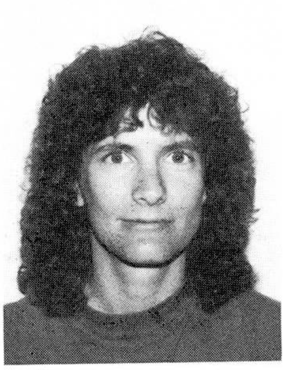
Synthesis of structural systems

Synthesis of Structural Systems

Synthèse du projet de construction

Synthese des Bauentwurfs

Mary Lou MAHER
Associate Professor
Carnegie Mellon University
Pittsburgh PA, USA



Mary Lou Maher received a BS from Columbia University in 1979 and a MS and PhD from Carnegie Mellon University in 1981 and 1984 respectively. Her teaching and research interests include computer aided design and AI applications to structural design.

SUMMARY
This paper discusses an expert system approach to the synthesis phase of structural design. In addition to formalizing the design knowledge for designing structural systems, a synthesis algorithm is developed. The use of such an algorithm in developing expert systems for structural design facilitates the development of a knowledge-base. This approach is illustrated with applications to the design of structural systems for buildings.

RESUME
Cet article expose une des approches par système expert de la phase de synthèse dans la conception d'une ossature. En plus de la formulation de la base de connaissance pour la conception des ossatures, un algorithme de synthèse est développé. L'utilisation d'un tel algorithme dans le développement des systèmes experts pour la conception d'ossatures facilite le développement d'une base de connaissances. Cette approche est illustrée par des applications concernant la conception d'ossatures de bâtiments.

ZUSAMMENFASSUNG
Die Arbeit diskutiert eine Möglichkeit eines Expertensystems für die Synthesephase des Gebäudeentwurfs Zusätzlich zur Formalisierung des Wissens für die Konstruktion von Gebäudesystemen wird ein Synthese-Algorithmus entwickelt. Die Verwendung eines solchen Algorithmus erleichtert die Schaffung einer Wissensbasis bei der Entwicklung von Expertensystemen für die Konstruktion. Dieser Weg wird anhand von Anwendungen für Hochbauten dargestellt.

1. INTRODUCTION

Structural design includes the synthesis of a structural system that satisfies a set of requirements. Synthesis can be considered at several levels of abstraction, where more information about the requirements as well as the evolving design description is available as the process proceeds. In this paper, the focus is on the early stages of design where the design knowledge is largely qualitative. During the early stages, or preliminary design, the major components or subsystems are identified and their composition is evaluated. Although the identification and composition may make use of associated quantitative models, the designer typically reasons about these models in a qualitative manner.

To support the designer in the identification and composition of components of structural systems requires both synthesis and evaluation methods. Such methods can provide a systematic approach to design, allowing the designer to pursue more alternatives and to evaluate the alternatives based on a discourse of criteria and value. The use of an expert systems approach for the exploration of alternative structural systems maintains a separation of method and knowledge, allowing the designer to guide the methods with qualitative or empirical knowledge without sacrificing the benefit of a systematic approach.

In this work, the synthesis of structural systems is based on a constraint directed search through a design space that is decomposed into components, subsystems and constraints. Evaluation of alternative structural systems is based on the concept of Pareto optimality, where multi objective optimization provides a basis for identifying a set of optimal solutions among a set of feasible solutions. Both synthesis and evaluation are integrated in a single model for producing alternative design descriptions for a given set of requirements. This model has been implemented as an environment for developing expert systems, where the experienced designer defines a knowledge base and the designer uses the resulting knowledge base to produce design solutions.

2. SYNTHESIS

There are many books that provide definitions and elaborations of the design process; in structural engineering such books include [3], [4], [2] and [1]. The design process can be considered as comprising different phases, synthesis being one of these phases. Although the phases may not be addressed hierarchically for the entire design process and are often carried out recursively, there is an inherent order in which designers approach a design problem. The following represents one formalism of the design process.

- *Formulation* involves identifying the goals, requirements and possibly the vocabulary relevant to the needs or intentions of the designer.
- *Synthesis* involves the identification of one or more design solutions within the design space elaborated during formulation.
- *Evaluation* involves interpreting a partially or completely specified design description for conformance with goals and/or expected performances. This phase of the design process often includes engineering analysis.

Formulation occurs at some level of abstraction and provides enough information to begin a synthesis process. The result of formulation is usually a set of design specifications. For example, the design of a 30 story office building with a regular 25 foot grid represents a partial set of specifications. Synthesis involves identifying the form of the design solution. For the office building, the result of synthesis may be a set of steel rigid frames along the grid lines with a reinforced concrete floor slab. Evaluation, during the early stages of design, is usually based on a subjective assessment of relevant criteria. For example, alternative structural configurations may be evaluated according to cost estimates, ease of construction, and stress-strain requirements. The knowledge used during synthesis and evaluation of preliminary designs is not well articulated. Experienced designers resort to trial and error less frequently than novice designers when searching for an appropriate or satisfactory form, suggesting that the use of an expert system to represent 'experience' may improve design synthesis and evaluation. The problem is how to represent this experience.

During synthesis a designer considers a design space which contains the knowledge that is used to develop the design solution. For structural design, a design space may include different types of framing systems, floor systems, wall systems, and materials. A human designer does not explicitly identify his design space, it is implicitly developed and expanded as he gains experience. A design program, however, does contain an explicit representation of the relevant design space. The formalization of the knowledge in the design space is of interest when considering an expert system approach to structural design.

3. EDESYN

EDESYN (Engineering DEsign SYNthesis) is a software environment for developing expert systems for design. The development of EDESYN was modelled after the expert system "shell" concept. An approach to developing an expert system for structural design was implemented as HI-RISE [5]. This approach was generalized and expanded to facilitate the development of expert systems for design. The design method is implemented as an algorithm to serve as an inference mechanism. The design knowledge is structured to provide a formalism for developing a knowledge-base.

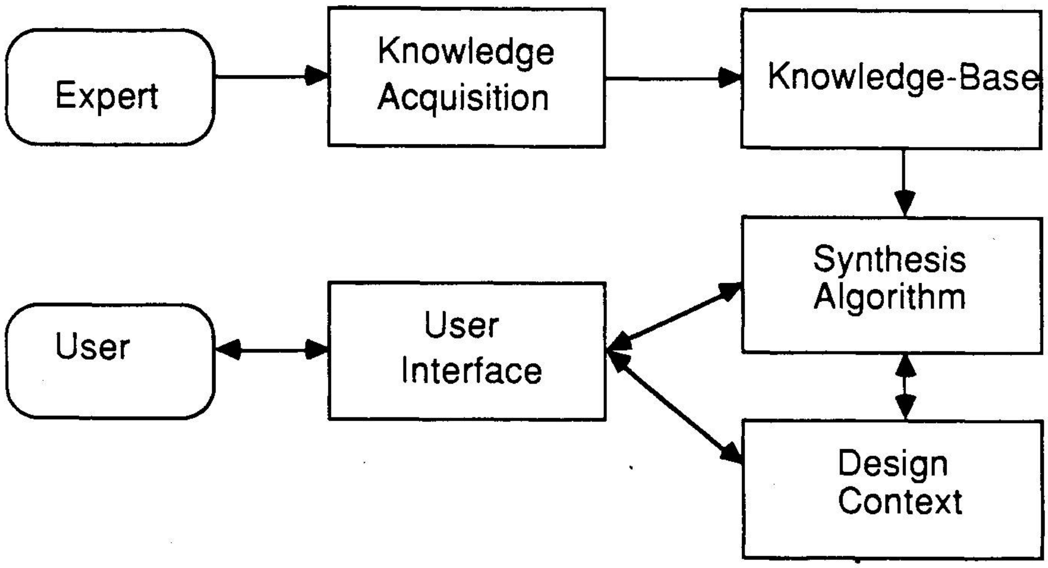


Figure 1: Architecture of EDESYN

EDESYN consists of five main modules: design knowledge-base, synthesis algorithm, design context, user interface, and knowledge acquisition facility, as illustrated in Figure 1. When using EDESYN, the knowledge acquisition facility is invoked first. During knowledge acquisition, the domain specific knowledge is read from files prepared by a domain expert. The domain specific knowledge is stored in the knowledge-base and the synthesis processor is invoked. The user then provides the problem specific information through the user interface to initialize the design context and guides the synthesis of design solutions to augment the context.

The knowledge-base includes decomposition, planning, constraint, and evaluation knowledge. The decomposition knowledge is specified as systems and subsystems, where each system is comprises a set of attributes. An attribute may be another system (i.e. subsystem), representing a synthesis node in a goal tree, or a simple attribute, representing a terminal node. The synthesis node is specified by another system. The terminal node is specified as a selection from a set of discrete alternatives or the evaluation of a Lisp function. The planning knowledge is associated with the system to identify the relevant attributes for the current design situation and the order in which the attributes should be considered.

An example of a system definition for designing the lateral load resisting system for a building is:

```
(system lateral
        3D-lateral one-of (core orthogonal-2D)
        2D-lateral subsystem 2D-lateral
planning
        If stories < 5 Then 2D-lateral
end system)
```

The design of a lateral load resisting system is described by the 3D lateral system and the 2D lateral system. The 3D lateral system can be selected from a set of alternatives and the 2D lateral system must be synthesized. The planning rule indicates that buildings with less than 5 stories should only have one attribute, i.e. the 3D lateral system is not appropriate.

The constraints are specified in the knowledge base as elimination constraints, where each constraint is a combination of design decisions and design context that is not feasible. The constraints are used during the synthesis process to eliminate infeasible alternatives. Examples of constraints in the structural design knowledge base are:

```
IF
stories > 30
3D-lateral = orthogonal-2D
THEN not feasible

IF
2D-lateral-x/material = steel
2D-lateral-y/material = concrete
THEN not feasible
```

The first constraint eliminates a 2D-orthogonal lateral system for buildings with more than 30 stories. The second constraint ensures that a concrete system is not built in the y direction if the lateral system in the x direction is defined to be steel.

The evaluation knowledge is specified by a set of criteria for each synthesis node or system. A criterion is described by a label, a weighting factor, a non-dimensionalizing factor, a normalization factor, and a function to determine the value of the criterion for a design solution. Example criterion for the lateral system are stiffness, compatibility, cost, and ease of construction. The value for each criterion is assessed using qualitative knowledge about structural systems since there is not enough information during preliminary design for a quantitative analysis. For example, stiffness could be assessed in a relative manner, where the designer knows that in most cases a braced frame structure is stiffer than a rigid frame structure.

The synthesis algorithm uses the design knowledge in the knowledge base to produce feasible design solutions consistent with the context. The overall algorithm is based on a constraint directed depth first search through the systems in the knowledge-base. The attributes of each system are assigned all legal values, where a legal value is one that does not get eliminated by the constraints. All feasible combinations are generated for each system, using the planning rules to define and order the attributes. After the alternatives for a system have been synthesized, the evaluation mechanism is invoked. The alternatives are compared for each criteria to produce a set of non-dominated solutions, which are then ranked using the preferences specified by the weighting factors. At this point, the solutions are presented to the designer along with the evaluation information and the designer chooses one solution for further consideration.

The design context initially contains the requirements and specifications associated with a particular design problem. For example, the intial context for a structural design problem includes the number of stories in the building, the occupancy, the structural grid, etc. The context expands as synthesis proceeds to include a tree of alternative solutions, where each node in the tree represents a solution for an attribute of a system. Along with the solution tree, a hierarchy tree is maintained to associate each attribute in the solution tree with the system for which it was generated.

The user interface is implemented using a multi-window, menu driven interaction style. During the design synthesis process, the user can view and change the design specifications, monitor the synthesis process as a tree of solutions is generated, and view a single solution in more detail.

The knowledge acquisition facility transforms the information provided by the expert to the frame based representation of the knowledge base. The expert provides the following design knowledge: preconditions, decomposition, constraints, evaluation criteria, and functions. The design knowledge is specified in a simple syntax and stored in files. Preconditions are specified as a set of names, default values, and allowable ranges. For example, one precondition may be wind load and its default value 30 psf, and its allowable range > 0.0. Decomposition knowledge includes the systems, subsystems, attributes, and planning rules. The constraints are specified as infeasible combinations of elements. Each evaluation criterion is sepcified by a name and a procedure for assigning a value using the goals and elements associated with the current solution. Functions are specified as Lisp functions that use the current state of the design solution to calculate the value of a parameter.

4. STRYPES AND STANLAY

EDESYN has been used to develop two expert systems for structural design: STRYPES and STANLAY. STRYPES generates alternative combinations of structural systems and materials for a given building. STANLAY accepts a feasible combination of structural systems and materials for a given building and generates alternative layouts and approximates the load requirements for the structural components. The knowledge bases for each of these expert systems is described below.

The knowledge-base for STRYPES is described by the decomposition knowledge and the constraints for recomposition. The decomposition knowledge is illustrated in Figure 2. The generation of alternative structural system types and materials is decomposed into the lateral and gravity load resisting systems. For the lateral system, a selection of alternative 3D systems and 2D systems in each direction are combined. The 3D systems are selected from 2D orthogonal systems and a 3D core system.The 2D systems are selected from rigid frames, braced frames and shear walls. For the gravity system, a selection of alternative 2D-horizontal systems and support conditions are combined. For example, a possible gravity system is a reinforced concrete slab supported on 4 edges without intermediate floor beams. Another possible system is a steel deck supported on two edges with intermediate floor beams.

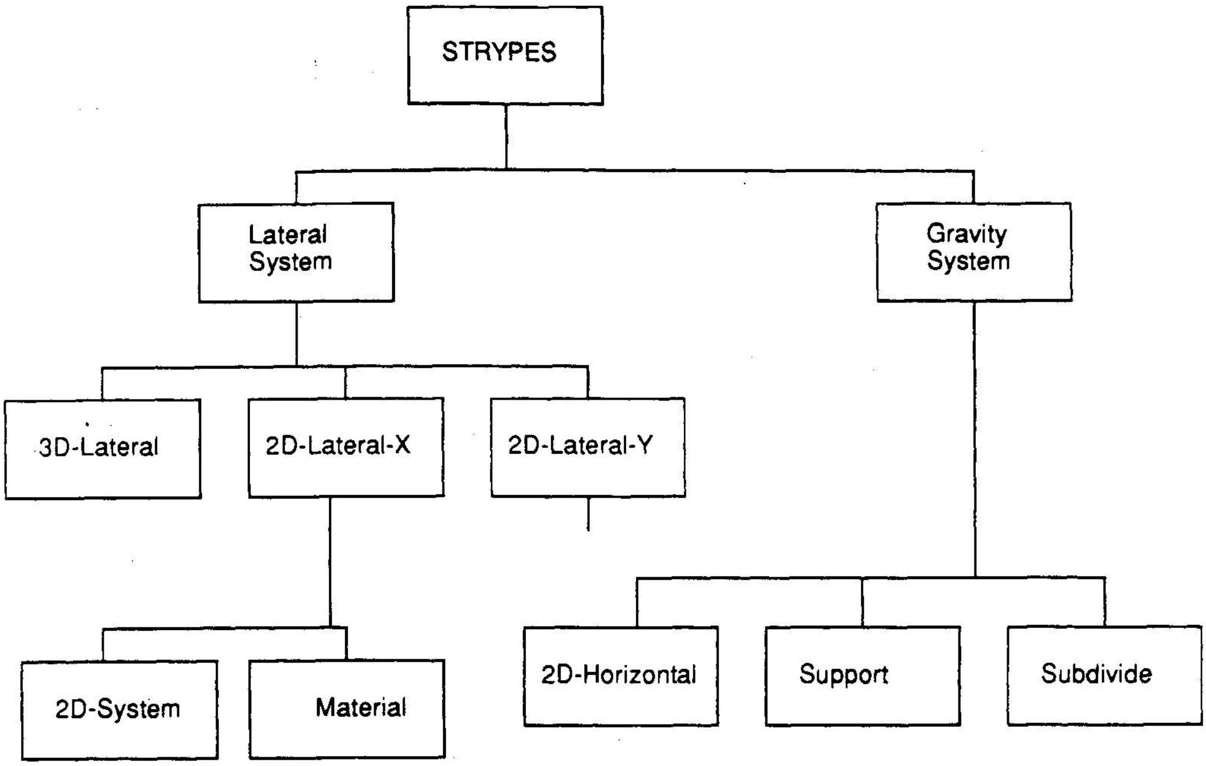


Figure 2: STRYPES Decomposition Knowledge

```
Gravity-System

2D-Horizontal   one-of   ( concrete steel-deck
                                  panels waffle )
Support   one-of   ( 0-edges 2-X-edges 2-Y-edges
                                  4-edges )
SubDivide   one-of   ( none X-direction Y-direction )
```

Figure 3: Gravity System in STRYPES

An example of a system definition in STRYPES is illustrated in Figure 3. The system represents the Gravity-System node in the decomposition tree. The alternative gravity systems are determined by combining selections from different 2D horizontal types and the number of edges supported and the decision to subdivide in one direction. The alternatives formed depend on the constraints and the design context. The use of a particular 2D horizontal type may depend on the lateral system and on the span of the structural grid. These constraints are generalized and stored in the knowledge-base.

The constraints on recomposition in STRYPES eliminate infeasible alternatives to reduce the number of solutions considered. Some constraints are based on design heuristics, eliminating alternatives that an experienced engineer would not consider. For example:

IF
lateral-system/3D-lateral = orthogoanl-2D
2D-lateral-system/2D-system = shear-wall
stories > 35
THEN not feasible.

This constraint eliminates the use of 2D shear wall systems for buildings with more than 35 stories. Other constraints eliminate unusual combinations of materials and systems. For example:

IF
2D-lateral-system/2D-system = shear-wall
2D-lateral-system/Material = steel.
THEN not feasible.

This constraint eliminates shear walls made entirely of steel.

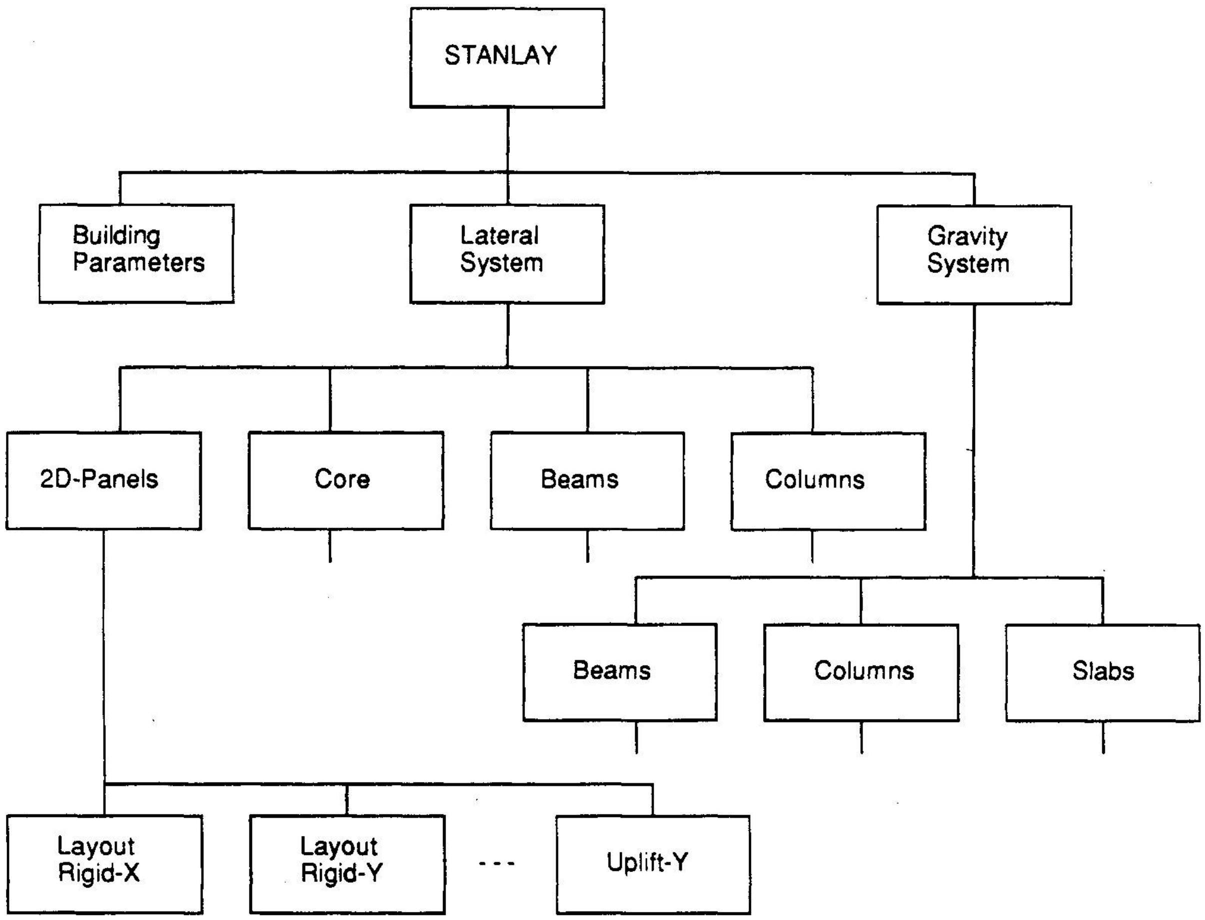


Figure 4: STANLAY Decomposition Knowledge

The decomposition knowledge in STANLAY is illustrated in Figure 4. The layout and load distribution is decomposed into three major decision groups: building parameters, lateral system, and gravity system. The building parameters system calculates and infers additional information about the building given the input conditions. The lateral system is considered by system and component type. The 2D-Panels system places the appropriate systems on the structural grid and distributes the lateral load to each panel. The 2D-Panels system generates alternative placement schemes. The core system locates the walls around the service shaft and determines the lateral load acting on the core. The beams and columns systems distribute the loads to each of the components using approximate analysis techniques. The gravity system, similar to the lateral system, distributes the gravity loads to the components using approximations.

```
2D-Panels

layout-rigid-X  one-of  (edges edges+1  . . . )
layout-rigid-Y  one-of  (edges edges+1  . . . )
- - -
Mover-X  function . . .
Mover-Y  function . . .
Uplift-X  function . . .
Uplift-Y  function . . .

Planning Rules:

IF  (2D-Lateral-  X = rigid-frame)
AND  (2D-Lateral-Y = rigid-frame)
THEN  (layout-rigid-X layout-rigid-Y  . . .)

IF  (2D-Lateral-X = braced-frame)
AND  (2D-Lateral-Y = rigid-frame)
AND  (TotalLength Y-Bays) > (TotalLength X-Bays)
THEN  (layout-braced-X layout-rigid-Y   . . .)
 - - -
```

Figure 5: 2D-Panels System in STANLAY

An example of a system definition in STANLAY is illustrated in Figure 5. The system represents the 2D-Panels node in the decomposition tree. The attributes of the 2D-Panels system include layout information and load information. The layout attributes are selected and ordered by the planning rules. The load attributes, i.e. overturning moment in each direction (M_{over}) and uplift forces, are computed by Lisp functions. The layout attributes have values that represent alternative placement schemes, e.g. edges indicates that the panels are placed on the edges of the building only, edges+1 places a panel in the center of the building in addition to the edges. The combination and use of the placement schemes are checked by constraints for consistency with building geometry and intended occupancy. Other constraints in STANLAY check the load attributes for each of the subsystems and components for appropriate magnitudes.

5. CONCLUSION

EDESYN provides a formalism for synthesis of structural systems that facilitates the incremental development of a knowledge-base for preliminary structural design. The expression of design knowledge as systems with attributes, planning rules, constraints, and evaluation criteria is easy to work with and expand. The development of STRYPES, STANLAY, and other expert systems using EDESYN has led to a better understanding of the knowledge required for preliminary design and the representations needed for expressing this knowledge in an expert system.

REFERENCES

1. Cowan, H.J. and Wilson, F., Structural Systems, Van Nostrand Reinhold, 1981.

2. Fraser, D.J., Conceptual Design and Preliminary Design of Structures, Pitman, 1981.

3. Holgate, A., The Art in Structural Design, Oxford University Press, 1986.

4. Lin, T.Y. and Stotesbury, S.D., Structural Concepts and Systems For Architects and Engineers, John Wiley and Sons, 1981.

5. Maher, M.L., 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.

Leere Seite
Blank page
Page vide