Expert Systems for Structural Design

by

M. L. Maher

EDRC-12-15-87 >

Expert Systems For Structural Design

Mary Lou Maher, AM ASCE[1]

1. Abstract:

This paper considers the role the computer can have is less formalized phases of the structural design process, particularly during preliminary design. Since preliminary design is based largely on the designers experience, an expert systems approach to developing computer aids is appropriate. Four expert systems are presented: HI-RISE, FLODER, LOCATOR, and STRUPLE.

2. Introduction

Structural design is a process that encompasses all activities between the defintion of a need to resist bads through the construction of the resulting structure. The most common reference to *structural design* includes the activities between the defintion of a structural configuration and a detailed specification of the structure to be constructed. These activities include the transformation of a representation of a physical structure to a mathematical model, the analysis of the model, and the interpretation of the results followed by modifications to the physical representation and/or the mathematical model. These activities are commonly referred to as the design/analysis cycle.

The computer plays an important role in the stnjctural design process by extending the computational facilities of the structural designer so that he may precisely consider complex structural systems. Traditionally, the role of the computer has been restricted to the design/analysis cycle. Software packages have been developed to facilitate the definition of a model of a structural system and the analysis of the. model. This paper considers the role the computer may play in other, less formalized, phases of the structural design process, particularly in the preliminary decision making part of the process.

The paper is organized into two major parts. The first part discusses the nature of preliminary structural design. The second part presents four expert systems, developed at Carnegie Mellon University, that explore and define the decisions made during preliminary design. The expert systems presented are HI-RISE, FLODER, LOCATOR, and STRUPLE. The paper closes with some remarks on the potential for developing computer aids for preliminary structural design.

[1]Assistant Professor,Department of Civil Engineering, Carnegie Mellon University, Pittsburgh PA 15213, (412) 258-3777

3. Preliminary Structural Design

The structural design process starts with the definition of a need to transmit loads in space to a support or foundation, subject to constraints on cost, geometry, and other criteria. The final product of the design process is the detailed specification of a structural configuration capable of transmitting these loads with the appropriate levels of safety and serviceability. The design process may be viewed as a sequence of three stages, as described befow.

1. PRELIMINARY DESIGN (conceptual design) involves the synthesis of potential structural systems satisfying a few key constraints, and the selection of one, or at most a few, systems to be pursued further. Synthesis requires a knowledge of structural subsystems and their appropriateness for different situations.

2. ANALYSES is the process of modeling the selected structural system and determining its response to external effects. This process involves transforming a physical structure to a mathematical model, analyzing the model, and interpreting the results of the analysis in terms of the actual physical structure.

3. DETAILED DESIGN is the selection and proportioning of the structural components such that all applicable constraints are satisfied.

During preliminary design, the form of the design solution is identified. During analysis and detailed design this form is refined. There may be significant deviations between the properties of components assumed at the analysis stage and those determined at the detailed design stage, which would necessitate a reanalysis. Other major and minor cycles of redesign may also occur. The process continues until a satisfactory (or optimal) design is obtained. The *conceptualize-analyze-detail* cycle is typical of many design paradigms.

Many of the conceptual aspects of structural design are embodied in the preliminary design phase. At this point, the only information available to the designer are the requirements of the end product. It is during this part of the design process that the creativity and experience of an engineer are needed. The increasing complexity of structural systems has magnified the importance of the preliminary design solution while, at the same time, made the synthesis of feasible solutions more difficult. Preliminary structural design can become critical to the total design process unless approached in a structured and organized fashion.

In order to find the most suitable structural system under the given conditions designers should be able to think in an overall way, concentrating on the relationship between spatial forms and structural systems and ignoring trivial details. On the other hand, designers should also be able to distinguish the details which must be considered before the properties of the whole system can be well understood. This requires that good designers be knowledgeable about architecture and structures, and also have experiences that enable them to make correct judgments and trade-offs as early as possible to avoid major conflicts later in the design process.

In fact, many major issues *are* settled before preliminary structural design at the schematic level when designers, usually architects working in collaboration with engineers of related fields, generate the spatial forms of buildings based on the considerations of the building function, potential structural systems, and all other design constraints. Unfortunately, as technology advances and complexity of modern building designs increases, architects, structural engineers, and other engineers of different fields become more and more specialized, confining themselves in their own areas. This can cause communication problems among the different designers of buildings at the early stage of the building design, which in turn results in inefficient use of knowledge, time and energy, and potential major conflicts in the following design processes.

Preliminary design is achieved by synthesizing potential alternative designs. There is no standard approach to the synthesis process suitable for all design problems. One approach is to decompose the design problem into the design of independent subsystems. The nature of these subsystems will depend on the nature of the problem at hand. In a similar manner, each subsystem is further decomposed into major components. Alternative design solutions can be synthesized by considering possible combinations of the various subsystems that result from combinations of lower level components. This hierarchical approach to the synthesis of a solution enables the designer to consider a large set of possibilities based on the manner in which the subsystems and the lower level components are defined. These definitions will depend on the nature of the particular problem as well as the engineer performing the design. A particular problem may justify the decomposition of a problem from an abstract level down to a set of detailed subsystems. Another design may be better approached by considering the details first and building up to more general systems. A correct selection of this sequence can increase the efficiency of the synthesis process.

A key consideration in the synthesis of design alternatives is the identification and satisfaction of constraints at the various levels of abstraction. These constraints control the qualification of various components of the design as well as the feasibility of combinations of such components. Typically, these constraints are based on the experience of the designer, but may also include client's specifications and regional restrictions. The formal identification of these constraints is a difficult process that is never completed because as more design experience is acquired, the set of constraints may need to be modified and expanded.

Although designers may not use an exhaustive, organized approach to the synthesis of alternative preliminary designs, they do enumerate the alternatives that appear to be the best of all. This ability to identify feasible designs with partial information has not been formalized; therefore, it is difficult to pass this knowledge to new engineers or to develop intelligent computer assistants to this stage of the design. The use of expert system techniques holds promise for the formalization of preliminary structural design.

4. Expert System Applications to Structural Design

This section describes four expert system applications to structural design developed at Carnegie Mellon University. All four systems address the preliminary stage of design during which decisions on structural system type, material and location are made. The first three systems, HI-RISE, FLODER, and LOCATOR address the synthesis process using a generate and test paradigm. The fourth system, STRUPLE, addresses the pre-synthesis stage during which the most promising structural system types and materials are identified using analogical reasoning.

Although these applications are referred to as expert systems, the developers did not formally interview experts prior to implementation. These expert systems have many of the characteristics of current expert system technology, including separation of control and domain knowledge, use of symbollic reasoning, and certain heuristics gathered from publications and local designers.

4.1. HI-RISE

HI-RISE [5] is a knowledge based expert system that performs the preliminary structural design of high rise buildings. Given the space planning of a building, which is described by a three dimensional grid, HI-RISE generates and presents alternative feasible structural systems. HI-RISE is implemented in PSRL [7], a combination of a production system and frame representation language. The declarative knowledge is represented in the form of frames and rules and the procedural knowledge in the form of Lisp functions.

The motivation for developing a system such as HI-RISE centers around the need for the engineer to identify and evaluate more design alternatives during the configuration stage. Many times the most obvious solution is selected because the engineer does not have the time or the money to develop multiple configurations. A system such as HI-RISE could take advantage of the speed and memory of a computer to quickly generate multiple, feasible configurations. The intention of HI-RISE is to provide an intelligent design assistant to suggest alternatives, not to automate the preliminary design process.

The input to HI-RISE is a three dimensional grid, specifying potential locations for beams, columns, and walls. The grid represents the result of space planning and the architectural restrictions. The topology of the grid is defined by the number of stories and the number of bays in each direction. The geometry is defined by the dimensions of the bays and the minimum required clearance for a typical story. Other spatial constraints, such as the location of service shafts or internal spaces, are specified in terms of their location on the input grid. Other input information required by HI-RISE is the intended occupancy of the building, and the wind and live load.

HI-RISE is currently restricted to buildings with a rectangular perimeter and a simple rectangular structural grid. The knowledge incorporated in HI-RISE is appropriate for buildings between 5 and 50 stories. These restrictions were imposed to allow a study of the feasibility of developing a computer aid for preliminary design. Many of these restrictions have been relaxed in more recent attempts to develop intelligent computer aids to preliminary structural design.

In HI-RISE, the preliminary design process is divided into two major tasks; each task addresses the design of a functional system. The functional systems are designed in a fixed order: first the lateral load resisting system, followed by the gravity load resisting system. Each of the two major tasks are deomposed into a set of similar subtasks. The subtasks have the same goals for each functional system, however, the details of reaching these goals differ. The subtasks are described below.

- *Synthesis.* The first task is to synthesize alternatives for the functional system under consideration. The synthesis is performed as a depth first search through the appropriate generic subsystems stored in the knowledge base. Each subsystem has a set of synthesis levels with itemized alternatives at each level. For example, the levels and alternatives for the lateral load resisting systems are: 3D-lateral, core or 2D orthogonal; 2D-lateral, rigid frame, braced frame, or shear wall; material, steel or reinforced concrete. Synthesis is implemented as sets of rules that represent the decisions to select an alternative at a given level, move to a lower level, move to a higher level, and eliminate an infeasible design alternative.

- *Analysis.* The purpose of the analysis subtask is to determine the feasibility of an alternative and to define its component groups. Feasibility is determined by the formulation and evaluation of one or more feasibility constraints. An approximate analysis provides one set of ingredients, namely, the required load capacity of the system components. Component groups are defined so that preliminary sizing or proportioning is performed only for one component in a group.

- *Evaluation.* Evaluation of a structural design may be based on many diverse features of the design. Evaluation is usually done by designers in an abstract form and is usually based on cost. Some of the other features considered are esthetics, efficiency, and structural integrity. HI-RISE evaluates alternatives with a linear evaluation function. There is a distinct function for each of the functional systems. The variables in the function are features of the system that may be quantified, e.g. drift.

- *System Selection.* HI-RISE presents all structurally feasible systems to the user, indicating which system has been determined to be the "best", selected as the system with the minimum value assigned by the evaluation function. The user may either accept the recommended design or override the decision of HS-RISE and choose one of the other structurally feasible systems.

Structural systems are represented as hierarchically defined frames that are used as templates for generating alternative configurations. A frame is an object with declarative and procedural attributes. The frames can be considered in three catergories: global, functional, and physical. The global frames, shown below, are the *building* and *grid* frames. These frames store information global to the entire building. The building frame stores information about the occupancy and the design loads. The grid frame stores information about the structural grid, such as the number of stories, the number of bays in each direction, and their dimensions. There are procedural attachments to some of the grid attributes to automatically store information about ratios and total dimensions when the appropriate informatbn is provided to infer these values.

```
{  building
              occupancy
              wind-load
              live-load }

{  grid
              part-of building
              stories
              story-dim
              min-clear
              narrow-bays
              narrow-dim
              wide-bays
              wide-dim
              mech-floor
              shaft
              shaft-sym ... }
```

The functional frames are the *lateral* and *gravity* frames, serving to divide the structural systems according to function. Only those structural systems responsible for resisting lateral loads are linked to the lateral frame, and gravity systems to the gravity frame. These frames contain information about function and status of alternative configurations.

```
{  lateral
              part-of grid
              best-lat  ... }

{  gravity
              part-of  grid
              uses   lateral
              best-grav   ... }
```

The frames that represent the physical level are hierarchically defined according to function. The *3D-lateral% 2D-lateral,* and *material* frames represent decisions made for the configuration of the lateral load resisting system. The *20-horizontal, support-edges,* and *support-div* frames represent decisions made for the configuration of the gravity, or floor, system. There are additional frames in the physical level that represent the information associated with components such as beams, columns, and diagonals.

```
{  3D-lateral
              is-alt  lateral
              3D-descrip'tion }
```

```
{   2D-lateral
            is-alt   3D-lateral
            paxrt-of
            uses
            direction
            2D-description }

{   material
            is-alt   2D-lateral
            mat-description
            dead-load-est   125
            story-dim-est   10.0}

{   2D-horizontal
            is-alt   gravity
            hor-description }

{   support-edges
            is-alt   2D-horizontal
            sup-edges }

{   support-div
            is-alt   support-edges
            subdivide-direction ... }
```

The frames are used to organize the representation of the alternative design solutions. An example of preliminary design solutions generated by HI-RISE is shown in Figure 4-1. The boxes in the Figure repesent frames, and the lines represent relationships between frames. An *is-alt* relationship implies that the connected frames are alternative solutions, a *part-of* relationship implies that all the connected frames are part of one solution, and the *uses* relationship is defined so that information may be inherited horizontally in the tree of solutions.

4.2. FLODER

FLODER [4] is a knowledge based expert system that designs alternative floor systems and framings for a given architectural floor plan. The architectural floor plan can be specified as a set of rectangles representing the building perimeter, hallways, and service shafts. FLODER generates a column grid and proportions the floor system and framing system for gravity loads. FLODER addresses the preliminary design of floor systenns when the column tocations are not fixed and can be specified for maximum efficiency of the structural system.

FLODER requires three different kinds of input from the user: architectural data, structural data, and evaluation data. The architectural data includes geomtetric data and functional data. The user is first required to specify the floor plan in 'rterms of its geometry, i.e. the size and location of rectangles, and function, i.e. hallways and service shafts. The user is then prompted for the type of structural material to

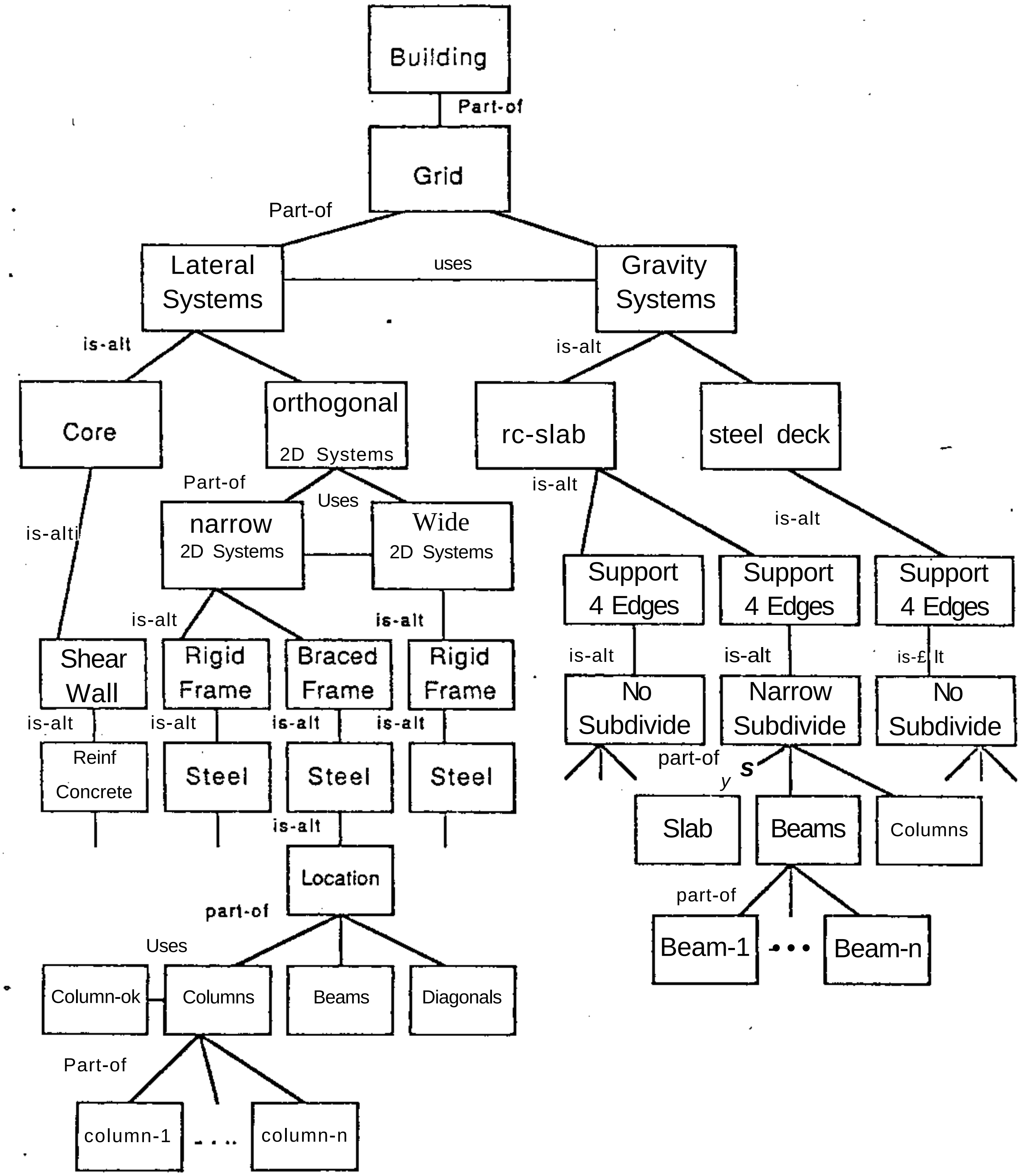

Figure 4-1: Example of Design Solutions in HI-RISE

be used when generating the floor system. The user makes a distinct choice for the 2D horizontal system, e.g. precast panels, and the 1D components, e.g. reinforced concrete. Finally, the *user* is prompted for weighting factors for the properties that are evaluated. The user may generate multiple design solutions by changing the choice of materials for the same architectural data.

The output of FLODER is a graphical representation of the alternative solutions. The graphical representation is provided incrementally; the plan perimeter is illustrated upon input, followed by the column locations, girders (if any), and beams (if any) as they are generated. The first generated floor system is displayed on the whole screen. Subsequent alternatives are displayed on one fourth of the screen and the first alternative is redisplayed.

An example of the graphical presentation of alternative framing plans is shown in Figure 4-2. Each quadrant of the figure illustrates one alternative. The distinction between alternatives is defined by the user when selecting the type of materials to be used. The upper left framing plan was generated using knowledge of reinforced concrete beams, columns and slabs. The upper right framing plan was generated using knowledge of prestressed concrete beams, columns and slabs. The lower left was generated using knowledge of steel beams, columns and steel deck floor systems. The lower right was generated using knowledge of reinforced columns and flat slabs (implies no beams used for support, only columns).

FLODER uses a generate and test method to synthesize a floor plan. The generate stage involves placing the columns, girders, and beams on the floor plan. The test stage involves analyzing portions of the floor system and evaluating the system for constraint satisfaction. These tasks are further described below.

The framing generation proceeds from the placement of the one dimensional elements to the three dimensional configuration. This configuration consists of columns and beams defining a three dimensional frame supporting the two dimensional surface elements. The generation depends on the structural and architectural constraints, since the distance between sequential columns should not exceed the limit for the material type, and no structural elements are allowed in the restricted areas defined by the architectural constraints.

In the analysis subtask, the structural elements to be analyzed are selected as those requiring the biggest and smallest amount of material to meet structural requirements. The analysis module consists of two parts. The first is performed for the comparison of required depths or deflections between elements in the same alternative, while the second part is performed for a comparison of the total amount of material

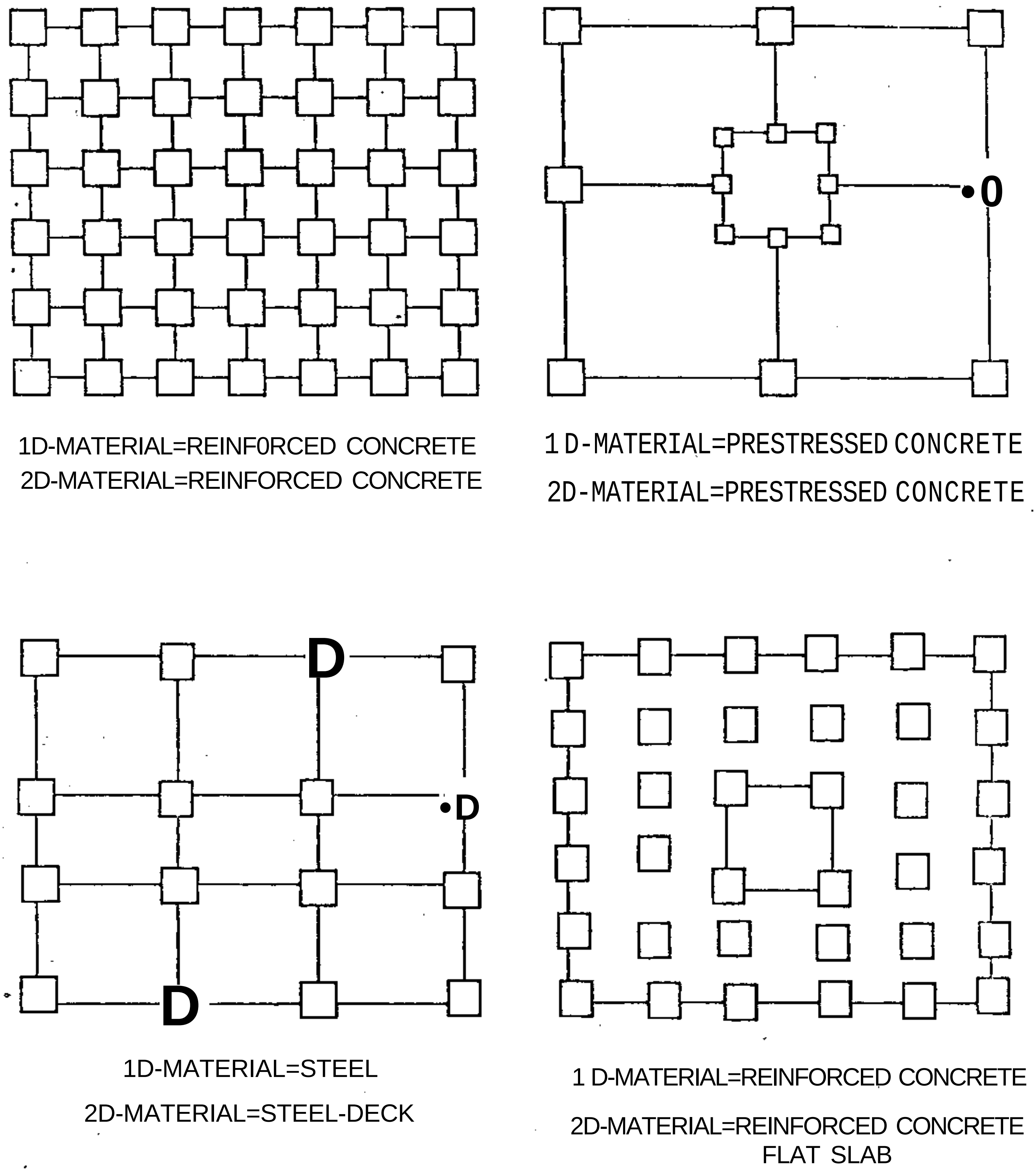

Figure 4-2: Presentation of Framing Plans in FLODER

between different alternatives.

The evaluation subtask also consists of two parts; the evaluation of different structural elements within the same framing, and evaluation of alternative framings for the given architectural plan. The first evaluation may result in revisions to a single floor framing, while the second provides information on the relative merits of alternative floor plans.

The knowledge base of FLODER is made up from a combination of OPS5 production rules and Lisp functions. The control and the heuristics are represented by OPS5 rules. Purely algorithmic sections, such as analysis of slabs, are programmed in Lisp which is much easier and faster for manipulating numerical *6c\.?* than OPS5. The user interface module is also coded in Lisp.

4.3. LOCATOR

LOCATOR [8] is an expert system to serve as an assistant in locating lateral load resisting systems on a three dimensional building grid. LOCATOR generates feasible alternatives for the placement of a lateral system configuration specified by the user. The alternatives are tested by checking the satisfaction of heuristic constraints.

The design of the lateral load resisting system is considered in two parts: configuration and layout. Configuration is the process of generating structural systems that can be described in all aspects except their placement on a building grid. In the context of LOCATOR, three choices are made during configuration:

- whether to design a 3D, 2D orthogonal, or combination of a 3D and 2D orthogonal system,
- the structural subsystems to be used, e.g. rigid frames, and
- the structural material to be used.

Layout of a lateral system usually follows its configuration and is concerned with the topology of the design. Once a system is configured, the design must satisfy the spatial or architectural constraints imposed on it. In the context of building design, a building grid is used to represent potential column locations, and layout involves placing the lateral system on this grid.

Upon entering LOCATOR, all configuration decisions are made and specified to the expert system. Locator generates and tests alternative placement schemes for the given configuration and presents them to the'user. The generate and test method is applied to levels of placement decisions: 3D layout, 2D layout in the narrow direction, and 2D layout in the wide direction. At each level, an alternative is selected and tested for feasibility. The alternatives for 3D layout are predetermined according to the type of 3D structural system, for example, the placement of a tube system is on the perimeter of the building. The

-alternatives for 2D layout are synthesized by combining alternatives for *column lines used and bays used,* Figure 4-3 illustrates some possible combinations for 2D layout.

LOCATOR was developed in Knowledge Craft [2], an environment for building expert systems. LOCATOR makes use of CRL, an object oriented representation language. The design knofwedge is represented in schemata (or frames) and rulesets. Schemata are used to form templates for alternative placement schemes and to explicitly represent decision levels. Rulesets, coded in Lisp, are sets of conditional statements that test intermediate solutions for feasibility.

4.4. STRUPLE

STRUPLE [6, 9] is an expert system for structural configuration planning. STRUPLE uses a database to store structural design solutions collected from several structural consulting firms and construction companies. The major capacity of STRUPLE is: given the description of a building, mainly the architectural information and some structural information such as the loads and the construction material that the designer prefers to use, STRUPLE will find relevant structural solutions in the database and use the information in the old solutions to plan the structural configuration of the new building.

Experiences used by STRUPLE are represented in the form of incomplete descriptions of building design solutions, which gives the following information about buildings.

- General information. This includes building number, name, design completion date, location, owner, general use of the building, and unit and total costs.
- Geometric information. This includes gross area, height, width, aspect ratio, overall shape and shape irregularity which is a real number roughly indicating how irregular the shape is.
- Other architectural specifications. This includes a number of stories below and above grade, intended use, bay numbers and sizes in two main orthogonal directions, floor-to-floor height, ceiling-to-ceiling height, number of service shafts and general description of their location in the build plan.
- Load information. This includes dead load, live load, wind load and seismic zone.
- Primary 3D systems. A building may use one than one type of 3D system, for example, one or more cores with a 2D orthogonal system, which may be composed of rigid frames, tied together.
- Lateral 2D systems. Lateral 2D systems are the subsystems of 3D- systems. The information about lateral 2D systems includes the structural type of 2D systems, the construction materials, and directions in which they are placed.
- Floor systems. This includes types of floor systems, which imply the construction material type, types of support systems, for example, columns in the case that floor system is flat slab, and construction materials of the support systems.
- Foundation systems. This includes types and bearing capacities of the foundation systems. Again, types also indicate the construction material used for the foundation systems.

The relations in the database are defined using the same organization. This description only includes

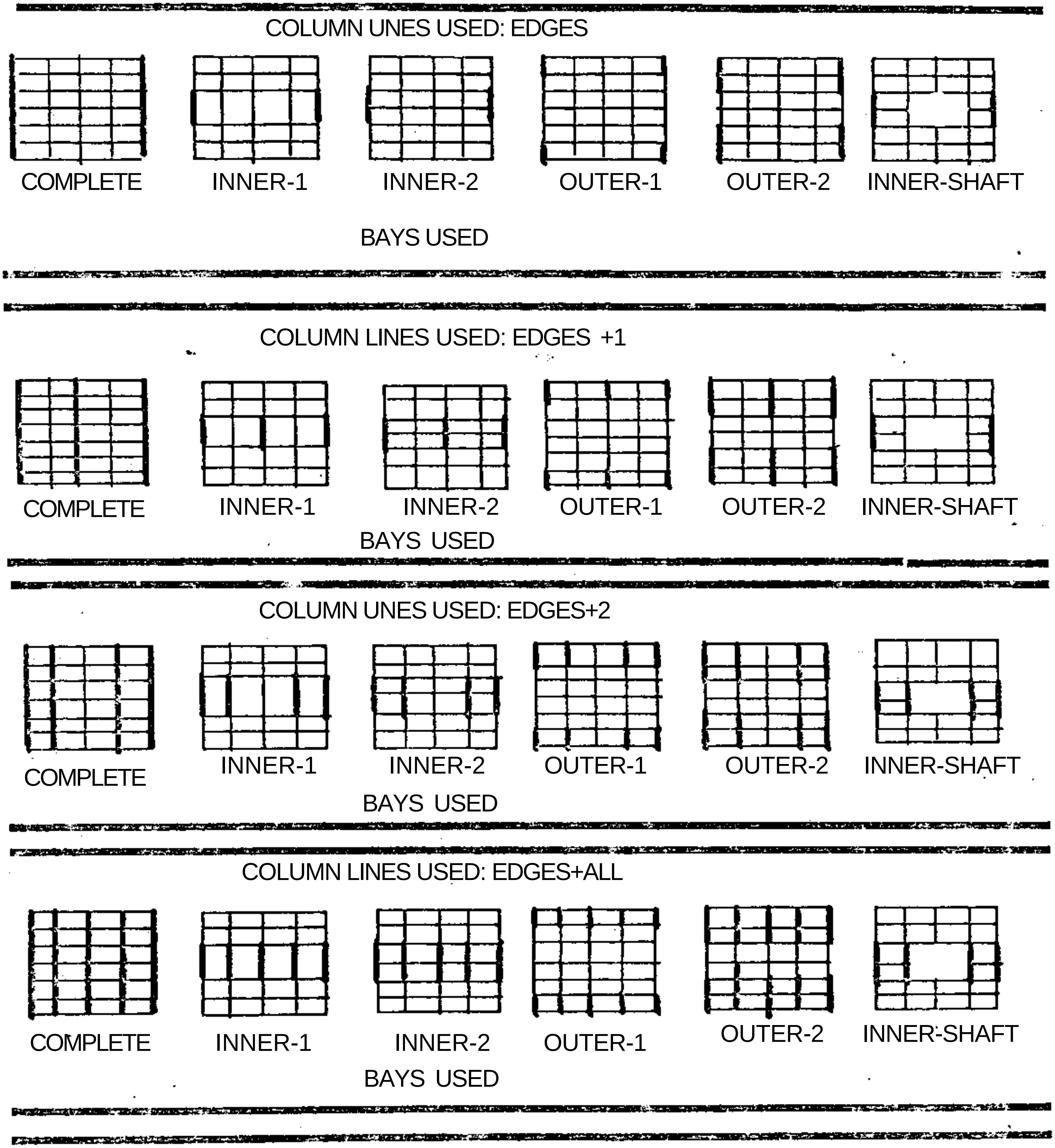

Figure 4-3: 2D Layout Alternatives in LOCATOR

the general characteristics of buildings. Some information that is important, as well as more detailed information, is missing because either it is difficult to represent or is not important to this study.

The first task of STRUPLE is to find similar experiences to the current design problem. The input is a partial description of the architectural plan and some structural information of a building to be designed. STRUPLE generates a list of similar buildings retrieved from the database. The evaluations of the buildings and their rankings based on their similarities to the input building are also included.

The buildings which are considered to be similar to the given building are called *matching buildings,* or *matches.* To find matches requires a similarity metric to determine which buildings in the database are similar to the input building. The similarity metric is described by a set of *criteria,* also called *matching criteria,* which define what significant common aspects the matching buildings and the input building should share. A matching criterion is a requirement of similarity imposed on a *feature* of a matching building. For example, the number of stories, the intended use, the design wind toad, the construction material, etc. all are features of a building and are potential criteria. Obviously, not all features of a building are equally important in terms of their impacts on the structural system. Thus, criteria are divided into three classes: required criteria, desired criteria and no-match criteria, where required criteria must be satisfied by a matching building, the satisfaction of desired criteria is desirable but not necessary and no-match criteria are not considered. *Required, desired* and *no-match* are qualifiers that represent the *status* of a criterion.

While searching the database for similar buildings, required criteria are used as qualifiers. In order to locate a matching building explicitly, we need to know the limits within which the building can be considered to be a match. Thus a *range* must be set for each criterion. For a required criterion, the range is defined by the limits of allowable difference of the feature values of the two buildings compared. To measure how well the matching buildings resemble the given building, each match is evaluated and ranked. A *single evaluation* is a measure of the difference between the values of one feature of a match and the given building. The *final evaluation* is a comprehensive measure combining all individual evaluations for a match.

Once matching buildings have been found which represent the experience that may be relevant to the new design problem, we have to determine what knowledge is to be transformed and used to solve the new problem. It is a fact that even though two buildings could have many common aspects, because of their different special design requirements, the design solutions may not be the same. However, it is also a fact that a certain group of design vocabulary frequently appears in certain types of buildings. Although we can not directly use the design solution of an existing building for a new building, we can use the design vocabulary of the old design to construct the new solution. Thus we choose the design vocabulary as the knowledge to be extracted from the old design solution and be transformed.

The database may have a vast amount of design solutions and many of them may be found to be similar to the new building. In order to use the information efficiently, the frequency of each element of design vocabulary used in the old designs is calculated. According to-their frequencies of appearance and evaluations based on their similarity to the new building, each type of element is assigned a priority of consideration. A very high priority indicates that this type of element is frequently used in the buildings of the type of the new one and should be considered first.

STRUPLE has a set of structural elements that are stored in its knowledge base representing the complete design vocabulary that the synthesis process knows about. The structural elements are organized into different abstraction levels. For each level of abstraction, STRUPLE determines and orders the design vocabulary to be considered for the new building, thus identifying a subset of the complete vocabulary of structural elements which is most promising. This subset of design vocabulary will be used during the synthesis process, in which each structural element will be examined to determine whether it is an eligible or efficient alternative.

STRUPLE is implemented in OPS83 [3], a language developed for expert system applications at Carnegie Mellon University. OPS83 "combines the rule-based programming paradigm of the earlier versions of OPS with the procedural programming paradigm of conventional programming languages" [1]. In OPS33, in addition to working memory elements, other data types and data structures present in Pascal are available. The basic program units are production rules and procedures. Users can determine and implement their own inference mechanism and conflict resolution strategies by writing necessary procedures with the help of some working memory retrieval functions buitt in OPS83. These features of OPS83 makes it more flexible and easier to build the user interface. The database used in STRUPLE is implemented by Ingres, a relational database management system. The communication between the database and the expert system is set up through a C procedure call which is quite straight forward.

5. Summary and Conclusions

Four expert systems for preliminary structural design were presented. HI-RISE was the first system developed and serves as the basis for current research. HI-RISE identifies a fomnalization for preliminary structural design as a synthesize, analyze, and evaluate paradigm. HI-RISE illustrates that although computer aids can be developed for preliminary design, much work needs to be done before such aids can be practical. FLODER is an expert, system that addresses the decisions made in generatirg a structural grid for a building. LOCATOR proposes a methodology for generating alternative placement schemes for a lateral toad resisting system on a given grid. STRUPLE serves as a preprocessor to the synthesis part of the preliminary design by using past experience to identify the structural subsystems and materials relevant to the current design.

In conclusion, the use of expert system techniques holds much promise for structural designers. Through the development of expert systems for preliminary design, designer's are asked to identify their own design process. This identification serves to improve the designer's and the engineering community's understanding of the design process. The use of expert systems can improve the quality of the preliminary design process .by bringing more experience to bear on the current problem than that of a single designer.

References

[1] Forgy, C.L
The OPS83 User's Manual and Report.
Technical Report CMU-CS-84-133, Department of Computer Science, Carnegie Mellon University, May, 1984.

[2] ~ .
Knowledge Craft User Manual
Carnegie Group Inc., Pittsburgh PA, 1986.

[3] Forgy, C. L.
OPS5 User's Manual.
Technical Report CMU-CS-81-135, Carnegie-Mellon University, July, 1981.

[4] Karakatsanis, Andreas.
FLOULIR: A Floor Designer Expert System.
Master's thesis, Department of Civil Engineering, Carnegie-Mellon University, 1985.

[5] Maher, Mary Lou.
HI-RISE: A Knowledge-Based Expert System For The Preliminary Structural Design Of High Rise Buildings.
PhD thesis, Department of Civil Engineering, Carnegie-Mellon University, 1984.

[6] Maher, M.L and Zhao, F.
Using Experience To Plan The Synthesis of New Designs.
Expert Systems in Computer Aided Design , 1987.

[7] Rychener, M. D.
PSRL: An SRL-Based Production-Rule System.
Reference Manual.
1984

[8] Smith, Douglas F.
LOCATOR A Knowledge Based Lateral System Locator For High Rise Buildings.
Master's thesis, Civil Engineering, Carnegie Mellon University, November, 1986.

[91 Zhao, Fang.
Application of Al Techniques to Planning Structural System Configurations.
Master's thesis, Department of Civil Engineering, Carnegie Mellon University, March, 1987.

Table of Contents

List of Figures