Case Studies of Design Methodologies: A Survey*

Jigang Yang and Jerzy W. Rozenblit
Dept. of Electrical and Computer Engineering
University of Arizona
Tucson, Arizona 85721

ABSTRACT

Design theory and methodology is an emerging area of research. It supports understanding of the design process and its automation in the form of Computer Aided Design. This paper is focused on design methodologies used in computer aided design systems. Several typical cases are reviewed. They are: Computational Model for Hierarchical Mechanical System Design, Optimal Directed Design Model, Knowledge-based Creative Design Model, Model-based System Design. Common trends in the area of design methodologies are also discussed .

1. INTRODUCTION

A design methodology is a very important basis for supporting automation of the engineering design process. This paper is a survey of design methodologies that employ model-based frameworks. A number of methodologies and design systems have been developed to aid the engineering design process in different domains. Although different systems use different models to represent design knowledge and different systems work in different domains, some common features can be found and comparisons can be made among them.

In order to understand a design methodology for CAD, several concepts must be clarified. Engineering design can be considered as a technological activity in which knowledge about a specific domain is employed. It is an activity that seeks all relevant knowledge and combines it to produce a design solution. One view of design is that it is a search process in which a satisfactory design solution is produced from a number of alternatives (Gero, John S. st al, 1989). Those alternatives come from knowledge of the relevant domain. The search is done in a design space which includes all knowledge and design decisions known so far. The design space can be expanded during the design process, i.e., new knowledge and design decisions can be added to the space when the existing knowledge and available design decisions are not sufficient to obtain the design solution. If a design solution can be achieved using only existing knowledge in the design space, the process is then termed *routine design*. In routine design, no new knowledge is added to the design space and the design space is not expanded. If a design solution cannot be achieved using only the existing knowledge in the design space, new knowledge must be added. This kind of design is called *creative design*. In creative design, new knowledge is added into the design space, and thus the space is expanded. The new knowledge marks the creativity of design. The design, which is a search process in the design space, is guided by some principles. These principles are given by a design methodology. In other words, design methodology provides a search method in the design space.

The first important feature of a design methodology is the *creativity factor*. If only the existing knowledge is used in the design process and there is no method to add new knowledge, then the methodology is suitable for routine design and the creative factor of the methodology is limited. If a design methodology allows for new knowledge to be added when necessary, the design space can be expanded during the design process and the design methodology can be considered creative. The second feature is the representation method used in different stages of the design process. Some methodologies use a unique representation method throughout the whole design process. Some employ several different representation methods. The third characteristic is *domain independence*. Some methodologies can be used only in a specific domain. Some are domain independent. Engineering design problems are often quite large and complex. The decomposition of the large, complex design problem into subproblems is another characteristic of design methodologies. In the following sections, several case studies illustrating the discussed characteristics of design methodologies are presented (Rozenblit, J. W. and Zeigler, B. P., 1988).

2. Computational Model for Hierarchical Mechanical System Design

This design methodology is based on a tree-like hierarchical model. It was proposed by J. R. Dixon, et.al. (1989,1988). It handles parameteric design problems in which the attributes of the design are known but the values of the attributes must be determined so as to satisfy a certain goal and contraints. The design process is directed by a tree-like structure. The root and interior nodes represent the subsystem manager. The terminal nodes represent the component designer. The subsystem manager is responsible for the problem specification and conflict resolution among lower level subsystem managers and component designers. The component designer is responsible for creating components that satisfy the constraints and design specifications. The design process is shown in Figure 1.

* This work was partially sponsored by Siemens Corporate Research and Support Division, Princeton

TH0308-7/90/0000/0136/$0.100 © 1990 IEEE

The first step in designing is to specify a design problem. To describe the problem, the following terms are defined:

- Problem Definition Parameters – The problem for each node is defined by a set of problem definition parameters (PDP). It's value is the specification for the system being designed.
- Design Variables – The design variables of a component are the set of parameters that define the component.
- System Design Variables – These are design parameters that are not chosen by any of the subsystems but are part of the overall system design.
- Performance Parameter – It is a quality measurement of a design in terms of the performance.
- Analysis Procedure – Performance parameters are calculated by the procedure which produces the value of the performance parameter by summarizing the performance parameter values of its subordinate nodes.
- Satisficing Objective – It indicates the relationship between designer satisfaction and the value of a performance parameter.
- Extremum Objective – It is one which requires maximization or minimization of performance parameters.
- Design Requirements – These are similar to contraints in optimization and indicate that these equation like constraints must be satisfied in order for the design to be acceptable.
- System Quality Level – It is based on the value of each satificing objective and the subsystem quality levels. Each of them has a priority so as to rate the level.
- Solution Parameters – The output of each subsystem includes the set of design variables' values and solution parameters. These are needed by the superior manager.

The design process starts from the design problem decomposition. The root creates the subsystem PDP and the subsystem in turn creates PDPs for the lower level subsystems or component designers. Each subsystem node and component node has its own PDP and a desired quality level. The subsystem manager decides what quality level the lower level nodes should have. The task of a lower level node is to create the design according to the specification (PDPs) within the quality level.

A design problem can be decomposed into two types of subproblems: order dependent and order independent. The first type requires the subproblem to be solved in a specific order. It means that some subsystems are determined by the solution parameters of other subsystems. The second type is order independent. The subsystems can be designed in parallel.

The computational model is based on a hierarchically managed specification. In this model, the interior nodes solve the assigned design problem. This is achieved by selecting values for system design variables and by specifying subproblems for the subordinates. When the subsystem solution is returned, the manager node integrates the subsystems and evaluates them. If the result is not satisfying with respect to the design requirements and satisficing objective, the manager node prepares new values for the system design variables and new specifications for the subordinates. The process continues until an acceptable solution is obtained. Thus, the design problem is decomposed into component design problems, and then integrated into a solution.

This model is in general domain independent, although it has been used only in the mechanical parametric design. The representation during the design process is unique and hierarchical. It is suitable for routine design to some extent, because it assumes that the parameters of a system are pre-defined.

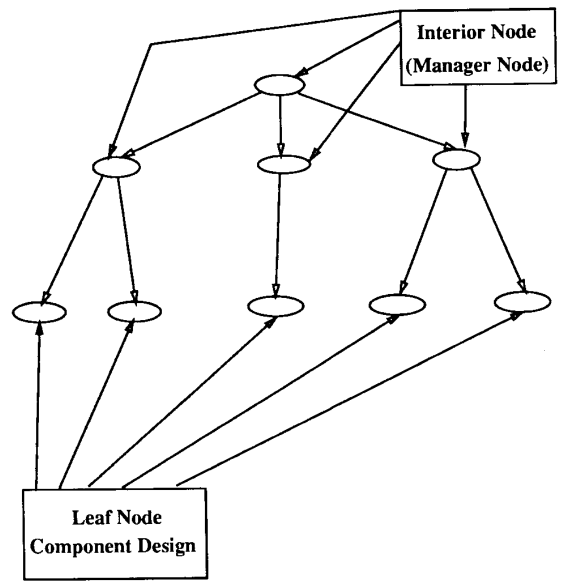

Figure 1. Hierarchical Structure

3. Optimally Directed Design Model

The optimally directed design model was proposed by Alice M.Agogino, et al(1989,1988). This model provides a theoretical framework to incorporate qualitative, functional (mathematical), and numerical methods in engineering design. The approach is based on the theory that a good designer combines both basic knowledge of engineering principles and optimizing trends with experience of successful and unsuccessful design. The design is viewed as a goal-directed, constrainted activity. Designers are optimizers with limited resources. Designers reason qualitatively, functionally and numerically. Effective design systems must reason at all those levels and incorporate both contraints and goals.

Optimization is essential since the design objective is to create a design that not only meets the minimum requirements but also ranks high relative to the goal. Analysis and simulation are fundamental concepts supporting the methodology. Optimization, analysis and simulation are both qualitative and quantitative.

The qualitative reasoning is defined as reasoning about objects and their qualities or parameters in the way that does not rely on specific numerical values. At this level, AI and knowledge-based concepts are used. Computer programs for equation solving or finite element analysis are examples of numerical reasoning.

The main components of the design system proposed by Agogino are:

- 1st Prince – 1st principle computational evaluator, deriving new design features from the previous ones.
- CODESIGNER – Conceptual design environment.
- SYMFUNE – Symbolic functional evaluator.
- SYMON – Symbolic monotonicity analyzer.

1st PRINCE incorporates the qualitative, functional, and numerical levels of reasoning. It utilizes SYMON and SYMFUNE to do the monotonicity analysis. The CODESIGNER system assists the designer in choosing initial conceptual design and related parameters (the initial prototype). The data are then passed to SYMON and SYMFUNE for qualitative and functional optimization using the monotonicity analysis and the mathematical functional backsubstitution. The symbolic form result can be translated directly to the form suitable for numerical analysis and optimization. The structure of the system is shown in Figure 2.

The CODESIGNER can be considered as the user interface. It uses a concept network to represent the design prototypes. A concept network is a set of linked concepts. A concept is a class of design prototypes with a number of models which model the design object qualitatively, mathematically or numerically. It describes a device with associated documents about the structure, function and manufacture. These are arranged hierarchically. Once a concept for a device is chosen, models associated with the device may be used to select parameters to describe a device and to impose constraints on them. Designer can build a new model and modify the concept network.

The optimally directed innovation of design is a process of deriving new design features from the previous ones. The innovative design process that has some relation to previous designs can be reached by the transformation of the prototypes. This innovation allows for creativity in this design methodology.

This model is quite domain dependent, because the internal mechanism is mechanical engineering oriented. It uses a hierarchical representation in the CODESIGNER subsystem (concept network). The model is suitable for both creative and routine design since the CODESIGNER can manipulate the existing concept network to produce new design knowledge.

4. Knowledge-Based Creative Design Model

The knowledge-based creative design model is based on the idea of a design space (Gero 1989). The design here is defined as a search process oriented through the design space. A design solution is represented by design operators that can be applied to produce design solutions. The search process is to find appropriate design operators and use them to generate the design solution. An initial state in the design space is the specification of the design problem. Each operator chosen by the search process constitutes a partial solution, as shown in Figure 3. The final state is the complete solution which consists of all operators.

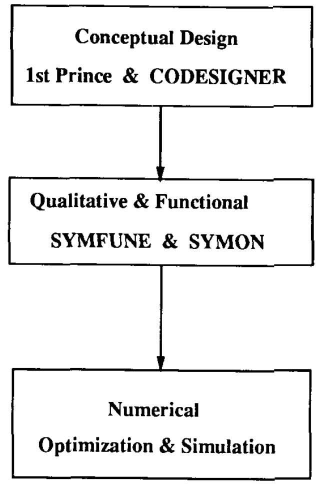

Figure 2. Optimally Directed Model

The *prototype concept* is used as a mechanism to employ knowledge. It serves as a basis for representing operators in a class of design solutions. Although prototypes are developed by individuals, there exists a general agreement about the constraints and disciplines of the design problem. A design process may start from an initial prototype. Then other prototypes are brought into the design so as to incorporate all necessary operators to reach the complete solution. Prototypes can be used in three ways: prototype refinement, prototype adaptation, and prototype generation. The prototype refinement involves an application of the prototype to the current state of the design solution. The prototype adaptation involves the modification of the contents of the prototype resulting in the production of a design description that is not derivable from the original prototype. The prototype generation involves producing new prototype that is a substantial modification of the existing prototype.

Analogy is used as an approach for solving new problems. By analogy, the relations between the new problem and some existing knowledge can be found. This knowledge can be placed in the new situation so as to get better understanding of the design solution. Analogy is an effective problem solving approach and is a tool leading to the creativity in design.

Another approach is *mutation* which is a deliberate action of changing attributes of an object (prototype) in an unconventional manner. The change is not restricted by the accepted rules or constraints. The purpose is to find new attributes of an old object which may resultin a new design. The knowledge-based creative design model is basicaly a design system for both creative and routine design by using prototype, analogy and mu-

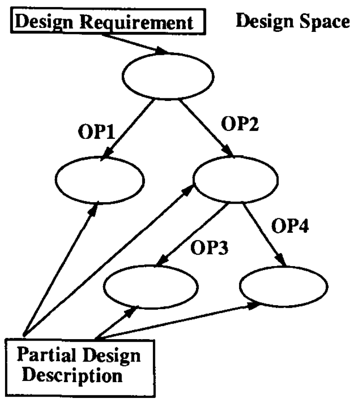

Figure 3. Design & Design Space

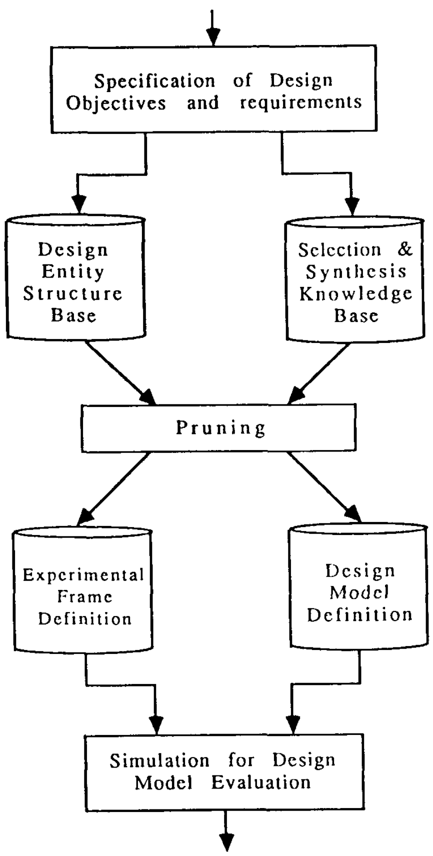

Figure 4. Model-Based Design Process

tation concepts. The representation is a tree-like hierarchical structure used throughout the design process. It is also a domain independent model because the mechanism here can be used in any domain.

5. Model-Based System Design

The term model-based design denotes the use of modelling and simulation techniques to build and evaluate models of the system being designed. As opposed to other approaches that model the design process itself (Nadler, 1981), in this methodlogy facilitates the development of models of design artifacts.

The design process is a series of successive refinements comprising two types of activities (Rozenblit, 1986). The first type of "vertical" activity concerns the specification of design levels in a hierarchical manner. The design levels are successive refinements of the decomposition of the system under consideration. The first, and thus the most abstract level, is defined by the behavioral description of the system. Next levels are defined by decomposing the system into subsystems (modules, components), and applying decompositions to such subsystems until the resulting components are judged not to require further decomposition. At each level, components are classified into different variants. This represents design structural alternatives.

The second type of activities are "horizontal" actions associated with design levels. Such actions include: requirements specification, system functional description, development of design models, experimentation and evaluation via simulation, and choice of design solutions.

The design process should proceed along both "vertical" and "horizontal" axes. The designer structures the designs, explores alternative structures, and derives specifications and models at every level of abstraction.

To appropriately represent the family of design configurations, a structure is needed that embodies knowledge about the following three relationships: decomposition, taxonomy, and coupling. Decomposition knowledge means that the structure has schemes for representing the manner in which an object is decomposed into components. Taxonomic knowledge is a representation for the kinds of variants that are possible for an object, i.e., how it can be categorized and subclassified.

The third type of knowledge that the structure should have is that of synthesis and selection relationships. The synthesis (coupling) constraints impose a manner in which components identified in decompositions can be connected together. The selection constraints limit choices of variants of objects determined by the taxonomic relations.

The methodology for supporting the design process is based on codifying appropriate decompositions, taxonomic, and coupling relationships. In other words, the knowledge about the design domain is modelled by finding pertinent decompositions of the domain, the possible variants that fit within these decompositions, and the constraints that restrict the ways in which components identified in decompositions can be coupled together. This constitutes the declarative knowledge base. Beyond this, procedural knowledge is available in the form of production rules. They can be used to manipulate the elements in the design domain by appropriately selecting and synthesizing the domain's components.

A formal object that meets the requirements stipulated above is the system entity structure (Zeigler, 1984). A system entity structure is a labeled tree with attached variables types. The system entity structure specifies a family of possible design structural configurations. The entities represent system components whose models we aim to build. Aspects and specializations allow designers to specify various design alternatives by selecting alternate components and decompositions. Thus, the system entity structure is a generative scheme from which a set of substructures underlying the construction of various models. The multiplicity of taxonomic relationships in a large design entity structure leads to a combinatorial explosion of possible model alternatives. Therefore, it is necessary to provide procedures that effectively reduce both complexity of the search process for admissible model structures and the size of the of the search space itself. Such procedures have been developed and implemented in the process called pruning (Rozenblit and Huang, 1990).

Pruning the system entity structure results in a set of composition trees (Rozenblit, 1986). A composition tree is a structure that uniquely specifies hierarchical decompositions of design components. The components have no further specializations since they have been selected from taxonomic relationships of the system entity structure. A simulation model is constructed hierarchically by coupling model specifications associated with the nodes of the composition tree.

The final step in the framework is the evaluation of alternative designs. This is accomplished by simulation of models derived from the composition trees. Discrete Event System Specification (DEVS) (Zeigler 1976, 1984) is used as a modeling formalism used for system specification in the methodology. DEVS provides a formal representation of discrete event systems. It is closed under coupling. This property facilitaes the construction of hierarchical DEVS network specifications. A detailed formal treatment of DEVS presented in (Zeigler 1984).

Performance of design models is evaluated through computer simulation in the DEVS-Scheme environment. DEVS-Scheme is an object-oriented simulation shell for modeling and design that facilitates construction of families of models specified in the DEVS formalism. Models are evaluated in respective experimental frames. The experimental frame concept is used to specify an evaluation (by simulation) study. Briefly, an experimental frame defines a set of input, control, output, and summary variables, and input and control trajectories. These objects specify conditions under which a model can be observed and experimented with. Alternative design models are evaluated with respect to experimental frames that reflect design performance questions. Results are compared and traded off in the presence of conflicting criteria. This results in a ranking of models and supports choices of alternatives best satisfying the set of design objectives. For illustration, the design process is shown in Figure 4.

The methodology is being substantiated by case studies involving design and simulation of distributed computer architectures, local area networks, and more recently, VLSI packages.

6.Common Traits in Research of Design Methodologies

Although the design methodologies that are used in various design systems vary in representation methods, creative factor and application domains, there are common traits in the development of engineering design systems.

In the computational model, a tree like structure is employed to represent the design process. In the optimally directed design model, CODESIGNER uses concept network which is a hierarchical representation scheme. The hierarchical representation schemes are also used in knowledge-based creative design model and model-based system design. Because the design process is hierarchical in nature, it must proceed from a top abstract level to lower detailed design levels. Most of the design systems employ a hierarchical representation scheme.

In the design process, creative design is more important than routine design. It can provide a design solution that does not relate to any existing solution. In order to achieve creativity, a knowledge-based system is introduced to exploit knowledge in the engineering design and some methods are needed to manipulate the existing knowledge to produce new features. In the knowledge-based creative design model, mutation and analogy are used to find relations between the new design problem and existing knowledge so as to achieve a creative design. The system entity structure of model-based system design provides a way to manipulate knowledge and thus supports creative design. The CODESIGNER of the optimally directed design model provides a method to derive new features from existing prototypes. Since the computational model is mainly for parameteric design, it is not very suitable for creative design which needs not only change the values of parameters but also change the parameters themselves. To provide a way leading to creative design is a common trend in engineering design system.

As the engineering design problems become large and complex, they can combine many different domains. Therefore it is essential to provide domain-independent methods in the design system. This will become a strong trend in the development of engineering design systems.

REFERENCES

Agogino, Alice M. et al (1989) AI/OR Computational Model for Integrating Qualitative and Quantitative Design Methods, *Proc. of NSF Engineering Design Research Conference*, Amherst, June, pp. 97- 112.

Agogino, Alice M. and A. S. Almgren, (1987) Techniques for Integrating Qualitative Reasoning and Symbolic Computation in Engineering Optimization, *Engineering Optimization*, Vol 12(2), Sept./Oct., pp. 117-135.

Choy, J. K., A. M. Agogino, (1986) SYMON: Automated SYMbolic MONotonicity Analysis System for Qualitative Design Optimization, *Engineering Optimization*, Vol 12(2), Sept./Oct., pp. 117-135.

Dixon, J.R. et al (1989) Computer-Based Models of Design Processes: The Evaluation of Designs for Redesign, *Proc. of NSF Engineering Design Research Conference*, Amherst, June, pp. 491- 506.

Dixon, J.R. et al (1988) Interactive Respecification: A Computation Model for Hierarchical Mechanical System Design *Proc.*

of NSF Engineering Design Research Conference, Amherst, June, pp. 491- 506.

Gero, John S. st al (1989) An Approach to Knowledge-Based Creative Design, *Proc. of NSF Engineering Design Research Conference*, Amherst, June, pp. 333- 346.

Huang, Y. M. (1987) Building An Expert System Shell for Design Model Synthesis in Logic Programming. *Master Thesis*, University of Arizona, Tucson, AZ.

Rozenblit, J. W., (1986) A Conceptual Basis for Integrated, Model-Based System Design, *Technical Report*, Department of Electrical and Computer Engineering, University of Arizona, Tucson, Arizona.

Rozenblit, J. W. and Y. M. Huang, (1987) Constraint- Driven Generation of Model Structures, *Proceedings of the 1987 Winter Simulation Conference*, Atlanta, Georgia.

Rozenblit, J. W. and B. P. Zeigler, (1988) Design and Modeling Concepts, *International Encyclopedia of Robotics Application and Automation*, John Wiley and Sons, Inc., New York.

Rozenblit, J. W. and Y. M. Huang, (1989) Rule-Based Generation of Model Structures in Multifacetted Modeling and System Design, *ORSA Journal on Computing* (in review).

Rozenblit, J. W. and Jhyfang Hu. (1989) Experimental Frame Generation in a Knowledge-based System Design and Simulation Environment, *Modelling and Simulation Methodology:Knowledge System Paradigms*, (M. Elzas et. al., eds), North Holland, Amsterdam.

Rozenblit, J. W. and Y. M. Huang, (1990) Rule-Based Generation of Model Structures in Multifacetted Modelling and System Design, ORSA Journal on Computing (in review).

Rozenblit, J. W., Kim, T. G. and B. P. Zeigler, (1988) Towards an Implementation of a Knowledge-Based System Design and Simulation Environment. *Proc. of the 1988 Winter Simulation Conference*, San Diego. December, pp. 226-230.

Rozenblit, J. W. and Zeigler, B. P., (1988) Design and Modelling Concepts, in: *International Encyclopedia of Robotics, Applications and Automation*, (ed. Dorf, R.) John Wiley and Sons, New York, pp. 308-322.

Rozenblit, Jerzy W. et al, (1989) An Integrated, Entity-Based Knowledge Representation Scheme for System Design. *Proc. of NSF Engineering Design Research Conference*, Amherst, June, pp. 393- 408.

Winston, Patrick Henry , (1984) *Artificial Intelligence*, 2nd ed., Addison-Wesley Publishing Company, MA.

Zeigler, B. P. (1984) *Multifacetted Modelling and Discrete Event Simulation*, Academic Press, London.

Zeigler, B. P. (1987) Hierarchical, Modular Discrete Event Modelling in an Object Oriented Environment. *Simulation Journal*, vol 49:5, pp. 219-230.

Waterman, D. A. and A. Newell, (1971) Protocol Analysis as a Task for Artificial Intelligence, *Artificial Intelligence*, 2, p.285.

Kahn, G., (1985) Strategies for Knowledge Acquisition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No.7, p.511-522.

Kessel, K. L., (1986) Methodological Tools for Knowledge Acquisition, *Proceedings of the 1986 IEEE International Conference on System, Man, and Cybernetics*, Atlanta, GA.

Minsky, M., (1977) *Frame-system Theory, Thinking, eds.* Johnston-Laird and Wason. Univ. Press., London.

Nadler, G., (1981) *Planning and Design Approaches*, John Wiley and Sons, New York.

Newell, A. and H. A. Simon, (1972) *Human Problem Solving*, Englewood Cliffs, NJ: Prentice-Hall.

Nilsson, N. J., (1971) *Problem-Solving Methods in Artificial Intelligence*, New York: McGraw-Hill.

Nilsson, N. J. et., (1987) Intelligent Communicating Agents, *Proceedings of the Second Annual Artificial Intelligence Research Forum*, Palo Alto, California, p.3-17.

Olson, J. R. and Henry H. Rueter, (1987) Extracting Expertise from Experts: Methods for Knowledge Acquisition, *Expert Systems*, August 1987, Vol.4, No.3, p.152-168.

Ohsuga, S., (1984) Conceptual Design of CAD Systems Involving Knowledge Bases, *Knowledge Engineering in Computer-Aided Design*, North-Holland, New York.

Quillian, M. R., (1968) Semantic Memory, *SIP*, pp.216-270.

Rehak, D. R., H. C. Howard, and D. Sriram, (1984) Architecture of an Integrated Knowledge Based Environment for Structure Engineering Applications, *Knowledge Engineering in Computer-Aided Design*, North-Holland, New York.

Ritchie, I. C., (1984) Knowledge Acquisition by Computer Induction, *Proceedings of UNICOM Seminar*, London, England.

Christakis, A. N., Keever, D. B., and j. N. Warfield, (1987) Development of Generalized Design Theory and Methodology, *Proc. of the NSF Workshop: The Study of the Design Process*, Oakland, CA.

Dhar, V. and H. E. Pople, (1987) Rule-Based versus Structure-Based Models for Explaining and Generating Expert Behavior, *Communications of the ACM*, vol.30, no.6, p.542-555.

Ericsson, K. A. and H. A. Simon, (1984) *Protocol Analysis: Verbal Report as Data*, Cambridge, Mass: MIT Press.

Gaines, B. R., (1987) An Overview of Knowledge- Acquisition and Transfer, *Int. J. Man-Machine Studies*, No.26, p.453-472.