Carnegie Mellon University
Research Showcase @ CMU

Department of Civil and Environmental Engineering

Carnegie Institute of Technology

1989

Synthesis and evaluation of preliminary designs

Mary Lou. Maher
Carnegie Mellon University

Carnegie Mellon University.Engineering Design Research Center.

Follow this and additional works at: http://repository.cmu.edu/cee

Recommended Citation
Maher, Mary Lou. and Carnegie Mellon University.Engineering Design Research Center., "Synthesis and evaluation of preliminary designs" (1989). *Department of Civil and Environmental Engineering.* Paper 38.
http://repository.cmu.edu/cee/38

This Technical Report is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase @ CMU. It has been accepted for inclusion in Department of Civil and Environmental Engineering by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

Synthesis and Evaluation of Preliminary Designs

by

Mary Lou Maher

EDRC 12-29-89
Carnegie Mellon University

SYNTHESIS AND EVALUATION OF PRELIMINARY DESIGNS

Mary Lou Maher

Department of Civil Engineering, Carnegie Mellon University, Pittsburgh PA, USA

ABSTRACT Design is a process of producing a description of a system or process to satisfy a set of requirements. To suppoit the designer in the identification and composition of components of design solutions 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 knowledge based techniques for the exploration of synthesis and evaluation methods 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 paper, models of knowledge based synthesis and evaluation are presented. Synthesis is based on a constraint directed search through a design space that is decomposed into subsystems, components, and constraints. Evaluation is based on the concept of Pareto optimality for identifying a set of optimal 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 knowledge baaed synthesis programs, where the experienced designer defines a knowledge base and the designer uses the resulting knowledge base to produce design solutions.

DESIGN AS A PROCESS

Design is a process during which design descriptions are generated to satisfy design intentions where identifying the design intentions is as important as identifying the appropriate design description. 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

University Libraries
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

preliminary design, the major components and subsystems are identified and their composition is evaluated

There are many books that provide definitions and elaborations of the design process; in structural engineering such books include Hogate [5], Lin [6], Fraser [4] and Cowan [2]. 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 ..^signer.
- *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. Synthesis involves identifying the form of the design solution. Evaluation, during the early stages of design, is usually based on a subjective assessment of relevant criteria. Although synthesis and evaluation may be ..sed on associated quantitative models, the designer typically reasons abou· iese models in a qualitative manner. The knowledge used during synthes: md evaluation of preliminary designs is not well articulated. Experience. designers resort to trial and error less frequently than novice designers when searching for an appropriate or satisfactory form, suggesting that the use of knowledge-based systems to represent "experience[9] may improve design synthesis and evaluation.

SYNTHESIS BY DECOMPOSITION

During synthesis a designer considers a design space which contains the knowledge that is used to develop the design solution. A human designer does not explicitly identify his design space, it is implicitly developed and expanded as he gains experience. A design program, however, docs contain an explicit representation of the relevant design space. The nature of the knowledge in the design space is of interest when considering a knowledge based approach to design.

Given that design can be modeled as a search process, the design space represents the space in which the search operates. A design goal is a concept central to design processes. A design goal can represent a functional or physical selection, or can be further decomposed into a set of subgoals. The set of design goals, the alternative solutions for each goal, and the relationships between goals and subgoals can provide a description of the design space for a class of engineering systems. For example, in designing a structural system for a building, design goals include: design a gravity load resisting system, select a structural material, design a beam section, etc. Knowledge of these goals and the possible solutions provides a basis for reasoning about design synthesis.

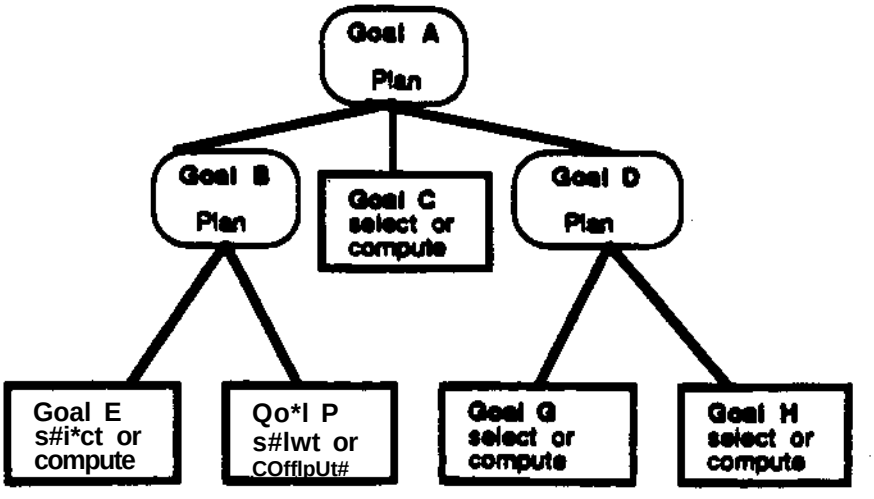

Figure 1: Goal Decomposition

As shown in Figure 1, a design space can be represented by goals and their decomposition. A goal can be satisfied directly or decomposed into subgoals. A terminal node in the goal tree represents a design decision that can be taken directly, either by selecting from a discrete set of alternatives or by calculating a numerical value. An example of a terminal node from structural design is the selection of a structural material, either steel or reinforced concrete, or the calculation of the depth of a beam. A non terminal node in the goal tree represents a goal that can not be satisfied directly. This type of node can be labelled a synthesis node, requiring that the satisfaction of the goal involves decomposition and synthesis. An example of a synthesis node is the design of a lateral system, since the goal can not be satisfied by the selection from a discrete set of alternatives or a calculation.

In addition to design goals, the design space also includes knowledge about legal operators or decisions. The legal decisions can represent either planning knowledge or design knowledge. Planning knowledge includes

information on which design goals are relevant to the current situation. For example, in designing a structural system, the goals relevant for the design of a concrete frame include selecting reinforcing bar sizes and spacing, which differs from the goals relevant for the design of a steel frame. The relevant goals for a given situation depend on the satisfaction of previous goals. A designer must be flexible enough to refine the relevant goals as the design process proceeds.

In addition to planning knowledge to identify legal operators, constraints are needed to identify legal decisions. Where planning knowledge guides the satisfaction of a synthesis node, constraints guide the satisfaction of a terminal node. Constraints represent relationships among variables and their values. The variables can have numerical values or non-numeric values. Constraints can represent relationships that must be satisfied or relationships that constitute an illegal decision. In the synthesis method employed in this project, the constraints represent illegal situations and serve to prune the search space by eliminating infeasible alternatives explicitly.

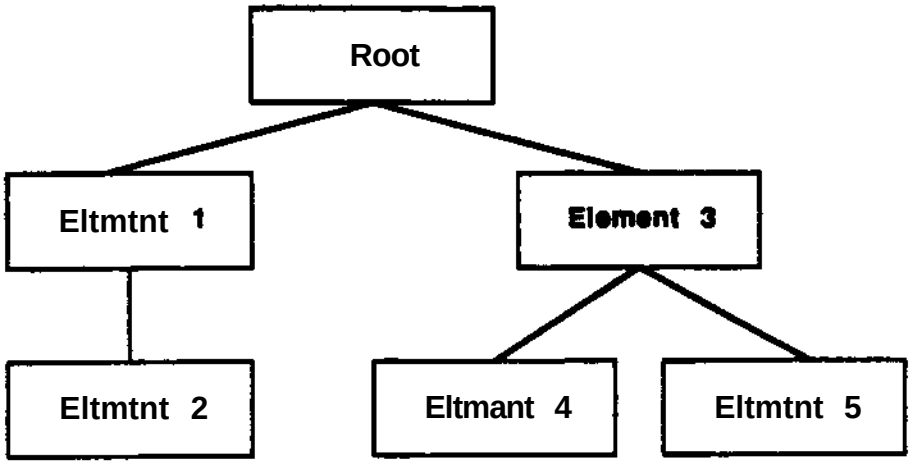

Figure 2: Element Recomposition

The result of synthesis is the recomposition of the solutions associated with each terminal node in the goal tree. Since a solution is eliminated only if a constraint identifies it explicitly, multiple feasible solutions are generated during synthesis. The solution tree shown in Figure 2 represents the alternative feasible designs as a collection of elements, where any path through the tree respresents one solution. The element numbers in Figure 2 represent the order in which the decisions were made; this order depends on the planning knowledge. The depth of the tree is equal to the number of terminal nodes in the goal tree. The branching factor depends on the number of alternatives associated with a terminal node and the number of constraints

available to prune the search space. Evaluation of partial designs is used to supplement the constraint knowledge in dfccriminating among alternative design solutions. Where a constraint represents a relationship that cannot exist, evaluation knowledge represents relationships that have relative merit.

EVALUATION USING MULTIPLE CRITERIA

During preliminary design, evaluation of feasible alternatives is based on multiple criteria and incomplete or partial information. A designer is interested in the "best" design, suggesting that optimization techniques may be appropriate for synthesis and evaluation. However, in the absence of complete information it is not feasible to use an optimization technique that requires a mathematical formulation of the problem. Hence, the mathematically rigorous optimization techniques do not serve our purpose. Also, most optimization techniques require the identification of a single objective function when it is difficult to come up with a single criterion to evaluate partial design alternatives. One way to consider multiple criteria is to discriminate between them using tradeoffs and weights for each of the criteria under which the solutions are evaluated. This approach is of interest but we would like to introduce it as late in the evaluation process as possible. To avoid using discrimination so early in the design process the notion of Pareto optimality is adopted. We use a two stage evaluation process. During the first stage, the number of alternatives is reduced by removing the dominated alternatives. During the second stage, subjective information about the designers preferences is used to rank the remaining alternatives.

The concept of Pareto optimality was formulated by V. Pareto in 1896 (Radfoid [9]). The optimum, for multiple criteria, is commonly stated as the following:

> A feasible solution to a multicritcria problem is Pareto optimal if no other feasible solution exists that will yield an improvement in one criterion without causing a degradation in at least one other cnierioa(Radfbrd [9])

This definition implies that there may be a set of solutions that can be considered the optimum before using preferences.

Pareto optimality can provide insight in identifying the appropriate or best design solutions to pursue given a set of feasible alternatives. Rather than using the concept of Pareto optimality to generate a set of alternatives, the concept is used to find a set of non-dominated solutions given a set of feasible alternatives. The set of the non-dominated (Pareto) solutions is determined by a pairwise comparison of the feasible alternatives for each criterion. The procedure is as follows:

Let P be a set of Pareto solutions where P_l is a member of the set,

S be a set of feasible alternatives where S_i is a member of the set,

C be a set of criteria where C_k is a single criterion.

The set of feasible alternatives is compared pairwise by assigning the first member, S_1, of the set of feasible alternatives, S, to be a member of the Pareto set P. S_i is then compared to the next member of the set of alternatives, say S_j. S_i is said to be less than S_j if and only if values of all the criteria of S_i are less than those of S_j, i.e. $C_k^i < C_k^j$ for all k. The comparison proceeds as follows.

- If $S_i < S_j$, where $i \neq j$, then S_i is replaced as the member of set P by S_j and S_i is discarded from further consideration.
- If $S_i > S_j$, where $i \neq j$, then S_i remains a member of set P and S_j is discarded from further consideration.

If neither of the above conditions are true, then both S_i and S_j are retained as members of set P. This procedure is repeated until the set of feasible solutions is exhausted and P is a set of non-dominated alternatives. Only these non-dominated solutions are considered for further evaluation.

Identifying the non-dominated set of solutions does not provide the designer with any information regarding their relative merit. This can be accomplished by ranking the solutions by discriminating between them based on preferences. The preferences are specified as weights for each criterion. The rank of each solution in set P is assigned based on a total value T, calculated as follows.

$$T_i = \sum_{j=1}^{k} W_j \ x \ NV_{ij}$$

$$NV_{ij} = \frac{V_{ij}}{nf_j \ x \ nd_j}$$

where T_i = Total Value for solution i

W_j = Weighting factor for criterion j

NV_{ij} = Normalized value for solution i under criterion j

V_{ij} = Value for solution i under criterion j

nf_j = Normalization factor for criterion j

nd_j = Non-dimensionalizing factor for criterion j

The normalized values are calculated by utilizing normalization and non-dimensionalizing factors. The normalization factor is the maximum value **expected for that feature.** The non-dimensional factor depends on the units of the **feature** being considered The normalized value, i.e. NV_{ij9} is normalized to a **range** of +1 to -1. The positive value indicates "good" performance while a negative value indicates poor performance of the solution in the particular criterion being considered.

The solutions are sorted and ranked in the decreasing order of their total value. This means the alternative with the highest numerical total value is assigned the 1st rank and so on. This alternative is the "best" in the set given the preferences.

EDESYN

EDESYN (Maher [8]) is an implementation of the methods described above that facilitates the development and use of a knowledge base for design; particularly for synthesis and evaluation of preliminary designs. The implementation of EDESYN follows the philosophy of current expen system trechniques, maintaining a separation of knowledge and inference mechanism. The methods described above are implemented as algorithms within the inference mechanism and the design knowledge is maintained in the knowledge base. The architecture of EDESYN is illustrated in Figure 3.

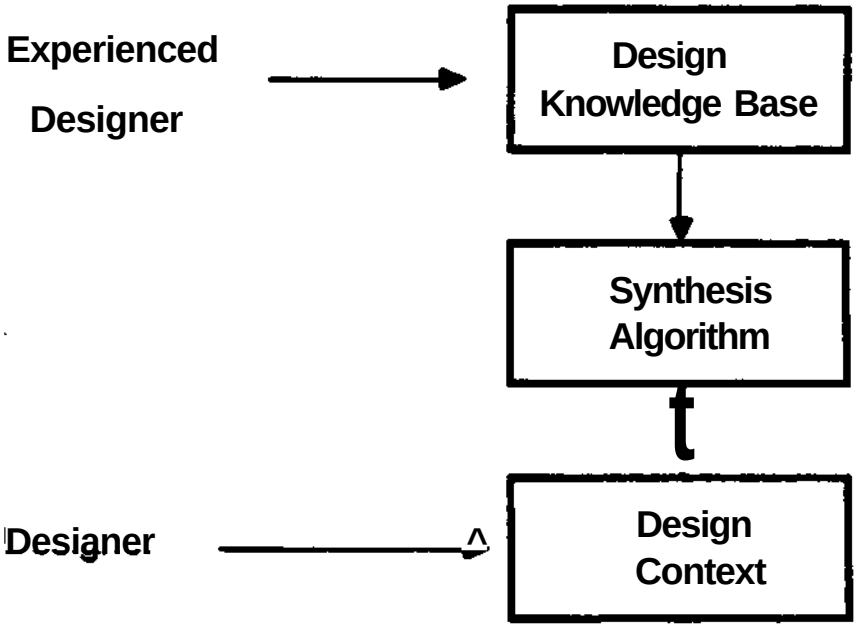

Figure 3: Architecture of EDESYN

Design Knowledge Base

The experienced designer defines a knowledge base that includes decomposition, planning, constraint, and evaluation knowledge. The decomposition knowledge is specified as systems and subsystems, where each system comprises a set of attributes. An attribute may be another system (i.e. subsystem), representing a synthesis node in the 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 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.

The constraints ait 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:

```
constraint1
stories > 30
3D-lateral » 2D-orthogonal

constraint2
2D-lateral-x/aaterial * steel
2D-lateral-y/material * concrete
```

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 die 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 suffer than a rigid frame structure.

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 goal tree. The terminal nodes are assigned all legal values, where a legal value is one that does not get eliminated by the constraints. All feasible alternatives are generated for a non-terminal (system) node, using the planning rules to define and order the decendant nodes. After the alternatives for a system node have been synthesized, the evaluation mechanism is invoked. In the structural design example, evaluation occurs after each system is synthesized, e.g. lateral system. The alternatives are compared for each criteria to produce the non-dominated set of solutions, which are then ranked using the preferences in the knowledge base. At this point, the solutions are presented to the designer along with the evaluation information and the designer chooses one solution for further consideration.

Design Context

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 t tree of alternative solutions, where each node in the tree represents a solution for a terminal node in the goal tree. 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 [1]), 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, constraint, and evaluation knowledge using a syntax similar to the description provided above. The designer uses the resulting knowledge base through a multi window user interface, as shown in Figure 4. The designer edits the precondition window

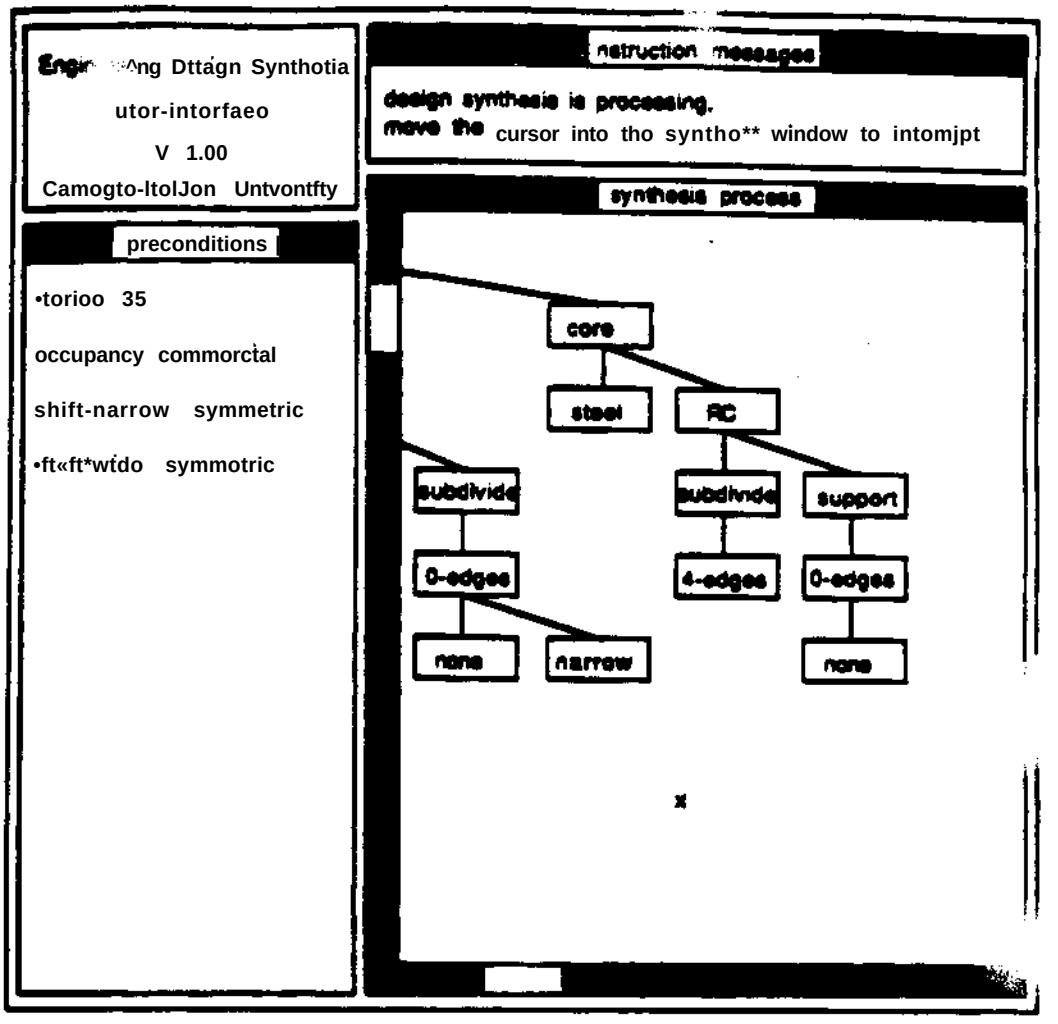

Figure 4: EDESYN's User Interface

to specify a particular design problem and then interacts with EDESYN during the synthesis process in the synthesis window. The feasible alternatives **are** presented to the designer in the form of the solution tree. The **designer can request more** infonnation for each alternative by pointing to a **node. The information** associated with every node includes an icon that **illustrates the alternative.** The information associated with a terminal node **includes the evaluation** information, such as criteria and their values.

APPLICATIONS

EDESYN has been used to develop design expert systems in several different domains. The synthesis process has evolved from the issues inherent in developing structural design expert systems. The structural design expert systems include STRYPES, STANLAY, and FOOTER. The STRYPES knowledge base contains knowledge for configuring alternative structural

systems for buildings between 10 and 40 stories. STRYPES knows about lateral and gravity systems and materials. The STANLAY knowledge base contains knowledge for the layout of lateral load systems given a building plan and approximate analysis of lateral and gravity systems. Much of the knowledge in STRYPES and STANLAY was identified through the development of HI-RISE (Maher [7]). The FOOTER knowledge base contains knowledge about the selection and design of footings and piles for building foundations. These three knowledge based expert systems are part of a larger design environment for integrated building design (Fenves [3]).

EDESYN has also been used to develop knowledge based expert systems in the following domains.

- Designing the manufacturing process for a gear, including the selection and combination of rough forming operation, pre- and post-machining heat treatment, and precision machining.
- Designing an industrial robot; including the design of the wrist, joints, motor and material selection.
- Designing heat exchanger configurations; including the selection and combination of alternative transfer processes, surface compactness, construction and flow arrangement
- Configuring computer equipment; including the selection and combination of alternative hardware and software.
- Synthesizing a finite element program from subroutines; including the identification and selection of appropriate solution methods, subroutines and element meshes.
- Designing stairwells; including the selection of dimensions (risers, treads, stair run, and stair width), construction (material and riser type) and configuration (shape and landings).

The applications of EDESYN have been limited to relatively small design knowledge bases. However the diversity of the applications have highlighted some of the advantages and limitations of this implementation. The major advantage is the ease in developing a knowledge base and the support for incremental development The major limitation in EDESYN is the clumsy way in which it reasons about parametric design due to the emphasis on symbolic manipulation rather than numeric.

DIRECTIONS

The directions for continuing to consider both synthesis and evaluation of designs using knowledge based techniques include the addition of multiple design strategies for each node in the goal decomposition tree. EDESYN is currently limited by its restrictions on goal satisfaction, where a goal can

either be decomposed or directly assigned a value. The additional strategies being considered are best first search for a single design solution, mathematically rigorous optimization techniques, and case based reasoning. The design strategy to be used at any node should depend on the design knowledge available.

Acknowledgements. This work was supported by the Engineering Design Reaserch Center at Carnegie Mellon University. Many students contributed to the development of EDESYN, including Shitul Patel, Chi-ming Yang, and James Lord.

REFERENCES

1. Carbonell, J. G. and Joseph, R. (1985). *FRAMEKIT+: A Knowledge Representation System*, Technical Report, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
2. Cowan, J.J. and Wilson, F. (1981). *Structural Systems*, Van Nostrand Reinhold.
3. Fenves, S.J., Flemming, U., Hendrickson, C., Maher, M.L. and Schmitt, G. (1987). An Integrated Software Environment for Building Design and Construction, in *Fifth Conference on Computing in Civil Engineering*, ASCE.
4. Fraser, D.J. (1981). *Conceptual Design and Preliminary Analysis of Structures*, Pitman Publishing Inc.
5. Hogate, A. (1986). *The Art in Structural Design*, Oxford University Press.
6. Lin, T.Y. and Stotesbury, S.D. (1981). *Structural Concepts and Systems for Architects and Engineers*, John Wiley and Sons.
7. Maher, M.L. (1984). *HI-RISE: A knowledge-Based Expert System for the Preliminary Structual Design of High Rise Buildings*. Ph.D. thesis, Department of Civil Engineering, Carnegie-Mellon University.
8. Maher, M.L. (1987). *Engineering Design Synthesis: A Domain Independent Representation*. AI EDAM, 1(3):207-213.
9. Radford, A.D. and Gero, J.S. (1988). *Design by Optimization*, Van Nostrand Reinhold.