Structural Design By Hierarchical Decomposition

by

Mary Lou Maher

EDRC 12-30-89
Carnegie Mellon University

Structural Design By Hierarchical Decomposition

Mary Lou Maher AM ASCE[1]

Abstract

Design by decomposition is a common approach to complex design problems. Ideally, decomposition results in loosely coupled subsystems and a reduction of the complexity of the overall problem. Design by decomposition also involves recomposirion when the subproblem solutions are combined to form a larger system. An experienced designer is able to recognize appropriate decompositions as well as the implications of recomposirion better than a novice designer, implying that knowledge based expert system techniques may facilitate the representation of this knowledge within a computer program. EDESYN is an expert system shell that facilitates the development of a design expert system incorporating the design by decomposition strategy. The experienced designer builds a knowledge base that includes both decomposition and recomposition knowledge. The decomposition knowledge includes a hierarchy of subsystems and planning rules to adapt the hierarchy for a specific problem. The recomposition knowledge includes constraints on the composition of subsystems. Two structural design expert systems have been developed using EDESYN: STRYPES and STANLAY. STRYPES synthesizes alternative structural configurations for a given high rise building using subsystems such as frames, walls, slabs, and grids. STANLAY generates alternative layouts for the structural system in a given floor plan and approximates the load and size specifications for the components of the system.

Synthesis

Design is a process during which design intentions are transformed into design descriptions. Design proceeds through several levels of abstraction, where more information about the requirements as well as the evolving design description is available as the process continues. 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 and subsystems are identified and their composition is evaluated.

[1] Assistant Professor, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA 15213.

There are many books that provide definitions and elaborations of the design process; in structural engineering such books include (Holgate, 1986), (Lin, 1981), (Fraser, 1981) and (Cowan, 1981). 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 cycle 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.

There is a need for design aids that support designers during synthesis. Many computer aids are available for evaluation, including the multitude of analysis, simulation, and optimization programs. Computer-aided design programs for drafting begin to support synthesis by providing visual feedback and facilitating changes in geometry. However, none of these computer programs support the designer in identifying and combining the appropriate design components. As a result, many of the design decisions must be made before a computer program can be used in structural design.

During synthesis a designer uses design knowledge to generate a design solution. A human designer does not need to explicitly define his design knowledge, it is implicitly developed and expanded as he gains experience. A design program, however, does contain an explicit representation of design knowledge. In this paper, the form this explicit representation of design knowledge can take is explored

In order to synthesize a design description for a complex structural system, the overall design problem can be decomposed into related subproblcms. The concept of decomposition is often applied to design problems when simpler subproblems can be identified and their interactions can be addressed. For example, in designing a structural system for a building, the designer may consider the lateral load resisting system and the gravity load resisting system separately before detailing their common components or constraints. Or a designer may find it more useful to consider the vertical subsystems and the horizontal subsystems as distinct design problems before detailing their interaction as a three dimensional system. The use of decomposition in design facilitates the process by allowing the designer to focus on a subset of the design decisions.

The development of design aids to support synthesis is facilitated by the use of

knowledge based expert system techniques. These techniques provide a programming environment for developing a knowledge base for a class of problems. The development of a knowledge base for a class of design problems, specifically for synthesis, requires the formalization of design knowledge. A knowledge base is typically represented as production rules, logic clauses, frames or objects. For synthesis purposes, using the decomposition approach, it is useful to think of the knowledge base in terms of the systems, subsystems, and components that are combined to form a system. The next section addresses a formalization of design knowledge useful during synthesis.

Formalizing Design Knowledge

The design knowledge for a class of design problems can be considered at multiple levels of abstraction. The representation of design knowledge for synthesis requires both high levels of abstraction as well as the lower levels of abstraction. The higher levels of abstraction represent the aggregations of components commonly used by designers. For example, a structural designer may consider tube structures or framed systems when designing a structural system for a building, but would consider cable systems or girder systems when designing a bridge. The lower levels of abstraction represent the individual components such as beams and columns, and their parametric values.

The aggregation of components into systems provides a basis for synthesis by hierarchical decomposition. Formalizing design knowledge requires explicitly identifying the commonly used and understood systems, subsystems, components, and their interactions. The knowledge associated with the systems includes planning knowledge on how to proceed with the design of the system in terms of the appropriate subsystems or components and parameters that describe the system in terms of its intended function and geometry. It is possible to generalize the knowledge required for synthesis into two broad categories: systems and constraints.

A system can represent an aggregation of components or a class of components. Each system can be described by a set of attributes and the knowledge needed to generate a description. The attributes can represent functional descriptions, dimensional descriptions, and properties. Each attribute can have an associated value range, indicating its intended use. For example, different classes of beams have different span ranges. The knowledge for generating a design description includes both planning knowledge and attribute value knowledge. The planning knowledge indicates the order in which the attributes are considered. The attribute knowledge indicates the procedure for generating a value for each attribute.

A constraint represents a relationship that must be true for a design description to be valid. Constraints provide the knowledge needed to produce valid compositions of systems. The concept of synthesis by decomposition implies a recomposition. Constraints impose restrictions on the compositions, both at the aggregation level as well as the component level. At the higher levels of abstraction, constraints explicitly represent the interaction between subsystems. At

the lower levels of abstraction, constraints represent restrictions on descriptions. For example, a constraint on interaction of subsystems may consider material compatibility in the lateral and gravity systems. A constraint on a description of a column may require that the section does not buckle.

EDESYN (Maher, 1987) is a programming environment that facilitates the development and use of a knowledge base for synthesis. The implementation of EDESYN follows the philosophy of current expert system techniques, maintaining a separation of knowledge base and inference mechanism. The architecture of EDESYN is illustrated in Figure 1 and the components are described below.

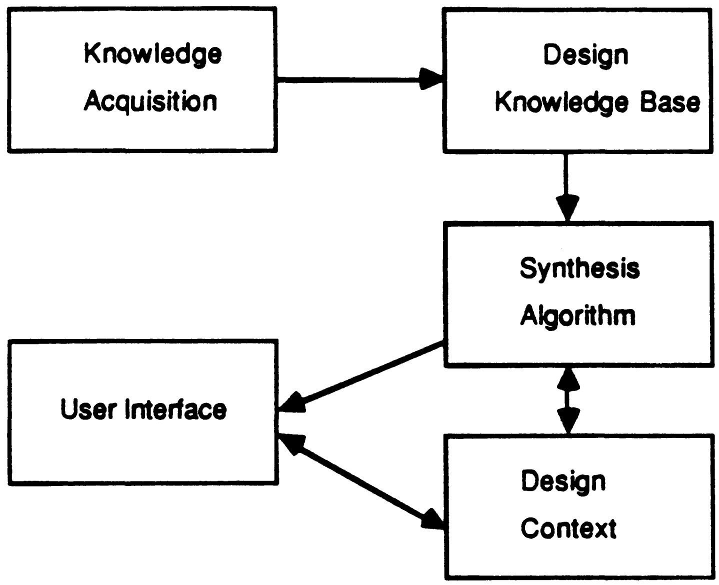

Figure 1: Architecture of EDESYN

Design Knowledge Base The experienced designer defines a knowledge base that includes decomposition, planning, and constraint knowledge. The decomposition knowledge is specified as systems and subsystems, where each system comprises a set of attributes. An attribute may be another system or a simple attribute. 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 tube 2D-orthogonal)
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.

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:

```
constraintl
stories > 30
3D-lat«ral • 2D-orthogonal

constraint2
2D-1ataral-x/material » stmml
2D-1ataral-y/material * concrata
```

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.

Synthesis Algorithm 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. The attributes are assigned legal values, where a legal value is one that does not get eliminated by the constraints. All feasible alternatives are generated for each system, using the planning rules to define and order the subsystems and attributes.

Design Context The design context initially contains the requirements and specifications associated with a particular design problem. For example, the initial 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 an attribute and its value. Along with the solution tree, a hierarchy tree is maintained to associate each node in the solution tree with the system for which it was generated.

EDESYN is implemented in Framekit (Carbonell, 1985), a frame based reasoning tool written in Common Lisp. EDESYN currently runs on a MicroVax II and a Sun 3/60. The experienced designer defines the knowledge base by creating files of decomposition and constraint knowledge using a syntax similar to the description provided above. The designer uses the resulting knowledge base through a multi window user interface. The designer specifies a particular design problem and then interacts with EDESYN during the synthesis process. The feasible alternatives are presented to the designer in the form of the solution tree. The designer can request more information for each alternative by pointing to a node, including an icon that illustrates the alternative.

Synthesizing Structural Systems

EDESYN has been used to develop two expert systems for structural system synthesis: STRYPES and STANLAY. Much of the knowledge in STRYPES and STANLAY was identified through the development of HI-RISE (Mahcr, 1984). These expert systems are part of a larger design environment for integrated building design (Fenvcs, 1987). STRYPES includes design knowledge for configuring alternative structural systems and materials for rectangular buildings between 10 and 50 stories. STANLAY includes design knowledge for the layout of lateral load resisting systems given a building plan and the design of the structural components using approximate analysis techniques.

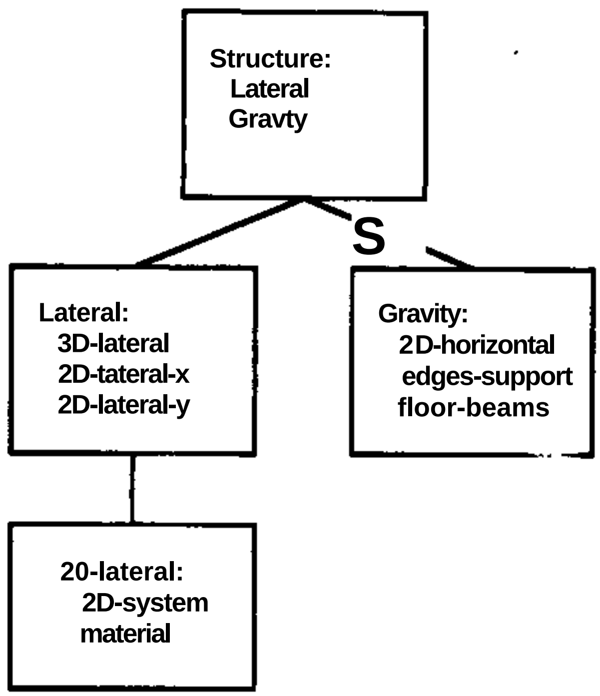

Figure 2: STRYPES Decomposition Knowledge

The STRYPES knowledge base includes four system definitions: structural system, lateral system, gravity system, and 2D lateral system. The synthesis of a structural configuration starts with the decomposition into the lateral and gravity systems. The lateral system is decomposed into the design of a 3D lateral system, represented by a selection among alternatives such as tube or core, and the design of 2D lateral systems in each of the orthogonal directions. The design of a 2D lateral system is decomposed into the selection of a 2D system, such as rigid frame, braced frame, or shear wall, and the selection of an appropriate material. The gravity system is decomposed into the selection of a 2D horizontal system, such as concrete slab, steel deck, or prefabricated panels, and the number of edges supported by girders

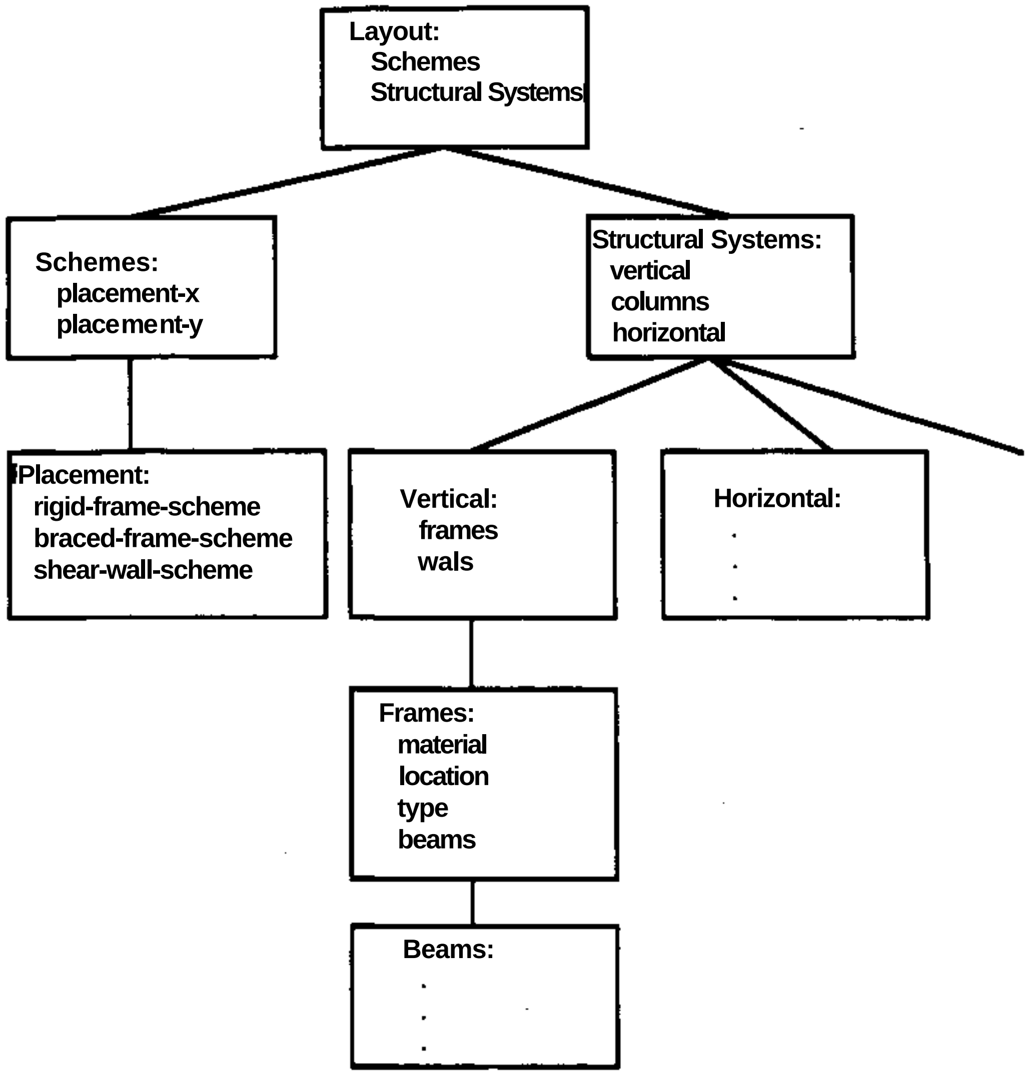

Figure 3: STANLAY Decomposition Knowledge

and the use and direction of intermediate floor beams. The decomposition knowledge for STRYPES is illustrated in Figure 2.

The constraints in STRYPES represent restrictions on structural system types and materials for different classes of buildings. Certain lateral and gravity systems are considered incompatible, for example a steel lateral system would not be used with prefabricated concrete floor panels. Also, certain lateral systems are not considered depending on the intended use of the building, for example an office building would not have multiple concrete shear walls for the lateral system. Constraints on the design of the gravity system include restrictions on the combination of a 2D horizontal system, the suppon conditions, and the structural grid.

The STANLAY knowledge base contains many system definitions, including layout systems and structural systems. The layout systems represent alternative classes of lateral system placement schemes for buildings; in which specialized schemes exist for framed structures and shear wall structures. The structural system definitions include frames, walls, slabs, beams, and columns. The attributes of the structural systems describe their geometry and components. For example, the frame system has the following attributes: material, grid location, type (e.g. rigid or simple), and beams. A subset of the decomposition knowledge in STANLAY is illustrated in Figure 3.

The constraints in STANLAY include layout constraints and structural system constraints. The layout constraints check the compatibility and stability of a potential layout. Compatibility restricts the use of inappropriate layout schemes for certain types of buildings. For example, an office building should have an open layout, restricting the number of interior bays that are blocked by lateral systems. Stability constraints check the lateral system layout for potential uplift. The structural system constraints restrict the assumptions about behavior. For example, a concrete slab can be designed for one way or two action depending on the span ratio.

Conclusion

Supporting structural system synthesis using knowledge based techniques requires formalizing both a synthesis process and design knowledge. One approach to the synthesis process is hierarchical decomposition; requiring design knowledge in the form of systems, subsystems, and constraints. An expert system shell, EDESYN, has been developed for engineering design synthesis and applied to the structural system domain. The resulting expert systems, 3TRYPES and STANLAY, illustrate the potential for supporting the synthesis of alternative design solutions by formalizing design knowledge.

References

Carbonell, J.G. and Joseph, R. (1985). *FRAMEK1T+: A Knowledge Representation System,* Technical Report, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Cowan, J.J. and Wilson, F. (1981). *Structural Systems,* Van Nostrand Reinhold.

Fcnves, S.J., Flemming, U., Hendrickson, C, Mahcr, ML. and Schmitt, G. (1987). An Integrated Software Environment for Building Design and Construction, in *Fifth Conference on Computing in Civil Engineering,* ASCE.

Fraser, DJ. (1981). *Conceptual Design and Preliminary Analysis of Structures,* Pitman Publishing Inc.

Hoigate, A. (1986). *The Art in Structural Design,* Oxford University Press.

Lin, T.Y. and Stotesbury, S.D. (1981). *Structural Concepts and Systems for Architects and Engineers,* John Wiley and Sons.

Maher, M.L. (1984). *HI-RISE: A knowledge-Based Expert System for the Preliminary Structural Design of High Rise Buildings.* Ph.D. thesis, Department of Civil Engineering, Carnegie-Mellon University.

Maher, M.L. (1987). *Engineering Design Synthesis: A Domain Independent Representation.* AI EDAM, 1 (3):207-213.