program development" phase. A "program" is a written document (not to be confused with a computer program) which defines the space planning problem. The contents of this document are usually based on conversations with the client, the results of questionnaire surveys and professional knowledge. In essence, the program document states the design requirements for the proposed facility.

After the program has been written, the project enters the design phase. The designer studies the program until he feels he has sufficient knowledge of the problem to begin laying out a plan for the facility. The information most pertinent to this process consists of a list of all the functional spaces within the facility, the square footage requirements for each of the spaces and a list of all the adjacency requirements between spaces. Figure-1 shows a sample set of this type of information.

The designer's early plans are generally in the form of bubble diagrams. A bubble diagram, as shown in Figure-2, displays each functional space as a free-hand oval (hence the term "bubble") such that its area is directly proportional to the amount of square footage required for that specific space. Adjacency requirements are shown by drawing the ovals of adjacent areas tangent

ID	NAME	RELATES	AREA
1	SALES DIRECTOR	2, 3	180
2	CONFERENCE ROOM	1, 3	250
3	SALES SECRETARY	1, 2, 4	100
4	LOBBY	3, 5	200
5	PROJECT SECRETARY	4, 6, 7	100
6	LIBRARY	5, 7	200
7	PROJECT DIRECTOR	5, 6	180

Figure-1: Adjacency Requirements

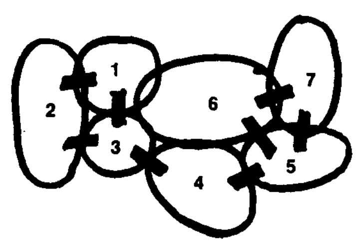

Figure-2: Bubble Diagram

to each other. The bubble diagram used by the designer is an attempt to relate many things not relative, yet this diagram helps organize the designer's thought on many points simultaneously. Many bubble diagrams may be drawn for each of the various parts of a given facility. This process eventually leads to a final diagram incorporation of all the functional spaces within the facility and as many of the adjacency requirements as the designer was able to satisfy. After an acceptable bubble diagram has been created, it is translated into a rectilinear plan.

Presentation of the design solution to the client usually emphasizes just that, the design solution. The design process is typically not presented. The total process of developing a program, designing the facility and presenting the solution is usually time consuming and costly. The solution is generally suboptimum and open to much criticism.

Methodology

The methodology described herein concerns that portion of the space planning process involved with the juxtaposition of the functional spaces within a proposed facility given their desired adjacency requirements. The method utilizes an adjacency matrix for defining the elements (functional spaces) and their proximal relationships. A complete description of this methodology is published elsewhere (Ref. 4). The following description has been included for the sake of completeness.

The previous sample data is shown formated as an adjacency matrix in Figure-3. The rows and columns of the matrix are labeled with the element identification numbers. Note that the numbering sequence must be the same for both the rows and columns. The presence of a one (1) in the interior of the matrix represents the existence of a relation between the addressing elements. If a cell is blank, then no relation exists between the addressing elements. For the sake of graphical clarity it is assumed that all elements relate to themselves, thus there is a "base diagonal" of ones running from the upper left corner to the lower right corner of the matrix. Since all relations are assumed bi-directional (if A is next to B, then B must be next to A) the matrix is symmetric about the base diagonal.

This same set of data can also be shown using a line-dot diagram (relational diagram) as indicated in Figure-4. The functional areas are represented by the nodes and the adjacency requirements between functional areas are shown as lines between nodes. This diagram is similar to the designer's bubble diagram. In practice it has been found that the line-dot diagram actually shows the relational structure of a space planning problem more effectively than the designer's traditional bubble diagram. The bubble diagram tends to obscure the relational structure of the problem by showing too much information, i.e., square footage and non-required adjacency constraints. The line-dot diagram shows only that information necessary to delineate the relational structure of the problem.

	1	2	3	4	5	6	7	
1	1	1	1					SALES DIRECTOR
2	1	1	1					CONFERENCE ROOM
3	1	1	1	1				SALES SECRETARY
4			1	1	1			LOBBY
5				1	1	1	1	PROJECT SECRETARY
6					1	1	1	LIBRARY
7					1	1	1	PROJECT DIRECTOR

Figure-3: Adjacency Matrix

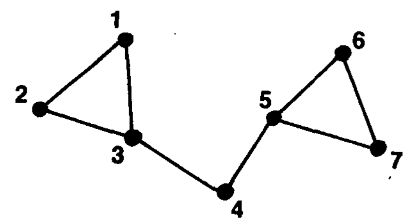

Figure-4: Relational Diagram

Note the visual relationship between the adjacency matrix of Figure-(3) and the diagram of Figure-(4). There are two clusters, or groups, of elements in the diagram and there are two clusters of one's in the matrix. For example, elements 1, 2, and 3 are clustered in the diagram with a corresponding cluster of one's lying in the intersecting rows and columns representing the same elements in the matrix.

Note, is we would have numbered the notes of our diagram, say as in Figure-5, we would find that the corresponding adjacency matrix as shown in Figure-6, would be nonsensical. If however, we could diagonalize this matrix by rearranging

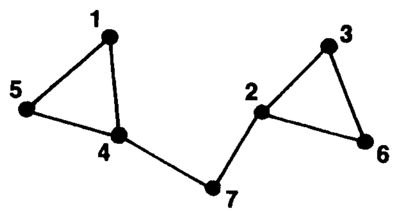

Figure-5: Revised Relational Diagram

	1	2	3	4	5	6	7	
1	1			1	1			SALES DIRECTOR
2		1	1			1	1	PROJECT SECRETARY
3		1	1			1		LIBRARY
4	1			1	1		1	SALES SECRETARY
5	1			1	1			CONFERENCE ROOM
6		1	1			1		PROJECT DIRECTOR
7		1		1			1	LOBBY

Figure-6: Revised Adjacency Matrix

	1	5	4	7	2	3	6	
1	1	1	1					SALES DIRECTOR
5	1	1	1					CONFERENCE ROOM
4	1	1	1	1				SALES SECRETARY
7			1	1	1			LOBBY
2				1	1	1	1	PROJECT SECRETARY
3					1	1	1	LIBRARY
6					1	1	1	PROJECT DIRECTOR

Figure-7: Revised Adjacency Matrix
Block Diagonalized

the corresponding rows and columns of the matrix to that of Figure-7, we would once again identify the visual relationship between the diagram and the matrix.

This visual relationship allows one to find the clusters for any set of interrelated elements, assuming:

1. The only attribute assigned to an element or relation is that of existence. (Intensity of a relation between elements may be included.)

2. All relations are bi-directional.

3. All elements relate to themselves.

The methodology is summarized as follows:

1. List the elements in the problem set and their relationships in terms of an adjacency matrix.

2. Block diagonalized the adjacency matrix.

3. Visually identify the clusters forming along the base diagonal.

4. Map this information into a relational diagram.

MATRAN-III

As one might well guess, the critical step of this process involves manipulating the rows and columns within the matrix such that the unit valves are forced to cluster along the base diagonal. If one were to attempt this task manually, he would be forever exchanging rows and columns. With the aid of a computer, however, it is possible to write a program that will perform this manipulation automatically. Such a program, MATRAN-III, has been written. It is composed of the following major segments: data input, data manipulation and data output.

Input data is comprised of three basic types: job title information, element base data and run status data. Job title data allows the user to identify the project name, number and descriptive comments. Element base data consists of the element identification number, the element name, the element area (sq. ft.), a list of the related elements and the degree importance of each relationship (as defined on a scale of high, medium or low). Run status data identified whether or not any set of relations within a given weighted range is to be supressed.

The program organizes this data into a format ready for matrix manipulation by creating an adjacency matrix such that the rows and columns are labeled with the element identification numbers. The field of the matrix is loaded with importance values associated with the relations between elements. The value for a non-relation equals zero (0), for a low relation: one (1), for a medium relation: two (2), and for a high relation: three (3). The main manipulation subroutine then causes the computer to block diagonalize the matrix by interchanging various rows and corresponding columns such the relational values converge to the base diagonal. In a weighted analysis the higher values converge to the diagonal faster than the lower values.

The computer first identifies two

rows (and corresponding columns) for possible exchange. The value of the matrix is then calculated. (The value of the matrix at any given time is equal to the sum of the products of the relational value within a cell, either 0, 1, 2, or 3, multiplied by its distance, number of cells, from the base diagonal.) This calculated value is placed in temporary memory. The computer then calculates the projected new value of the matrix assuming that the rows and columns in question are exchanged. This is done without actually making the exchange. If the value of the matrix prior to the proposed exchange is greater than the projected value then an improved condition has been identified and the computer is allowed to make the exchange. If the value of the matrix prior to the exchange is less than or equal to the projected value then no improvement is projected and the exchange does not take place. The computer next identifies two more rows (and corresponding columns) for possible exchange. This process continues until no further improvement can be made.

Once the computer has blocked diagonalized the data matrix it is ready to output information. The output is composed of a list of all the elements and their relations (this is simply a reflection of the input data), a print out of the data matrix (the imput data formated as an adjacency matrix) and a print out of the solution matrix (diagonalized data matrix).

The solution matrix is then visually interpreted by manually partitioning the matrix to identify clusters and other recognizable patterns along the diagonal. The partitioned matrix is then mapped into a relational diagram. This diagram is presented to the designer to provide him with a memory pattern from which he can begin laying out or plan for the facility.

Application Description

The following example describes the application of MATRAN-III to planning the space for a Wells Fargo branch bank in Pomona, California. The description concentrates on the computerized space planning portion of the problem. Brief comments are made regarding the program's interface with the total design process.

The first step of the design process was to establish the design requirements for the proposed branch bank. This task was performed jointly by architectural designers and banking operations experts. From the set of requirements came a list of the functional areas and their corresponding proximal relationships. The geometric constraints were neglected at this point. The only attribute assigned to each functional space was that of existence. The relations, however, were said to exist with either a high, medium or low degree of importance. Since relations between spaces were based on geographical proximity all relations were assumed to be bi-directional.

This information was then listed on standard coding forms. Data cards were punched and submitted to a UNIVAC 1108 computer under the control of the MATRAN-III program. The computer first created a symmetric adjacency matrix (data matrix) and identified those areas that were totally unrelated to the problem set. A revised adjacency matrix was then generated eliminating the unrelated areas from the matrix. A computer then block diagonalized the revised matrix. The diagonalized matrix (solution matrix) was then printed utilizing various graphic embellishments. The computer output is shown in Figures-8, 9 & 10 .

The next step was to visually partition the solution matrix. This partitioned matrix provided the basis for the relational diagram shown in Figure-11. This relational diagram was then presented to the designer.

The designer started laying out the geometric space by first planting those elements of the relational diagram that had definite geographical locations. These elements were:

- 9 Trash Pickup
- 15 Garey Avenue
- 19 Walk Up Teller
- 20 Drive Up Teller
- 21 Parking Lot
- 22 Alley

Once the fixed elements were planted those adjacent to the fixed elements were located. The process continued until all of the elements were positioned within the geometric confines of the facility. Figure-12 shows the element nodes planted on the site. Some of the elements were located on a second level and thus are not shown in this figure. Also, element No. 23, the patio, was deleted.

Following this, the nodes were allowed to grow into bubbles such that the size of each bubble was directly proportional to the square footage requirement for the particular func-

```
PROGRAM:  MATRAN - II
SOURCE:   ALBERT C. MARTIN AND ASSOCIATES

PROJECT:  WELLS FARGO BANK, POMONA                         69136-01-500
          THIS ANALYSIS INCLUDES WEIGHTED RELATIONS


INPUT DATA
==========
```

ID	ELEMENT NAME	AREA	RELATES TO
1	WORK AREA	1500	2H 12H 13H 16M 17M 19H 20H 28M
2	TELLERS COUNTER	300	1H 4H
3	NEW ACCOUNTS	200	4H 5M
4	PUBLIC BANK LOBBY	1700	2H 3H 5H 6L 7H 13H
5	OFFICES PLATFORM	800	3M 4H 11M
6	ESCROW	800	4L
7	BUILDING LOBBY	1600	4H 15H 22H 25H 27H
8	JANITOR	100	9L 16L 17L
9	TRASH PICKUP	100	8L 22M
10	SAVINGS DEPOS CONF ROOM	150	13H
11	OFFICERS CONF ROOM	150	5M
12	VAULT	400	1H 13H
13	SAFE DEPOSIT LOBBY	200	1H 4H 10H 12H 14H
14	COUPON BOOTHS	100	13H
15	GAREY AVENUE	0	7H
16	MEN	200	1M 8L 28M
17	WOMEN	200	1M 8L 28M
18	STORAGE	250	0
19	WALK-UP TELLER	50	1H 23H
20	DRIVE-UP TELLER	50	1H 22H
21	PARKING LOT	0	22H
22	ALLEYWAY	0	7H 9M 20H 21H 23H
23	PATIO	0	19H 22H
24	LEASE FLOOR	10	25H 26H 27H
25	STAIRWAY A	200	7H 24H
26	STAIRWAY B	200	24H
27	ELEVATORS	150	7H 24H
28	STAFF ROOM	300	1M 16M 17M

```
NUMBER OF ELEMENTS =  28
NUMBER OF RELATIONS   H =  24   M =   8   L =   4   TOTAL =  36
TOTAL AREA IS NOT AVAILABLE
```

Figure-8: Input Data

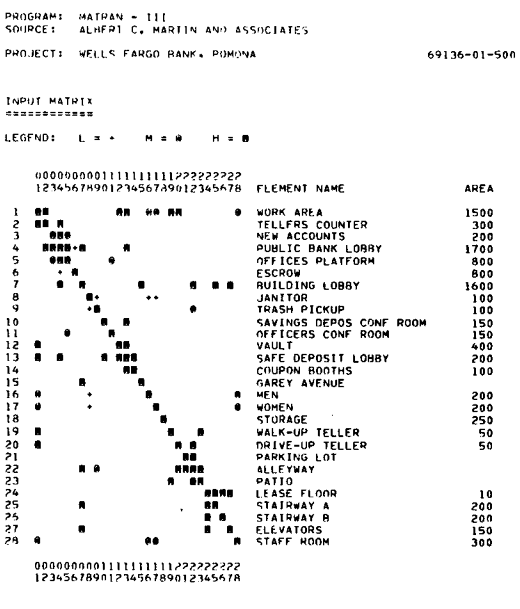

Figure-9: Input Matrix

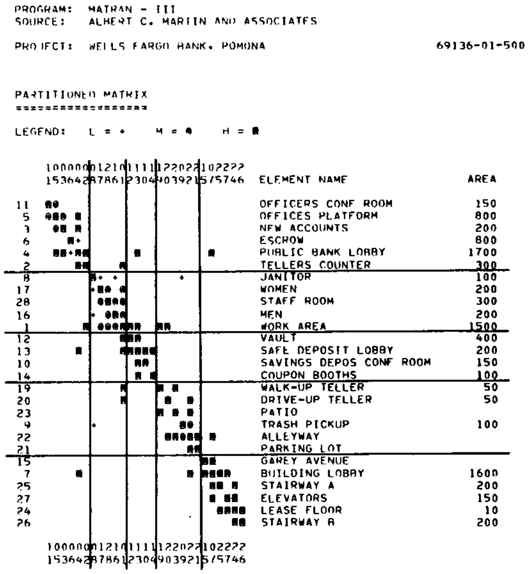

Figure-10: Partitioned Matrix

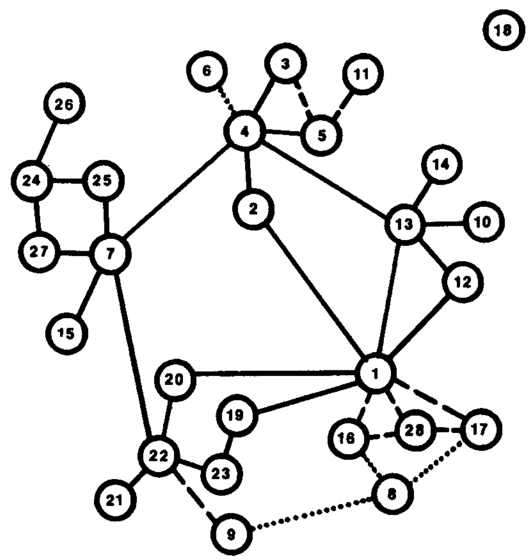

Figure-11: Relational Diagram

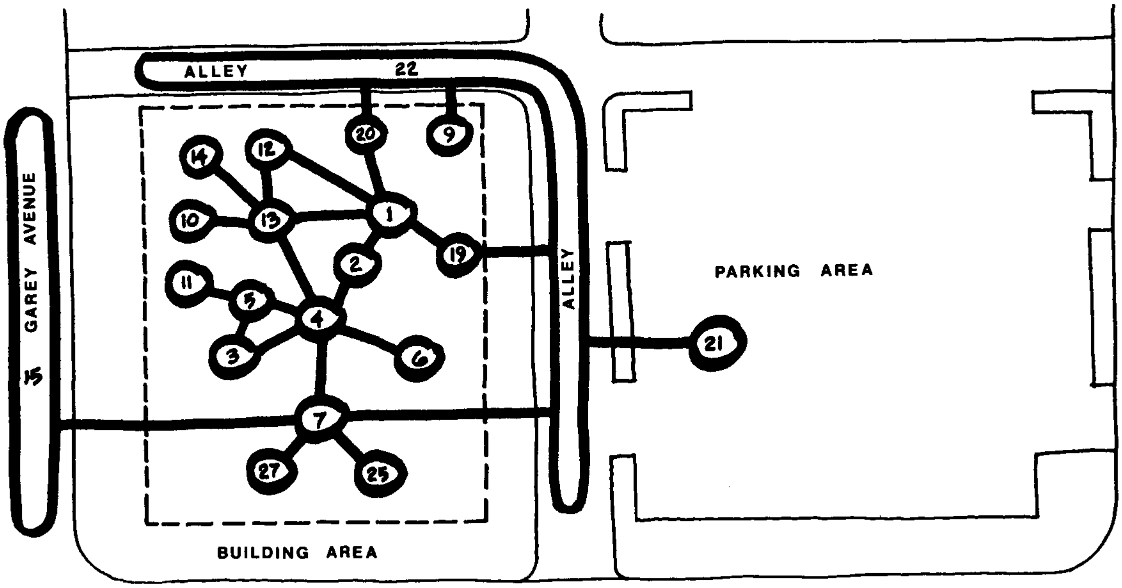

Figure-12: Ground Floor Element Nodes Planted on the Site

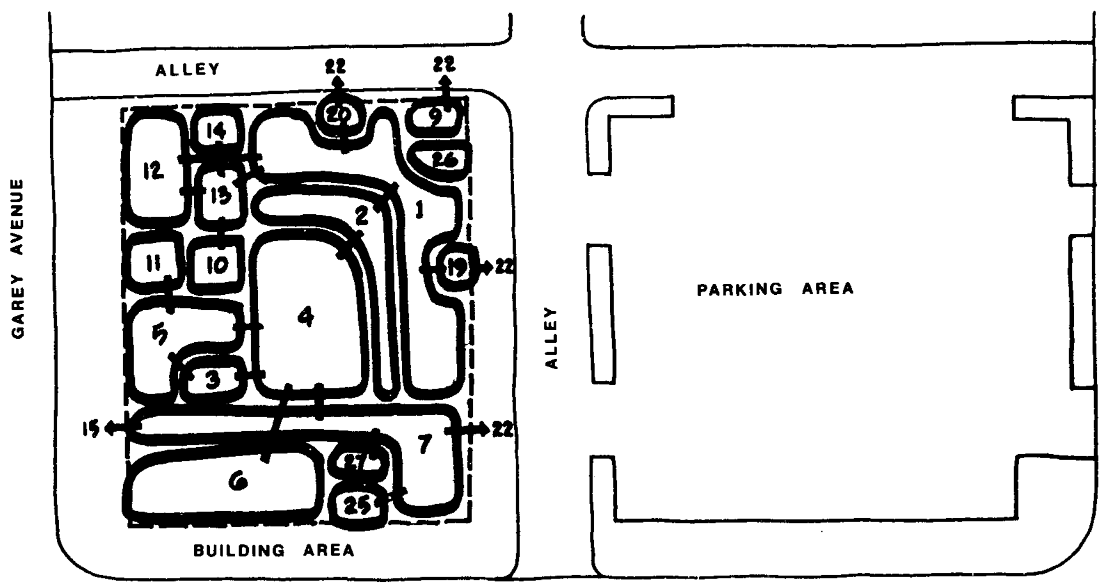

Figure-13: Ground Floor Project Bubble Diagram tional space. Conflicts occured, however, in that two or more elements, would often be competing for the same area. Some of these conflicts were resolved by making design compromises, others were resolved by simply relocating the element. This process lead to the architectural bubble diagram as shown in Figure-13. The balance of the design process followed traditional methods.

Conclusion

MATRAN-III is a perfectly general program which can be used to solve any relational problem which can be described as a non-directed graph and where the objective is to identify sub-clusters of elements. The program allows the user to simply list the elements in a problem set and their inter-relations and receive in return an optimum relational diagram.

Concerning space planning the advantages of the computer-aided approach over the traditional intuitive methods are:

1. The relational solution generated is optimum.

2. The time frame required to generate the solution is reduced.

3. Total production costs are reduced.

An additional advantage is that the explicit nature of the methodology helps the designer isolate particular parts of the total design process and to specialize his thoughts accordingly. The designer will (should), however, always return to the other processes necessary to design whatever they may be.

Acknowledgements

I should like to acknowledge assistance provided by various groups and individuals during the development and implementation of MATRAN-III. Appreciation is given to James Lord, Principal Systems Engineer; Thomas Mason, Director of Interior Design and Paul Jensen, Project Director, all with Albert C. Martin and Associates, for their encouragement and suggestions during the development of the program. Special acknowledgement is given to Wells Fargo Bank for providing MATRAN-III with the space planning problem described in the paper. I should also like to thank David Martin, Anthony Schnarsky, James Olsten, and Vahe Khachooni for their comments and assistance. Final appreciation is given to Albert C. Martin and Associates for providing an environment conducive to research and development work and for the resources that supported this effort.

Bibliography

1. Ball, G. H., "A Comparison of Some Cluster Seeking Techniques", Stanford Research Institute, November, 1966.

2. Ball, G. H. and Hall, D. J., "Background Information on Clustering Techniques", Stanford Research Institute, June, 1969.

3. Busacker, R. G. and Saaty, T. L., Finite Graphs and Networks, McGraw-Hill, 1965.

4. Miller, W. R., Khachooni, V. and Olsten, C. J., "Matrix Method for Grouping an Inter-related Set of Elements", Proceedings: Environmental Design Research Association Conference, Chapel Hill, N. C., June, 1969.

5. Propst, R., The Office: A facility based on change, The Business Press, 1968.