# ILL Article Request - GWLA

6/8/2011 12:10:21 AM

Oregon State University
OSU Interlibrary Loan -- Phone: (541) 737-7311 Email: valley.ill@oregonstate.edu Fax: (541) 737-1328 Ariel: 128.193.170.35



**OSU ILLIAD TN#: 511824**

**Journal Title:** Preprints ; NSF Engineering Design Research Conference /
**Volume:**
**Issue:**
**Month/Year:** 1989
**Pages:** 393-408

**Article Author:**
**Article Title:** Rozenblit, J.W. and Hu, J.F. and Huang, Y.M.; An Integrated, Entity-Based Knowledge Representation Scheme for System Design

**Borrower: AZU**
**Lending String:**
*ORE,UIU,AUM,PMC,CUV

**Patron:** Suantak, Liana Son
**Maxcost:** $50IFM

**ILL Number: 78614382**

**Borrower: AZU**

**Odyssey:** 150.135.238.6

**Ariel:** 150.135.238.50

**Email:** askill@u.library.arizona.edu

**Fax:** 520-621-4619

**OCLC Number: 20610358**

## This item will be invoiced at the end of the month

**Photocopy/Loan/Invoice/Postage Charges: Exempt**



**Shipping Address:**
UNIVERSITY OF ARIZONA
LIBRARY - ILL
1510 E UNIVERSITY DR
TUCSON AZ 85720-0055

Pagers:

Initials: ______ NOS: ______ Lacking: ______

BC: Checked Table of Contents: ______ Checked Index: ______

Scanners:

Initials: ______ Date:

# An Integrated, Entity-Based Knowledge Representation Scheme for System Design †

Jerzy W. Rozenblit, Jhyfang Hu and Yueh-Min Huang

Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721

## Abstract

Along with the rising complexity of design problems, design knowledge management, which includes knowledge acquisition, representation, control, and processing, becomes a very difficult task. A good knowledge management scheme is needed to increase the reliability and efficiency of knowledge-based systems. In this paper, an efficient representation scheme for system design applications called Frames and Rules Associated System Entity Structure (FRASES) is introduced. With FRASES, complex engineering design knowledge is organized into a hierarchical, entity-oriented tree structure that facilitates control and processing of knowledge. Exploiting well-defined axioms and operations of FRASES, the knowledge acquisition process accomplished conventionally with time-consuming interviews can be improved.

## 1. Introduction

In the last decade the technology of knowledge-based systems has been widely used in solving various engineering problems such as system diagnosis, production scheduling, capacity planning, operation monitoring, design and synthesis, and performance evaluation. The expected contributions of knowledge engineering to CAD/CAM are the integration of various complex components and the construction of an efficient model for the design (Ohsuga, 1984). In order to be successful in these efforts, design knowledge has to be organized in such ways that it can be manipulated effectively and efficiently. In general, the performance of a knowledge-based system is determined by its knowledge management scheme which includes techniques for knowledge acquisition, representation, and inferencing. Unfortunately, the strategies for knowledge management are usually application-dependent. Up to now, there is no universal method of knowledge management which can cover all the diverse design knowledge. To assure high reliability and efficiency of knowledge-based systems, system designers are responsible to identify the characteristics of a design application and select the most appropriate knowledge management scheme.

## 2. Model-Based System Design

Our research applies modelling and simulation concepts to unify engineering design activities and develop a methodology for systematic design model construction and evaluation. We recognize the multiplicity of objectives and requirements as a sine qua non condition of the modern engineering design process. Therefore, design objectives play a fundamental role in driving design structuring, i.e., the specification of a family of design structural configurations

[0] †Supported by NASA-Ames Cooperative Agreement No. NCC 2-525.

(Rozenblit, 1986). Furthermore, the objectives guide the design model building process and specification of circumstances under which design models are experimented with and evaluated.

The term model-based design denotes the use of modelling and simulation techniques to build and evaluate models of the sy  m being designed. As opposed to other approaches that model the design process itself (Christakis et.al., 1987; Nadler, 1981), we develop models of design artifacts.

We consider the design process as 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 we also classify components into different variants. This represents design 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 should be able to structure the designs, explore alternative structures, and derive specifications and models at every level of abstraction.

## 2.1 Formal Concepts for Model-Based Design

To appropriately represent the family of design configurations, we need a structure 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. By taxonomic knowledge, we mean 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 our 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, we seek to model the knowledge about the design domain 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, we must provide procedural knowledge in the form of production rules which 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.

When a variable V is attached to an item occurrence I (node I), this signifies that a variable I.V may be used to describe the item occurrence I. The structure satisfies the axioms of:

1. alternation of entity/aspect and entity/specialization
2. strict hierarchy
3. uniformity
4. valid brothers
5. attached variables types

and allows for the following operations:

1. naming scheme
2. generation distribution/aggregation relations
3. transformation to taxonomy free form
4. pruning for design configurations
5. attachment of coupling and selection constraints

For full explication of entity structure axioms and its properties the reader is referred to (Zeigler, 1984).

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 the 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, 1987).

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 hierarchical model specification is described in detail in (Zeigler, 1984).

The final step in our framework is the evaluation of alternative designs. This is accomplished by simulation of models derived from the composition trees. We use the experimental frame concept to specify a 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 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 1.

In this paper, we describe an effort to refine the system entity structure representation

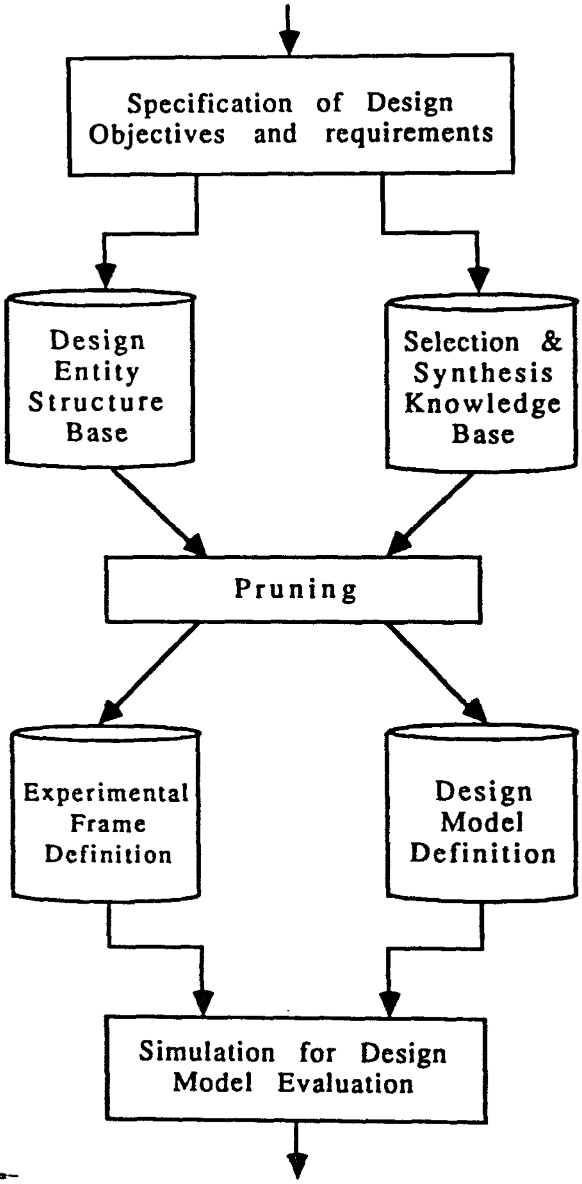



Figure 1 Model-based system design process

and combine it with a frame-base scheme. The integration results in a knowledge representation called Frames and Rules Associated System Entity Structure (FRASES) for engineering design applications (Hu et.al., 1989). The main objective of FRASES is to facilitate the knowledge management task in knowledge-based engineering design systems. The discussion of this paper will focus on issues of representation, acquisition, refinement, and inferencing.

## 3. Design Knowledge Representation

Knowledge-based systems are becoming increasingly useful in solving complex design problems. To support engineering design effectively, a knowledge-based system should be capable of encompassing both static and also dynamic characteristics of the design. Application-independent schemes should be available to govern design provisions, to know what to do, or how to change the design so that the provisions will be satisfied (Rehak, 1984). Consequently, a qualified knowledge representation has to facilitate both static and dynamic knowledge representations, which can be classified as follows:

- Static Knowledge
  1. structural characteristic of objects
  2. taxonomy/decomposition of objects
  3. constraints and rules for design selection/synthesis
- Dynamic Knowledge
  1. functional characteristics of objects
  2. procedures for generating design alternatives
  3. procedures for design verification and evaluation

Such a scheme should also facilitate knowledge management within the system. For example, it should assist in efficient knowledge acquisition, knowledge inferencing, and decision making. Finally, knowledge reflected by the representation scheme must be transparent to domain experts, knowledge engineers, and system users.

Increasing demand on the quality of knowledge-based systems has resulted in knowledge representation becoming a major topic in AI research. Different schemes such as production rules (Newell and Simon, 1972), frames (Minsky, 1977), structure models (Dhar, 1987), semantic networks (Quillian, 1968), AND/OR trees (Nilsson, 1971), and system entity structure (Zeigler, 1984; Rozenblit and Huang, 1987) have been defined for representing knowledge. These conventional representation methods often do not provide enough expressive power when applied to engineering design domains individually. For example, the frame representation is suitable for the problem domain of classification, whereas production rules are preferred in diagnosis domains. Based on this observation, in previous work we have successfully applied production rules to generate a design model represented in the system entity structure and created a hierarchical knowledge base (Huang, 1987). This was our first step toward the integration of different knowledge representations. Furthermore, by explicitly defining the relationships between objects and taxonomies in the system entity structure, we have incorporated the concept of inheritance into the system entity structure.

are preferred in diagnosis domains. Based on this observation, in previous work we have successfully applied production rules to generate a design model represented in the system entity structure and created a hierarchical knowledge base (Huang, 1987). This was our first step toward the integration of different knowledge representations. Furthermore, by explicitly defining the relationships between objects and taxonomies in the system entity structure, we have incorporated the concept of inheritance into the system entity structure.

To further unify the above descriptions, we have developed an integrated knowledge representation scheme—FRASES. It is a scheme that combines an entity-based representation with production rules and frames. FRASES is a superclass of the system entity structure that encompasses the boundaries, decompositions, and taxonomic relationships of the system being designed. All the axioms (i.e., Uniformity, Strict hierarchy, Alternating mode, Valid brothers, Attached variables) and operations (i.e. Naming scheme, Distribution and Aggregation, Transformation, Pruning, Inheritance) defined for the system entity structure (Zeigler, 1984) are also valid for the FRASES–based representation.

An entity in FRASES signifies a conceptual part of the system which has been identified as a component in one or more decompositions. Each such decomposition is called an aspect. In addition to decompositions, there are relations that facilitate the representation of variants for an entity. Called specialized entities, such variants inherit properties of an entity to which they are related by the specialization relation. Each node of a FRASES tree is associated with a cluster of knowledge, termed Entity Information Frame (EIF). An Entity Information Frame (EIF) is a frame object (Winston, 1984) containing the following variable slots:

$$< M,\ ATTs,\ DSF,\ ESF,\ CRS,\ CH >$$

where

M: is the name of associated entity or model
ATTs: are design attributes and parameters of M
DSF: is the design specification form
ESF: is the experiment specification form
CRS: are constraint rules for pruning and design synthesis
CH : children entities of M

With FRASES representation, behavioral characteristics of objects are described by simulation models defined in the model base. M represents the key to access a model of the entity to which EIF is attached.

ATTs are attributes or parameters used to characterize the associated object. Attributes of an entity are partitioned into two groups, e.g. static and dynamic (Rozenblit and Hu, 1989). Static attributes are variables used to describe an object's general information. For example, typical static attributes for a microprocessor are a source, chip size, technology type, and packaging style. Static attributes are usually instantiated with values (i.e., quantitative or qualitative) while constructing the FRASES structure. During application, static attributes are directly referred to by passing frame messages. They do not require further computation during dynamic design evaluation by a simulation study. On the other hand, dynamic attributes are related to design details and/or behavioral characteristics of objects. To ease processing, dynamic attributes are further classified into performance attributes and design parameters.

design parameters for a microprocessor are datawidth, on- chip cache size, and cache hit-ratio. Quantitative functions, heuristic rules, and CAD tools should be associated with the FRASES system to provide rough design estimates and analysis. Design rules derived from architectural, algorithmic, or technical constraints may also be included in FRASES to make adjustments of design parameters.

Design Specification Form (DSF) is a slot used to accept the user's design specification of objectives, constraints, and criteria weighting scheme. The contents of DSF define the system requirements that must be satisfied by the system being designed. Each entity of FRASES may have its own DSF so that design specification can be described in a hierarchical manner.

Experiment Specification Form (ESF) is used to accept the specification of simulation requirements such as an arrival process, service process, and simulation controls. ESF provides information to direct the automatic construction of experimental frames. Again, ESF is placed with the entity nodes of a composition tree. Multiple ESF specifications will result in a distributed experimental frame organization. For illustration, a typical DSF specification for the processor of a personal computer is shown in Figure 2. With the specified DSF, the processor being designed should be capable of executing 10 Million Instructions Per Second (MIPS), the cost of the processor is less than 300 dollars, and the power consumption of the processor must be less than $0.5\mu$ watts. Figure 3 shows a simplified experiment specification form for the processor. This simulation experiment defines that the input arrival rate for the first 100 events will follow the Poisson distribution with the mean value of 10. For subsequent events, the normal distribution with mean 1 will be employed. Each event is composed of a symbolic identification and numerical workload. Simulation will be executed for 300 events. For each 50 system time units, the measurement of performance indices must be reported.

Constraint Rules (CRS) slot contains pruning and synthesis knowledge for generating a system configuration. Selection constraints for pruning alternatives are associated with specialization nodes. Constraints for synthesizing components are associated with the aspect nodes.

Children (CH) indicates the children nodes of the entity.

A simple FRASES structure for representing the design of a personal computer is shown in Figure 4. As shown in the figure, each FRASES entity is associated with an Entity Information Frame (EIF). With EIF, both declarative and procedural knowledge about the entity are appropriately represented.

The following features distinguish FRASES from other representation schemes:

- Implicitness of Knowledge: FRASES is a generative scheme capable of representing a family of possible design configurations. Alternative designs are constructed by a set of procedures operating on objects and their associated frames.
- Reducing Complexity of a Problem: FRASES employs a top-down design methodology to describe knowledge from an abstract level to more specific levels in a hierarchical manner. For a complex problem domain, this kind of a problem-reduction technique results in decreasing complexity of the knowledge manipulation. This representation method also benefits the knowledge-based design process by increasing its modularity and completeness.
- Uniformity of the Knowledge Base: The characteristic of inheritance and uniformity highly

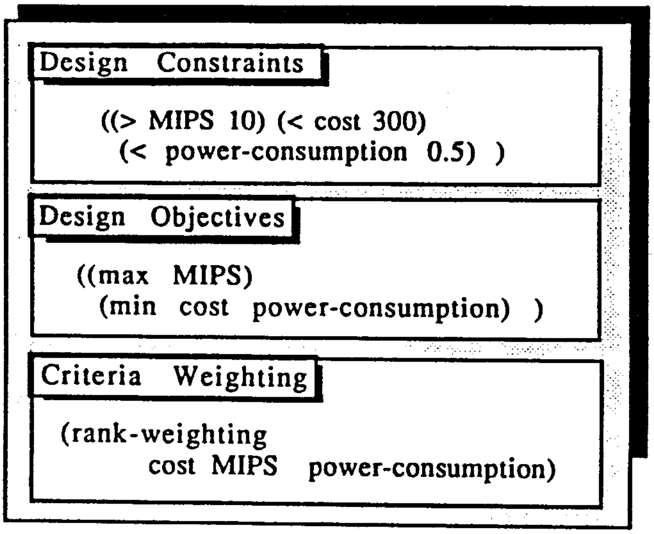



Figure 2 Design specification form (DSF)

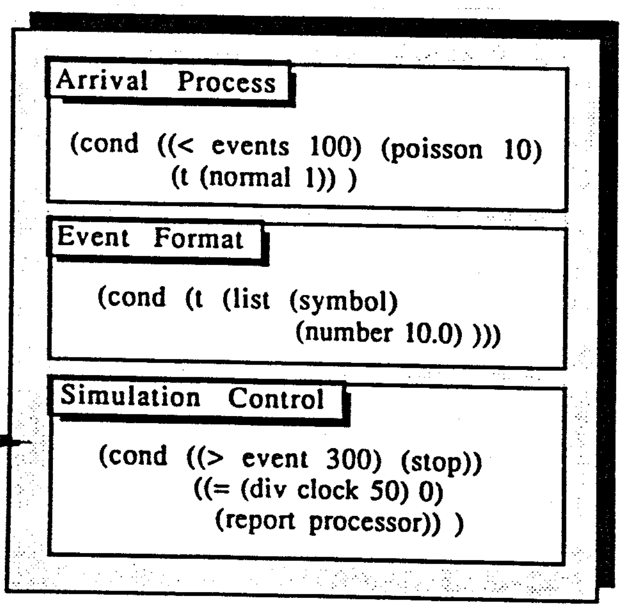



Figure 3 Experiment specification form (ESF)

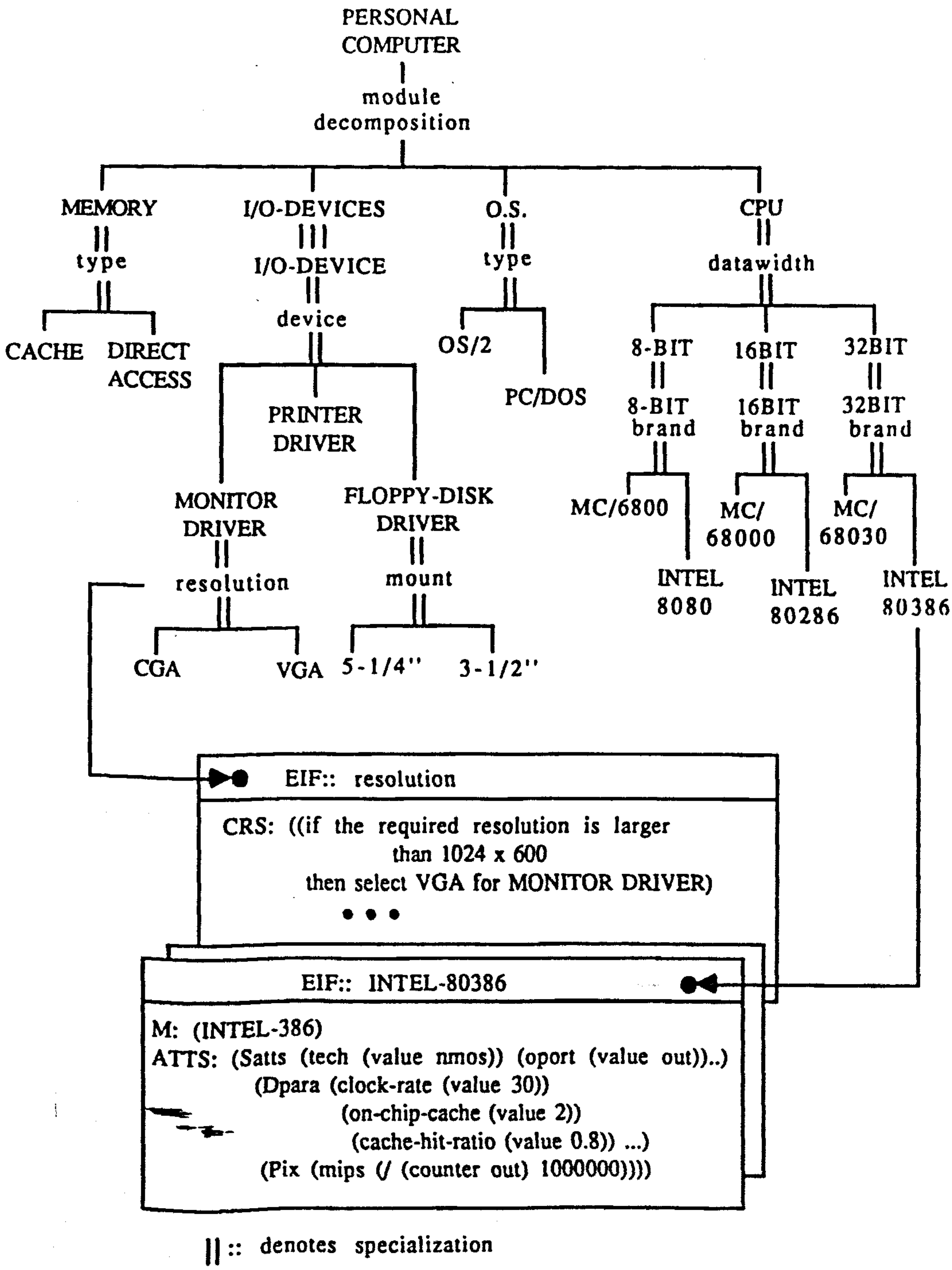



Figure 4 PC representation with FRASES

reduces the size of a knowledge base required for the same design application. In FRASES, all the attached attributes and substructures are inherited through the specialization of an entity. Every occurrence of an entity has the same Entity Information Frame and isomorphic substructures. Identical nodes located in different paths are updated automatically according to the axiom of uniformity. This eliminates duplicate descriptions for the same design components.

- Hierarchical Organization of the Knowledge Base:

  Hierarchical design is important in complex system design problems but unfortunately most currently available knowledge- based systems seem to be "flat" (Nilsson, 1987). By employing FRASES representation, knowledge base is organized as a hierarchical tree structure. Each node of the tree contains a frame which possesses a cluster of knowledge for managing its subtree. This hierarchical knowledge concept facilitates knowledge validation, inferencing, and refinement tasks in the knowledge base.

- Flexible Refinements of the Knowledge Base: It is not possible to acquire all of the expert's knowledge for a specific problem domain at one time. The knowledge base needs to be updated until the information embedded in it closely resembles the actual knowledge of experts. With FRASES, the hierarchy of knowledge organization facilitates refinements in depth (or levels of abstraction) and breadth (or decomposition details). Furthermore, the axiom of uniformity allows all modifications to be updated simultaneously for all other identical nodes distributed in the FRASES tree.

- Verification of the Knowledge Base: With the growing size of the knowledge base, it becomes difficult to validate all reasoning paths. The hierarchy and entity basis of FRASES would reduce potential "gaps" in a knowledge base. In FRASES, heuristic rules are hierarchically distributed. Each rule deals only with the knowledge about its subtree nodes. The characteristic of rule locality in FRASES facilitates validation of the knowledge base.

## 4. Knowledge Acquisition Based on Representation

Although a number of methodologies such as interviewing, protocol analysis, observing, induction, clustering, prototyping (Waterman, 1971; Ericsson and Simon 1984; Ritchie, 1984; Kahn, 1985; Kessel, 1986; Gaines, 1987; Olson, 1987) etc., have been proposed for knowledge acquisition, it is difficult to demonstrate their efficiency in engineering design applications. Different design applications require different strategies for knowledge acquisition and representation to avoid misunderstanding and/or loss of important knowledge from a human expert. It is very difficult to acquire all aspects of design knowledge simply via the question/answer elicitation process. For example, the knowledge engineer may be unable to pose all relevant questions pertaining to a design application; or the knowledge to be acquired may be too complicated to be asked in questions, or the human expert may misunderstand the knowledge engineer's questions. All the above situations may result in unnecessary, duplicate, and conflicting knowledge. Acquiring complex knowledge with conventional acquisition methods is costly due to preparation, verification, organization, and translation of the information acquired from experts. Knowledge acquisition should be directed or supervised under a certain scheme. The scheme should help in: acquiring knowledge, detecting conflicts, identifying missing facts, and eliminating duplicate or redundant knowledge.

FRASES is a complete representation scheme for engineering design application which conveys not only the static knowledge but also the dynamic knowledge required for design applications. By using FRASES, the complex knowledge acquisition task can be automated. At each design level, question patterns about decomposition and taxonomic knowledge of design objects can be generated automatically. If an Entity Information Frame (EIF) is missing, this fact will be detected and signaled to design experts. Appropriate verification procedures will be integrated for axiom examination and elimination of conflicting or duplicate knowledge. Acquired knowledge will be automatically translated into internal representation without human intervention. Knowledge acquisition assisted or activated by following the FRASES organization is called the Knowledge Acquisition based on Representation (KAR) (Hu and Rozenblit, 1989). A simplified KAR approach used to acquire knowledge for a personal computer design is illustrated in Figure 5. At each iterative acquisition, related query rules are referred to and interpreted based on the structural nature of FRASES to generate question patterns. To assure the consistency of knowledge, information provided by users on each query cycle is automatically validated with a variety of verification rules. Several advantages are expected from the using the KAR approach:

- Efficiency: Questions patterns necessary to acquire design knowledge for decomposition, taxonomy, pruning, and synthesis of systems are first predefined into query rules. Appropriate questions can be automatically generated to query knowledge. KAR approach directly translates the acquired knowledge into FRASES representation. By exploiting the defined axioms and operations, the time-consuming acquisition task can be accomplished efficiently.

- Flexibility: Simple modification of FRASES will fit other AI research applications. For example, in object classification FRASES can be modified into a pure specialization structure with selection constraints associated with each specialization entity.

- Controllability and Observability:

  FRASES will be implemented using extensive graphic facilities. The entity-oriented, hierarchical structure of FRASES allows the represented knowledge to be updated easily.

- Cost-Effectiveness: Unlike the conventional approaches which always require human intervention in knowledge acquisition, verification, translation, and organization, the knowledge acquisition task can be automated with KAR. The fast turnaround of knowledge acquisition highly reduces the development cost of knowledge-based systems.

## 5. Design Process with FRASES

After the design knowledge is built into FRASES, the design process is aided in the following way. A set of design objectives and constraints are accepted through the Design Specification Form (DSF) interface. These user-specified requirements and constraints are employed to derive design configurations. From the view point of problem-solving, aspect and specialization nodes of the system entity structure are the states of the solution space. The process of design generation can be interpreted as a search directed by constraints associated with these states. The mechanism which drives this search is called pruning (Rozenblit and Huang, 1987). The resultant design is accomplished by forming a path made of these states through a process of analysis, synthesis, and evaluation. Production rules direct the search. The type of rules can be either selection or synthesis. This depends on the type of a node. The synthesis rules are associated with decomposition nodes. The selection rules are associated with specialization nodes (Rozenblit and Huang, 1987, 1989).

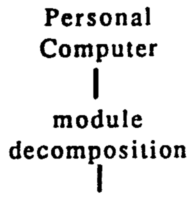
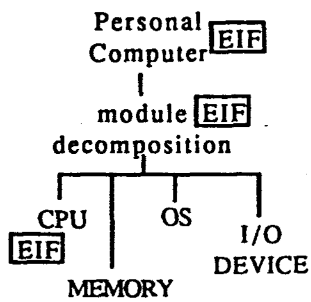
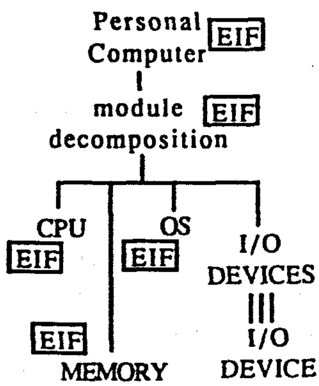
| KAR/EXPERT INTERACTION | FRASES CONVERSION |
| --- | --- |
| KAR> What is your design problem domain?<br>=> personal-computer<br>• • • | Personal Computer |
| KAR> Can you classify a Personal-Computer based on certain specialization?<br>=> why<br>KAR><br>-> This question is used to query how experts<br>-> classify variants of an entity. For example,<br>-> a local area network can be classified into<br>-> RING, BUS, and TREE based on Topology.<br>• • •<br>KAR> Can you decompose a Personal-Computer based on a certain aspect?<br>=> module-decomposition<br>• • • |   |
| KAR> What are the subcomponents when you decompose a Personal-Computer based on the module-decomposition?<br>=> cpu memory os i/o-device<br>• • •<br>KAR> What kind of performance indices will be considered to evaluate the CPU?<br>=> mips<br>• • • |   |
| KAR> Can the number of I/O devices vary with design requirements?<br>=> yes<br>• • • |   |

Figure 5 KAR with FRASES for PC design

Design constraints are classified into two categories, static and dynamic (Rozenblit and Hu, 1989). Although we use dynamic constraints in the pruning process (e.g., we may specify the following rule: "if desired processing speed is high then select Intel 80386 CPU") their satisfaction must be verified by either analytical or simulation methods. This allows us to establish conformity of the design model to the design specifications.

The pruning inference engine derives one design model recommendation at a time. To save time, all possible design configurations should be generated all at once. To assure the efficiency and reliability of design reasoning, the Weight-Oriented FRASES Inferencing Engine (WOFIE) (Hu 1989) was developed to derive all possible alternative design models at once. By associating appropriate weight information with each entity, WOFIE is able to perform design reasoning with a flexible design methodology (e.g., top-down, bottom-up, or mixed).

Consider the design of personal computers in Figure 4. If design constraints are specified as shown in Figure 6a, two alternative design structures (Figure 6b) will be pruned from the original FRASES in Figure 4. To select the best design structure, simulation and trade-off design evaluation using multi-criteria decision making will be activated to suggest the most appropriate design based on the criteria weighting specified in the Design Specification Form (Rozenblit and Hu, 1989).

Performance of design models in our framework is evaluated through computer simulation in DEVS-Scheme environment (Zeigler 1987, Rozenblit et. al. 1988). DEVS-Scheme is an object-oriented simulation environment for modeling and design that facilities construction of families of models in a form easily reusable by retrieval from a model base. Models are evaluated in respective experimental frames. As we indicated in Section 2.1, an experimental frame defines a set of input, control, output, and summary variables. Those objects specify conditions under which a model is simulated and observed. The environment supports construction of distributed, hierarchical discrete event models and is written in the PC-Scheme language which runs on IBM compatible microcomputers and AI workstations.

## 6. Summary

Because knowledge-based systems explicitly represent and reason with knowledge supplied by human experts, they offer considerable promise in modern engineering design. This paper has presented a knowledge management scheme used in our system called Knowledge-Based Design Support Environment (KBDSE) (Rozenblit and Hu, 1989). A flexible and efficient knowledge representation scheme called FRASES was introduced for representing complex modern engineering design knowledge. The hierarchical, entity- oriented FRASES representation not only facilitates the knowledge acquisition but also eases control and processing of design knowledge. Knowledge management with FRASES, the time and cost spent in acquisition, verification, translation, organization and application are highly reduced.

A.)

- OS.process = multi-tasking
- Memory.performance = high
- Monitor-Driver.resolution > 1024 x 600
- CPU.clock-rate > 12 MHz
- Floppy-Disk.mount < 4"

B.)

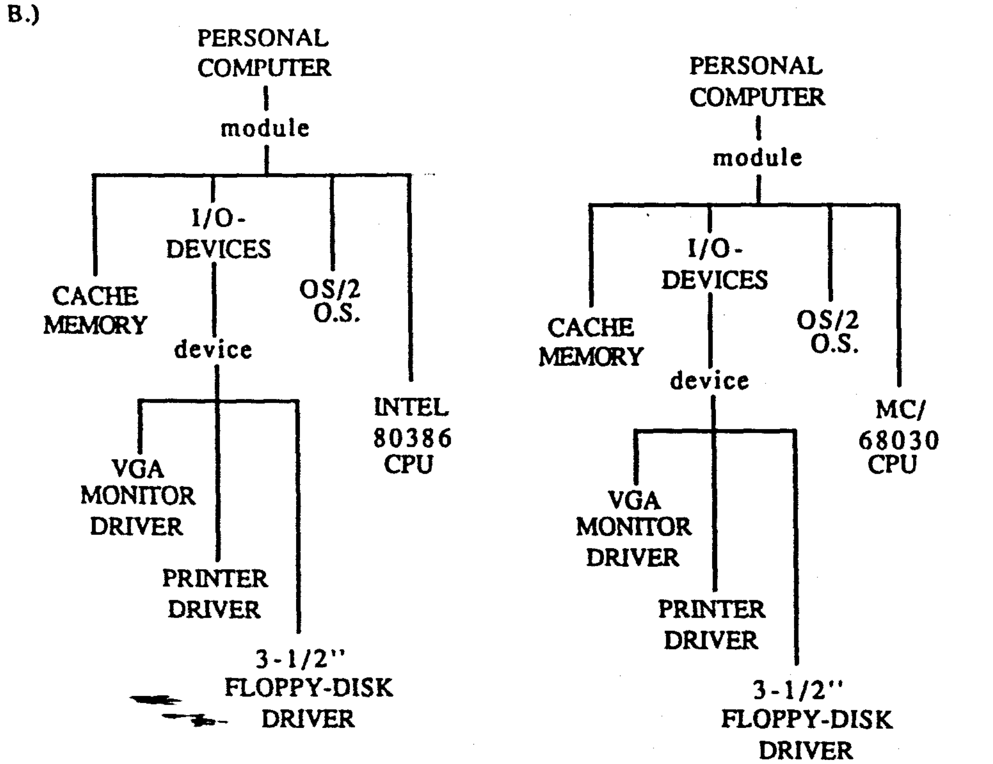



Figure 6 Pruning of PC design

## References


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.

Hu, J., Huang, Y. M., and J. W. Rozenblit 1989. " FRASES - A Knowledge Representation Scheme for Engineering Design", *Advances in AI and Simulation*, SCS Simulation Series, vol.20, no.4, p.141-146.

Hu, Jhyfang and Jerzy W. Rozenblit 1989, "Knowledge Acquisition Based on Representation (KAR) for Design Model Development," to appear in: *Knowledge-Based Simulation: Methodology and Applications*, (ed. P. Fishwick and R. Modejeski), Springer-Verlag, NY.

Hu, Jhyfang 1989, "Towards a Knowlege-Based Design Support Environment for Design Automation and Performance Evaluation," Ph.D. Dissertation, University of Arizona, Tucson, Arizona.

Huang, Y. M. 1987. "Building An Expert System Shell for Design Model Synthesis in Logic Programming", *Master Thesis*, Dept. of Electrical and Computer Engineering, Univ. of Arizona, Tucson,Arizona.

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" In *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," In *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.

Rozenblit, J. W., 1986. "A Conceptual Basis for Integrated, Model-Based System Design", *Technical Report*, Dept. of Elec. and Computer Engineering, Univ. of Arizona, Tucson, AZ.

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

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 Y. M. Huang 1989. "Rule-Based Generation of Model Structures in Multi-facetted Modeling and System Design," *ORSA Journal on Computing* (in review).

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

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

Zeigler, B. P. 1984. *Multifaceted Modelling and Discrete Event Simulation*, Academic Press, London.