CLUSTR: A Program for Structuring Design Problems

Murray A. Milne Associate Professor
School of Architecture and Urban Planning
University of California at Los Angeles

Abstract

CLUSTR is a computer program which assists the designer in finding the structure inherent in his design problem. The designer supplies the list of elements which define the design problem, and then decides which of these elements are related. The computer decomposes the problem into subsets in which each element is related to every other element. In theory each of these subsets represents the smallest "structural" component of the problem: a coherent functional or behavioral sub-system. The most closely related subsets are then combined into larger clusters. This process continues until all clusters have been recombined. The computer then draws a diagram to show how these subsets are combined to form the final problem structure. The computer also identifies the dominant elements at each node in the structure to assist the designer in finding the solutions to each sub-problem.

How does a designer begin to design?

To me the most intriguing phase in the process of design is its very beginning. At the outset the designer has only a vague and uneasy feeling that something in the environment is not quite right -- there is anomaly. He is unable to conceptually categorize or name this anomaly because, in fact, it is a void, a null set. It is the need for a "something" which does not yet exist. As the designer begins to search for this "something," the way he initially structures his problem will have an impact on each succeeding phase in the process.

Each phase of the design process has its own unique characteristics. Tools have been developed to assist the designer during many of these phases: laying out components, designing structural elements, analyzing cost, specifying production procedures, etc. But the initial phase, when the designer begins to define his problem, is probably the most complex and the least understood.

The designer first tries to identify all the elements of his problem. These are the goals, requirements, constraints, or performance specifications which his final solution must satisfy. At this point, the situation seems to be in complete chaos. The designer's dilemma is not the lack of information; in fact, he is usually overwhelmed by more knowledge and more data than he can handle. Instead, his difficulty is the lack of structure. His task is to somehow organize all this information into a precise and consistent problem description and then break it down into manageable sub-problems. By the process of articulating the problem elements and then establishing their relational structure, the designer begins to replace chaos and uncertainty with content and order.

In this paper we are concerned primarily with design problems for which there is no known well-developed prototypical solution.

This means that the process of solving a design problem is much more than simply a matter of performing a set of precisely specified operations or of selecting from a finite list of acceptable solutions. We can therefore define design as a process which organizes information in a way that it has never been organized before such that it satisfies a set of previously stated criteria. This is sometimes called a "creative" process.

Before we can develop computerized techniques to assist the designer, we must try to discover exactly how his decision-making process operates, especially during these initial phases of the design process when he is confronted with a very complex problem for which there is apparently no well-developed protypical solution.

Unfortunately, the human cognitive capacity is not well suited for dealing with conceptual tasks requiring the simultaneous manipulation of a great many different elements which are interrelated in a great many different ways. Psychologists tell us that tasks of this size and complexity exceeds the capacity of the human's immediate memory. On the average, humans are able to simultaneously manipulate in immediate memory no more than seven simple concepts (plus or minus two). (1) If they are complex concepts, each having a multitude of attributes, he is doing well to consider more than two or three. But in the beginning of the design process, there may be hundreds of problem elements which must be considered. Therefore, designers have devised techniques which help to overcome this cognitive overload. First, they externalize as much information as possible; they write down lists and draw thumbnail sketches. Second, they try to simplify the problem by encoding or categorizing as many separate pieces of information as possible into one subsolution, and then they have to remember only this one sub-solution as they continue to grapple with the rest of the problem. Obviously, there are serious dangers inherent in both techniques; it is unavoidable that information is lost, elements are categorized incorrectly, and conclusions are reached prematurely. Although designers are exceptionally good at a great many phases of design, their overwhelmed cognitive capacities and their less-than-perfect "external memory" techniques indicate ways in which the computer could be used to assist the designer, especially at the very beginning of the design process.

If you ask designers to tell you how they begin to design, although their descriptions usually are rather vague, they all seem to have the same general pattern. First the designer tries to figure out the most important factors in his problem. Then he picks a part of the problem which he thinks he can solve. The easiest parts are separate little problems that involve only a few factors, or else remind him of problems that he has solved many times before. The hardest parts are problems that involve many factors which are related to each other in many different ways. When he has either found a solution or is tired of working on it, he moves on and works on another part of the problem. He keeps track of his progress by drawing sketches or making notes, or by trying to remember facts or images of what he has done. Gradually he begins to fit together his solutions for different parts of the problem. Sometimes he discovers new factors in the problem which he had not been aware of before, which means he must go back and change or throw out some of the solutions he has already found. Eventually he either solves all the parts of his problem and puts them together into the final solution, or else he quits and goes out for coffee.

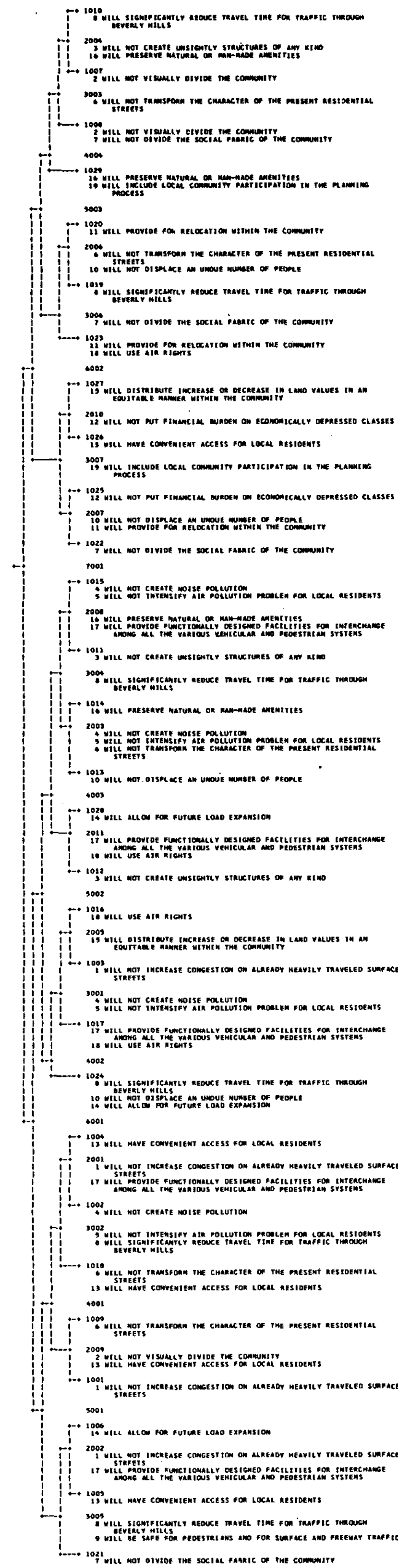
Although this scenario may appear to be hopelessly vague, if we carefully analyze what the designer has said, we find that he has given us a great deal of useful information. Without realizing it, he is following a simple model of systems analysis. He is telling us that he begins by establishing a set of finite elements and that he occasionally adds new elements to this set. This set of elements apparently contains the criteria for evaluating the various outcomes of the process. He partitions or decomposes this set of elements into sub-problems. There are apparently two methods of doing this: one is by pattern recognition, and the other is by a mini-max procedure for finding the subset of elements with minimum size and maximum isolation. Once a subset is identified, the procedure for finding the "solution" probably involves generating an hypothesis and then evaluating it against the elements in the sub-set. There is no evidence to indicate that this is an optimization process, but rather it would seem to be a decision-making process based on satisficing criteria. (2) There is a time constraint which terminates unsuccessful searches for sub-solutions and which thus avoids infinite loops. There are two types of long-term random access memories in which are stored both graphic and verbal data. The process is iterative. There also appears to be a procedure by which the various sub-solutions are recombined or integrated into the final solution. The total process has a termination criteria based either on a time constraint or on the discovery of a final solution which satisfies the set of all previously defined evaluation criteria.

Neither of these two obviously parochial descriptions completely or satisfactorily explains all aspects of the design process. The most challenging design problems demand more creativity than computers can muster and more precision than designers can provide. Design obviously must be an interactive process in which the computer performs the tedious computation and massive data storage and retrieval functions, while the human provides the information generation, pattern recognition, problem solving, evaluation, and management functions.

CLUSTR

This computer program assists the designer in finding the structure inherent in his design problem. The designer supplies the list of elements which define the design problem and then decides which of these elements are related. The computer decomposes the problem into subsets in which each element is related to every other element. In theory each of these subsets represents the smallest "structural" component of the problem: a coherent functional or behavioral sub-system. The most closely related subsets are then combined into larger clusters. This process continues until all clusters have been recombined. The computer then draws a diagram to show how these subsets are combined to form the final problem structure. The computer also identifies the dominant elements at each node in the structure to assist the designer in finding the solutions to each sub-problem. (Fig. 1)

Graph theorethic models and network analysis techniques were first applied to the analysis of architectural design problems in Germany in 1959 at the Hochschule fur Gestaltung, at Ulm. (3) Shortly thereafter, Christopher Alexander described in detail a method in which graph theory was applied to the task of structuring design problems. (4) Although Alexander has abandoned this method, other researchers have continued to develop new applications of network analysis and graph theory to computer-assisted design programs. (5) CLUSTR is one of these.



1. FINAL PROBLEM STRUCTURE: An illustration of CLUSTR applied to a small hypothetical design problem.

Problem Definition

The process of developing a list of problem elements has been given extensive discussion elsewhere. (6) All elements should be written at the same level of generality, should deal unambiguously with only one issue, should not overlap with other elements, and all the elements taken together should completely describe the problem, leaving no issues uncovered. At the inception of the design process, the designer can usually define most of the elements in his problem. These problem elements may be in the form of goals, requirements, constraints, or performance specifications. (7) The designer should use whatever format seems most appropriate as long as each element can function as an evaluation criteria against which he can judge the various components of his solution. (Fig. 2)

Interaction

The designer establishes the relational structure of his problem by deciding which pairs of problem elements interact with each other. This decision is usually facilitated if the designer asks the question, "Will my solution of element 'A' either conflict or concur with my solution of element 'B'?" If the answer is "yes," an interaction exists. If the solution to element "A" is indifferent to the solution to element "B," then no interaction exists. The designer proceeds in this manner to test every pair of elements. To be considered a part of the design problem, every element must interact with at least one other element. (Fig. 3)

Problem Structure

If the problem is small, its structure can be revealed by manually plotting a network of the problem statements (nodes) and their interactions (links). (8) By examining this network diagram, the designer should be able to pick separate little sub-problems which he thinks he can easily solve. He also should be able to locate the more complex parts of the problem, where many elements interact with each other in many different ways. If he is lucky he might also be able to see how to separate these large complex areas of the problem into smaller, more manageable sub-problems. Once he has identified and solved all of the sub-problems, the network diagram will show how they should be combined into the final solution. For very simple design problems, hand-drawn networks diagrams are quite useful. But most design problems consist of at least fifty problem elements and are seldom less

LISTING OF VERBAL TEXT STATEMENTS

1 WILL NOT INCREASE CONGESTION ON ALREADY HEAVILY TRAVELED SURFACE STREETS

2 WILL NOT VISUALLY DIVIDE THE COMMUNITY

3 WILL NOT CREATE UNSIGHTLY STRUCTURES OF ANY KIND

4 WILL NOT CREATE NOISE POLLUTION

5 WILL NOT INTENSIFY AIR POLLUTION PROBLEM FOR LOCAL RESIDENTS

6 WILL NOT TRANSFORM THE CHARACTER OF THE PRESENT RESIDENTIAL STREETS

7 WILL NOT DIVIDE THE SOCIAL FABRIC OF THE COMMUNITY

8 WILL SIGNIFICANTLY REDUCE TRAVEL TIME FOR TRAFFIC THROUGH BEVERLY HILLS

9 WILL BE SAFE FOR PEDESTRIANS AND FOR SURFACE AND FREEWAY TRAFFIC

10 WILL NOT DISPLACE AN UNDUE NUMBER OF PEOPLE

11 WILL PROVIDE FOR RELOCATION WITHIN THE COMMUNITY

12 WILL NOT PUT FINANCIAL BURDEN ON ECONOMICALLY DEPRESSED CLASSES

13 WILL HAVE CONVENIENT ACCESS FOR LOCAL RESIDENTS

14 WILL ALLOW FOR FUTURE LOAD EXPANSION

15 WILL DISTRIBUTE INCREASE OR DECREASE IN LAND VALUES IN AN EQUITABLE MANNER WITHIN THE COMMUNITY

16 WILL PRESERVE NATURAL OR MAN-MADE AMENITIES

17 WILL PROVIDE FUNCTIONALLY DESIGNED FACILITIES FOR INTERCHANGE AMONG ALL THE VARIOUS VEHICULAR AND PEDESTRIAN SYSTEMS

18 WILL USE AIR RIGHTS

19 WILL INCLUDE LOCAL COMMUNITY PARTICIPATION IN THE PLANNING PROCESS

2. ELEMENTS: The small hypothetical problem used to illustrate this method was the design of a controversial link in the Los Angeles Freeway System through the City of Beverly Hills. The design team has defined these elements as a first approximation of an evolving problem description.

INTERACTION MATRIX

1	T																		
2	T	T																	
3	F	T	T																
4	T	F	F	T															
5	T	F	F	T	T														
6	F	T	T	T	T	T													
7	F	T	F	F	F	T	T												
8	T	F	T	T	T	T	T	T											
9	T	F	F	F	F	F	T	T	T										
10	F	F	F	T	T	T	T	T	F	T									
11	F	F	F	F	F	T	T	F	F	T	T								
12	F	F	F	F	F	F	F	F	F	T	T	T							
13	T	T	F	F	T	T	F	T	T	F	F	T	T						
14	T	F	F	F	F	F	F	T	T	T	F	F	F	T					
15	T	F	F	T	T	F	F	F	F	F	F	T	F	F	T				
16	F	T	T	T	T	T	F	T	F	F	F	F	F	F	F	T			
17	T	F	T	T	T	F	F	T	T	F	F	F	T	T	F	T	T		
18	F	F	T	T	T	F	T	F	F	F	T	F	F	T	T	F	T	T	
19	F	F	F	F	F	F	T	F	F	T	T	T	T	F	T	T	F	F	T

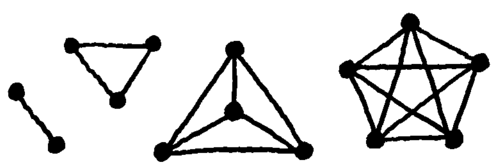
3. INTERACTIONS: The design team compared every pair of elements to determine whether or not an interaction exists. This data is displayed in a logical (true-false) matrix.

than 10% connected, which means that the resulting network would contain at least 245 links. Obviously graphic representa- · of this scale are far too complex to be of any use to the designer. Therefore, we depend on CLUSTR to produce a more usable representation of the problem structure.

After the designer has considered every pair of elements and has decided whether or not they interact, he inputs this information to CLUSTR, usually in the form of a square binary matrix.

Problem Decomposition

Given a symmetrical binary interaction matrix, CLUSTR identifies every simplex (i.e., every completely interconnected subset containing two or more elements). Thus, every element in the problem and every interaction appears in at least one simplex. A simplex is defined as a set in which all the elements interact with each other (such a set is also called a complete graph or a universal graph). (9)



If we are attempting to decompose a large complex network into its smallest components, the simplex would logically be the smallest indivisible unit. A simplex therefore represents the smallest and most coherent functional or behavioral sub-system the designer could consider. But still, the task of finding a valid solution sometimes turns out to be fairly challenging, because whenever the designer proposes a solution which satisfies one element in a simplex, it must simultaneously satisfy every other element in that simplex. This is because the designer had previously decided that the solution to each of these elements either conflicts or concurs with the solution to every other element in that simplex. It should be noted that simplex subsets are not necessarily disjointed, because an element may appear in more than one simplex. (Fig. 4)

Problem Recomposition

Once every simplex has been identified, the process of recomposing the problem can begin. When two simplexes are combined they form a cluster. Hundreds of different cluster-finding processes have been developed, more

```
THE SIMPLEX LIST

A SIMPLEX IS DEFINED AS A COMPLETELY CONNECTED SUBSET, THAT IS,
A CLUSTER IN WHICH EVERY ELEMENT INTERACTS WITH EVERY OTHER ELEMENT.
IN THEORY, A SIMPLEX IS THE SMALLEST "STRUCTURAL" COMPONENT OF THE PROBLEM.
IN PRACTICE, A SIMPLEX REPRESENTS A COHERENT FUNCTIONAL OR BEHAVIORAL
SUBSYSTEM.

THE NUMBER IN PARENTHESES IS THE NUMBER OF ELEMENTS IN THAT SIMPLEX


 1001 SIMPLEX ( 3) =    1   2  13

       1 WILL NOT INCREASE CONGESTION ON ALREADY HEAVILY TRAVELED SURFACE
         STREETS

       2 WILL NOT VISUALLY DIVIDE THE COMMUNITY

      13 WILL HAVE CONVENIENT ACCESS FOR LOCAL RESIDENTS


 1002 SIMPLEX ( 5) =    1   4   5   8  17

       1 WILL NOT INCREASE CONGESTION ON ALREADY HEAVILY TRAVELED SURFACE
         STREETS

       4 WILL NOT CREATE NOISE.POLLUTION

       5 WILL NOT INTENSIFY AIR POLLUTION PROBLEM FOR LOCAL RESIDENTS

       8 WILL SIGNIFICANTLY REDUCE TRAVEL TIME FOR TRAFFIC THROUGH
         BEVERLY HILLS

      17 WILL PROVIDE FUNCTIONALLY DESIGNED FACILITIES FOR INTERCHANGE
         AMONG ALL THE VARIOUS VEHICULAR AND PEDESTRIAN SYSTEMS


 1003 SIMPLEX ( 4) =    1   4   5  15

       1 WILL NOT INCREASE CONGESTION ON ALREADY HEAVILY TRAVELED SURFACE
         STREETS

       4 WILL NOT CREATE NOISE POLLUTION

       5 WILL NOT INTENSIFY AIR POLLUTION PROBLEM FOR LOCAL RESIDENTS

      15 WILL DISTRIBUTE INCREASE OR DECREASE IN LAND VALUES IN AN
         EQUITABLE MANNER WITHIN THE COMMUNITY
```

4. DECOMPOSITION: Based on the design team's decisions about element interactions, CLUSTR identified a total of 29 simplex subsets.

```
RECOMPOSITION STRATEGY

THE STRUCTURE OF THE PROBLEM IS REVEALED AS THE MOST COHERENT PAIRS OF
SUBSETSARE COMBINED INTO HIGHER LEVEL CLUSTERS.
COHERENCY IS COMPUTED AS THE RATIO OF EXISTING INTERACTIONS IN
THE DISJUNCTION OF THE TWO SUBSETS DIVIDED BY THE MAXIMUM POSSIBLE
NUMBER OF INTERACTIONS.
THE CONJUNCTIVE ELEMENTS COMPRISE A SMALLER SIMPLEX WHICH IS CONTAINED
IN THE NEW CLUSTER AND IN EVERY SUBSET BELOW IT IN THE STRUCTURE.

NOTICE THAT A CONJUNCTIVE ELEMENT MAY "DROP OUT" AT ANY LEVEL
IN THE STRUCTURE



 2001 CLUSTER IS FORMED BY ADDING TOGETHER SUBSETS   1004 AND  1002

        THE CONJUNCTIVE ELEMENTS ARE   1   5   8  17

              FROM  1004 DROP OUT  1 CONJUNCTIVE ELEMENTS

      13 WILL HAVE CONVENIENT ACCESS FOR LOCAL RESIDENTS

              FROM  1002 DROP OUT  1 CONJUNCTIVE ELEMENTS

       4 WILL NOT CREATE NOISE POLLUTION


 2002 CLUSTER IS FORMED BY ADDING TOGETHER SUBSETS   1006 AND  1005

        THE CONJUNCTIVE ELEMENTS ARE   1   8   9  17

              FROM  1006 DROP OUT  1 CONJUNCTIVE ELEMENTS

      14 WILL ALLOW FOR FUTURE LOAD EXPANSION

              FROM  1005 DROP OUT  1 CONJUNCTIVE ELEMENTS

      13 WILL HAVE CONVENIENT ACCESS FOR LOCAL RESIDENTS
```

5. RECOMPOSITION: The order in which various sets are combined is computed on the basis of their relational structure. The conjunctive elements which "drop out" at each level of the structure are also identified.

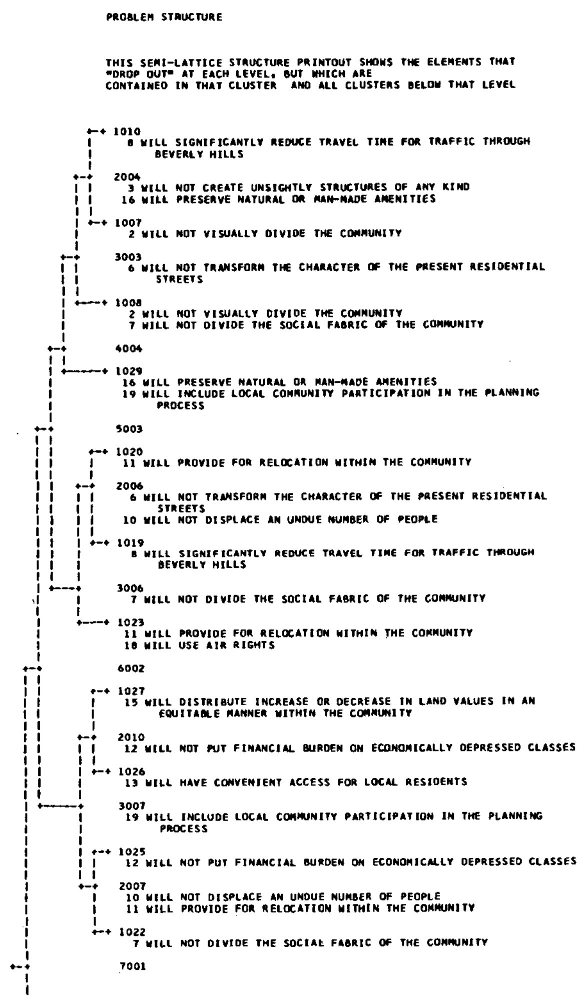
than one of which would be appropriate in his application (10) The choice depends in part on the density of the interaction matrix. The procedure which is currently used in CLUSTR is a function of the number of nodes (problem elements) and the number of links (interactions) in the disjunction of the two sets. It compares every pair of clusters or simplexes and computes the ratio of the total actual number of links to the theoretical maximum number of links. It then combines the pair that has the highest ratio. The process is repeated until all clusters have been combined and thus the problem is completely restructured. This procedure has proven quite satisfactory for matrices that are at least 10% dense. However, with less dense matrices this procedure occasionally combines sets which have no common elements. Work is currently under way to test the effectiveness of other cluster-finding procedures for matrices that are less than 10% density. (Fig. 5)

Problem Structure

As suggested above, network diagrams of large problems are usually so complex as to be effectively useless as graphic representations of the structure of design problems. Therefore, another technique had to be found which could display the structure of large design problems in a valid and usable way. Binary trees are simple enough but are probably invalid. It has been suggested that a semi-lattice would be a more accurate representation of the structure of design problems; however a true semi-lattice diagram for problems of any size might easily approach the complexity of the equivalent network. (11). Although the diagram of the problem structure which CLUSTR produces may appear to be a binary hierarchical "tree," it is in fact a modified hierarchical semi-lattice. This is because the sets on the first level (simplexes) are not exclusive but instead have a high degree of overlap, due to the fact that the same element may appear in more than one simplex.

Conjunctive Elements

When two simplexes are combined, their area of "overlap" contains a smaller simplex. This smaller simplex is made up of only those elements which are common to both simplexes and are called the conjunctive elements of the new cluster. As other simplexes and clusters are combined with this cluster, the number of conjunctive elements gradually decreases until at a certain point in the "tree" they disappear. A conjunctive element is one which appears in a given cluster



6. FINAL PROBLEM STRUCTURE: This section of the "tree" diagram illustrates how three of the major problem areas are structured. The first deals with the way the proposed freeway might change the physical character of this residential community. The second describes the affect the freeway might have on the social structure of the community. The third identifies the issues which will be contended in the participatory decision-making process.

and in every cluster or simplex below it in the "tree." Thus by reading the highest conjunctive elements in each branch of the "tree" plus every conjunctive element below it, the designer can get a fairly reliable indication of the dominant issues which are common to all the design problems that he will have to solve in this particular branch. By using the conjunctive elements in this way, the designer can identify the dominant problem elements and thus can more quickly find the solution to each simplex. (Fig. 6)

Solution Finding

Once the designer has found a solution for each simplex, theoretically he has solved the total design problem, because every problem element and every interaction has been accounted for. (12) Now all that remains is simply to combine the various sub-solutions into one final solution. In practice, however, a simplex usually represents an under-constrained problem description and so there is sometimes more than one solution to each simplex and the one that is selected may not fit easily with the solutions to all the other simplexes. Therefore, a certain amount of redesign is necessary as the solution to each simplex is carried up the structure towards a final design solution. It has been found that if solutions to sub-problems emphasize verbal descriptions rather than diagrammatic representations, it will probably be easier to directly combine (concatinate) sub-solutions as they are carried into the problem structure. (13) This type of solution representation will be increasingly useful as designers are confronted with more and more "interdisciplinary" problems for which there are "non-form" solutions, for example, problems which are best solved by organizational, administrative, political, or educational means.

Discussion

A question is often asked about the validity of binary interaction decisions as opposed to using, for instance, a weighing scale. It is true that other problem structuring algorithms of this type have attempted to use weighted interactions. Unfortunately, these approaches encounter considerable computational difficulty, especially in problem decomposition. On the other hand, forcing the designer to make those difficult unequivocal binary interaction decisions induces him to consider all of the design implicators much more carefully than he otherwise might, especially if he were tempted to avoid difficult issues by simply selecting the middle point on a weighted scale. In any event, because the designer invariably defines design sub-problems in terms of discrete elements, ordinal information would be of little value. In fact, the designer creates a far richer and more subtle kind of relational structure at the time when he "designs" a solution which satisfies a given set of elements.

A new feature which is currently being implemented in CLUSTR will also allow the designer to input a verbal description of how and why he decided that a particular pair of elements interact. Later this information will be retrieved and printed out as part of the problem structure as an aid to the designer. In this way he can easily recall his earlier decisions as he begins the process of finding design solutions for each sub-problem.

An essential aspect of this approach is that all decision-making criteria must be stated in plain language. No design solution is justified unless it satisfies explicitly stated criteria. Therefore, this approach serves as a much-needed means by which the designer can communicate easily and explicitely with all those who will be affected by his design decisions. If all the elements of the design problem can be precisely articulated, then the designer can directly test the validity of his decisions and elicit suggestions for improvements by consulting the client, the potential users, or the agents who enforce legal, social, or economic constraints. Because the decision-making process is no longer imbedded in an esoteric language everyone is qualified to participate. This is especially important when considering matters of public design policy which have traditionally been conducted as an elitest activity, hidden from the public's eyes and control. Participatory design is much more feasible when the criteria are explicit, the process is visible, and open debate on substantive issues is a viable means of reaching agreement. Only in this way can the process of design respond directly to the legitimate desires of all elements of society.

1. Miller, George A., "The Magical Number Seven," Psychological Review, 63:81, 1956.

Milne, Murray A., The Design Process, Masters Thesis, University of California, Berkeley, September 1965.

2. Simon, Herbert A., Models of Man, New York, Wiley, 1957.

3. Hochschule fur Gestaltung, Quarterly Bulletin, Ulm, April 1959.

4. Alexander, Christopher W.J., Notes on the Synthesis of Form, Harvard University Press, Cambridge, 1964.

5. Alexander explains the reasons behind his new "non-methodological" interests in the Design Methods Group Newsletter, Vol. 5, No. 3, March 1971.

Others who have recently developed new computer based problem structuring algorithms include:

Bernholtz, Allen, and Edward Bierstone, "Computer Augmented Design," Design Quarterly, 66/67, 41-52.

Davis, Charles F., and Michael D. Kennedy, "EPS: A Computer Program for the Evaluation of Problem Structure," Emerging Methods in Environmental Design and Planning, Gary T. Moore, ed., MIT Press, 1970.

Milne, Murray A., "CLUSTR: A Structure-Finding Algorithm," ibid.

Owen, Charles, "DCMPOS: An Algorithm for the Decomposition of Non-directed Graphs," ibid.

6. Alexander, Christopher, ibid.

Milne, Murray A., and Charles W. Rusch, "A Method for Systematic Design in Architecture," draft, School of Architecture and Urban Planning, University of California at Los Angeles, 1969.

7. Rittel, Horst, "The Universe of Design," draft, Institute for Urban and Regional Development, University of California, Berkeley, 1964.

8. Milne, Murray A., "Architectural Applications of Computer Based Network Analysis Models, AIA Architect Researchers Conference, October 1967, unpublished.

Ore, Oystein, Graphs and Their Uses, Random House, New York, 1963.

Simon, Herbert A. The Sciences of the Artificial, M.I.T. Press, 1969 (Chapters 3, 4).

9. Busacker, Robert G., and Thomas L. Saaty, Finite Graphs and Networks, McGraw-Hill, 1965 (p. 77).

10. Ball, Geoffrey H., "Classification Analysis," Office of Naval Research, Contract Number 4918 (00), SRI Number 5533 (lists over 500 methods).

David Wishart, "An Algorithm for Hierarchical Classifications," Biometrics, p. 25, 1969.

11. Alexander, Christopher, "A City is Not a Tree," Architectural Forum, p. 25, 1969.

12. For a discussion of form diagrams or patterns see:

Alexander, Christopher, op cit, 1965.

Alexander, Christopher, Sara Ishikawa, and Murray Silverstein, "A Pattern Language Which Generates Multi-Service Centers," Center for Environmental Structure, Berkeley, California.

13. Milne, Murray, and Charles W. Rusch, "The Cal-Oregon Experiment in Design Education," American Institute of Architects Journal, March 1968.