Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

THE ROLE OF KNOWLEDGE MANAGEMENT IN HIERARCHICAL MODEL DEVELOPMENT

Yueh-Min Huang, Jerzy W. Rozenblit and Jhyfang Hu
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721

ABSTRACT

The methods for transforming real-world problem into simulation models are being increasingly explored with the availability of inexpensive computing power. In general, traditional model building procedures involve a lengthy problem formulation and interaction between the analyst and the client. Furthermore, after the model definition is arrived at, the simulation model is often programmed manually. Recent developments in simulation modeling have focused on automatic model generation employing artificial intelligence (AI) techniques. Such developments focus on the transfer of the user's knowledge about the system into an executable simulation model. Current techniques still lack effective knowledge acquisition tools and a global database from which model alternatives can be generated. In this paper, a set of knowledge bases (KBs) will be proposed to aid in the hierarchical model construction.

1. INTRODUCTION

The schemes for modeling a real-world object are usually problem domain dependent. Different problem domains require different modeling schemes. Traditional approaches often involve a lengthy model formulation phase. This phase include time-consuming interactions between the analyst and the client, manual programming for executable model, visual simulation with model validation, and statistical packages with analysis of results (Doukidis and Paul, 1985). With the increase of problem size, traditional approaches are becoming very complex. Along with the availability of inexpensive computing power, computer-aided model development is in a period of significant transition. Recent developments in computer-aided systems embed Artificial Intelligence (AI) and Expert Systems (ESs) into the Simulation Modeling.

Both AI and ESs create programs which simulate domain experts to achieve some tasks such as giving expert advice, proving theorems, and understanding natural language. These techniques can be applied in information management for the model formulation phase. Current implementations include Knowledge-Based Model Construction (KBMC), Computer Aided Simulation Model (CASM), Knowledge-Based Simulation System (KBS) (Murray and Sheppard, 1988; Balmer and Paul, 1986; Reddy et al., 1986). Murray and Sheppard give a description of how to automate model construction using domain and modeling knowledge and implement KBMC. CASM is a project developed at the London School of Economics (Balmer and Paul, 1986). It is used to investigate the ways of making the process of simulation modeling more efficient using a natural language understanding system (Balmer and Paul, 1986). Reddy et al. (1986) describe a knowledge-based simulation system (KBS) by employing AI-based knowledge representation system for modeling. The system provides several functions such as interactive model creation and alteration, simulation monitoring and control, and graphics display. For other surveys, we refer readers to (Balci and Nance; 1987).

In our previous work, we have implemented a model development environment called MODSYN (MODel SYNthesizer) (Huang, 1987; Rozenblit and Huang, 1987). This environment is used to generate a model structure based on a set of modeling objectives and requirements expressed as production rules. MODSYN supports the hierarchical model development process. The model representation in MODSYN is the system entity structure (Huang, 1987; Zeigler, 194). Recently, we have augmented the system entity structure by combining the frame and production rules formalisms into a new representation scheme called FRASES (Hu et al., 1989). We have also proposed a knowledge acquisition process called KAR (Knowledge Acquisition based on Representation) to support model development using FRASES (Hu and Rozenblit, 1989).

2. MODEL KNOWLEDGE ACQUISITION AND REPRESENTATION

Applications of AI in simulation modeling emphasize how to build a simulation model without the knowledge of a domain simulation language. In other words, they focus on how to transfer user's knowledge about a system and its specifications into executable simulation code. Current implementations of simulation model generation still lack: 1) an effective knowledge acquisition tool to direct modellers in building complete simulation models for a complicated system; 2) a global database (or framework) to generate possible models. (A global database is quite practical to organize time-dependent data so that it can keep generated models up-to-date or can retrieve new models by change without going through the model generation process again); 3) organization of a hierarchical model abstraction which allows models to be described at any desired level of detail.

Thus, model representation plays an important role in the life cycle of model development. Here, we use FRASES as a framework for hierarchical model representation. We propose the following knowledge bases (KBs) to aid in the automatic model construction:

- **Meta-Acquisition Knowledge Base (KAKB):** This knowledge base will direct users in building models by using the problem-reduction principle (Nilsson, 1971). KAKB will question users about possible decompositions of components, their taxonomies as well as constraints on their couplings. We term this acquisition process Knowledge Acquisition Based on Representation (KAR) (Hu and Rozenblit, 1989).
- **Global Knowledge Base (GKB):** This KB will store a family of models generated by Meta-Acquisition KB. Models will be selected from GKB based on one specification of model parameters and attributes.
- **Construction Knowledge Base (CKB):** This is a static knowledge base that will store modeling knowledge for the conversion of a model specification into simulation code.
- **Validation Knowledge Base (VKB):** This base will contain rules for model validation after relevant data have been collected from simulation runs.

The four knowledge bases for configuring the life cycle of automatic model development environment are shown in Figure 1.

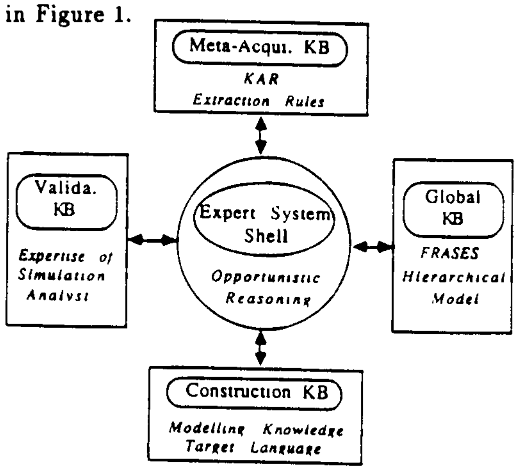


Figure 1. Four KBs with an Opportunistic Reasoning Engine

2.1 FRASES Representation

To improve the model specification process and unify knowledge representations employed in the framework (i.e., the semantic net-like system entity structure and production rule formalism), we have developed an integrated representation scheme called FRASES (Rozenblit at.al.). FRASES combines an entity-based representation with production rules and frames. FRASES is a supercalss of the system entity structure. Although developed for model based system design, the scheme is generic and particularly suitable for hierarchical modeling.

Each entity node of a FRASES tree has a cluster of knowledge, termed Entity Information Frame (EIF) associated with it. An Entity Information Frame (EIF) is a frame object (Winston, 1984) containing the following slots:

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

Where

M: the name of associated nodes
ATTs: design attributes and parameters of M
DSF: the desig specification form
ESF: the experimental specification form
CRS: constraint rules for pruning and model synthesis
CH: children entities of M

With FRASES representation, behavior characteristics of objects are described by simulation models defined in the model base. M represents the key to access the model of an entity to which the EIF is attached. ATTs are attributes or parameters used to characterize the associated objects. 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 such as an arrival process, service process, and simulation controls. ESF provides information to direct the automatic construction of simulation experiments (experimental frame).

Constraint Rules (CRS) slot contains pruning and synthesis knowledge expressed as production rules for generating a configuration of model components. Selection constraints for pruning alternatives are associated with specification nodes. Constraints for synthesizing components are associated with the aspect nodes. Children (CH) indicate the children nodes of the entity. An example of a FRASES structure is presented in Figure 2.

The following features distinguish FRASES from other representation schemes:

- **Generative Model Structure Knowledge:**

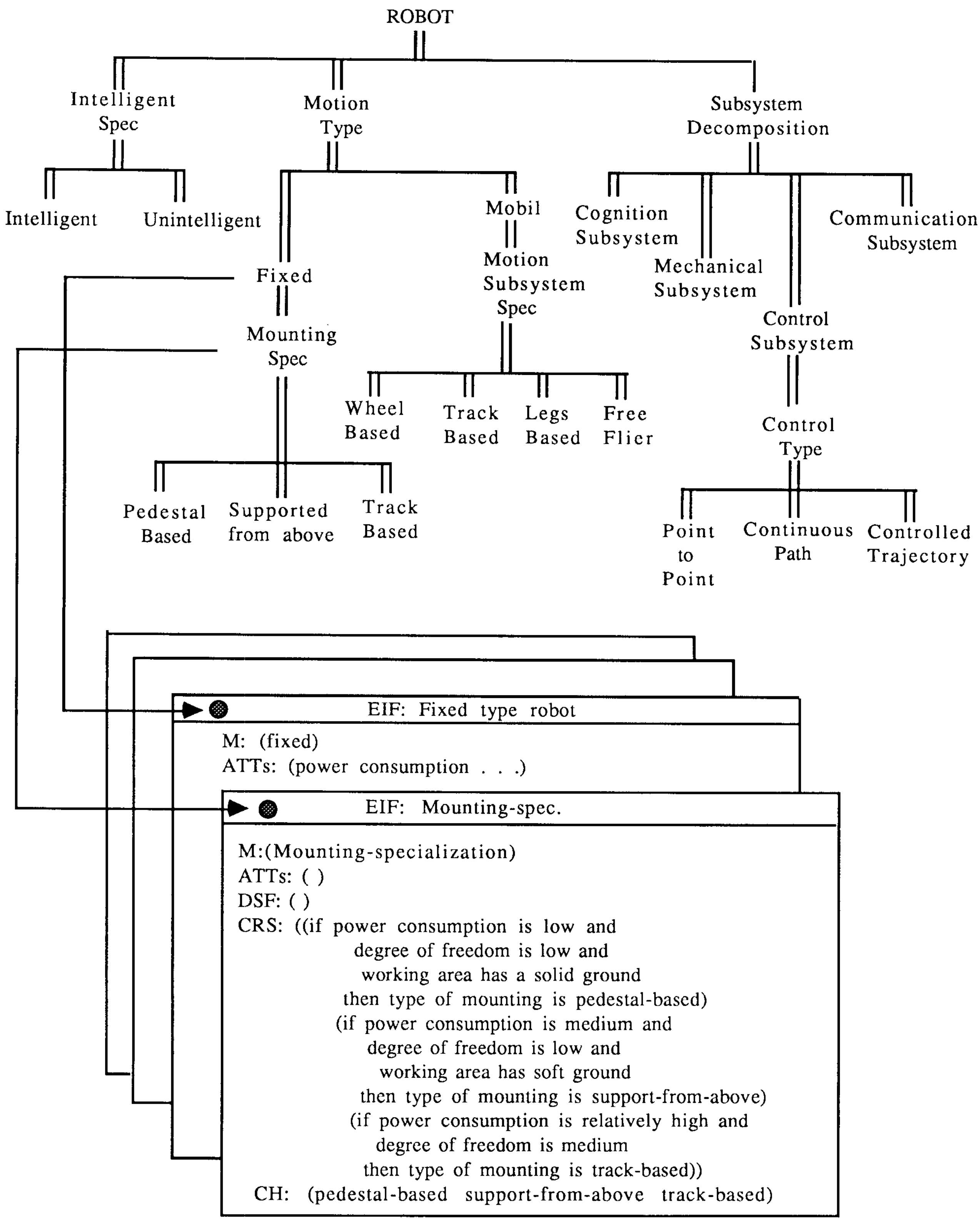


Figure 2 Hierarchical Robot Model in FRASES

FRASES is a generative scheme capable of representing a family of model structures.

- **Hierarchical Organization of the Knowledge Base:** FRASES employs a top-down methodology to describe knowledge from an abstract level to more specific levels in a hierarchical manner. This approach reduces the complexity of knowledge manipulation and improves knowledge completeness and representation of modularity.
- **Uniformity of Knowledge Base:** The characteristic of inheritance and uniformity reduces the size of the knowledge base. In FRASES, all the attached attributes and substructures are inherited through the specification 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 of the same model objects.
- **Incremental Refinement of the Knowledge Base:** In simulation studies the knowledge base needs to be updated incrementally as new results are analyzed. 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 and Validation of the Knowledge Base:** The hierarchy and modularity of FRASES reduced 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 verification and validation of the knowledge base.

2.2 Modeling 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 simulation modeling applications. Different applications require different strategies for knowledge acquisition and representation to avoid misunderstanding and/or loss of important knowledge from a human expert. Acquiring complex knowledge with conventional acquisition methods is costly due to preparation, verification, organization, and translation of the information elicited 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.

A FRASES-based approach for modeling knowledge acquisition (Hu and Rozenblit 1989) is currently being formulated. Termed KAR (Knowledge Acquisition Based on Representation), the method consists in generating question patterns about decomposition and taxonomic relationships of the model objects. At each iterative application, domain relevant query rules are referred to and are 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 verification rules. A KAR example is given in Figure 3.

Several advantages are expected from the application KAR approach to modeling knowledge acquisition:

- **Efficiency:** Questions patterns necessary to acquire knowledge for decomposition, taxonomy, pruning, and synthesis of systems are expressed in the query templates. Appropriate questions can be generated automatically and directly translated into FRASES representation.
- **Universality:** FRASES is applicable to any modeling domain which can be structured using the system entity structure methodology.
- **Cost-Effectiveness:** Conventional approaches require human intervention in knowledge acquisition, verification, translation, and organization. Automating the knowledge acquisition task with KAR reduces the development cost of knowledge-based modeling support systems.

3. MODEL GENERATION

After the process of model acquisition, a family of models generated by Meta-Acquisition KB will be stored in the Global KB. These candidate models are represented by FRASES representation and can be later manipulated by an inferencing mechanism according to user's specifications. In our previous work, we developed an inferencing mechanism using backward reasoning strategy in the MODSYN environment (Rozenblit and Huang, 1987). MODSYN shell later incorporated forward reasoning and was rewritten in Common Lisp (Pan, 1989). Both reasoning strategies are optional in running the latest version. In the following, we propose an adaptive inferencing mechanism to make model generation more efficient.

The model generation based on FRASES representation can be regarded as the pruning process with respect to the knowledge sources and external specifications. A problem-solving model is required for organizing reasoning steps and domain knowledge to construct a solution for a problem. In a production system,

Expert/Question Pattern Generator Interaction

1.) What is your design domain (a list)?
=> (robot)

2.) Can you classify robots based on certain specialization (a list)?
=> (intelligence motion)

3.) Please specify selection rules for distinguishing intelligent and motion robots (IF-THEN clause)

=> (if desired autonomy is not "don't care"
then robot is determined by intelligent type)
(if desired autonomy is "don't care"
then robot is determined by motion type)

==> You already specified two rules for them based on autonomy. Any other rules for them (y/n)?

===> n

4.) What are the components of a robot (a list)?
=> (Cognition subsystem
Mechanical subsystem
Control subsystem
Communication subsystem)

==> You have specified four entities for robot decomposition. Any multiple decomposition for above entities (y/n)?

===> n

5.) Is there any further specialization for intelligent robot?
=> yes
Please specify them (a list)

==> (intelligent unintelligent)

.
.
.

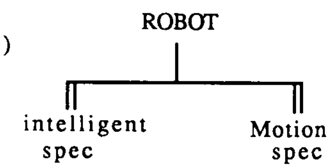
{a crtl key was entered to indicate no more construction}

n.) Any modification for this model (y/n)?
=> n

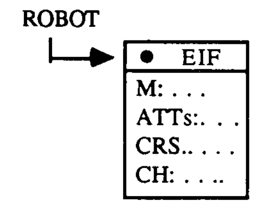
System Internal Conversion

1.) ROBOT

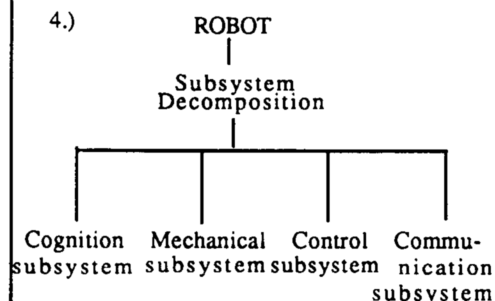
2.)



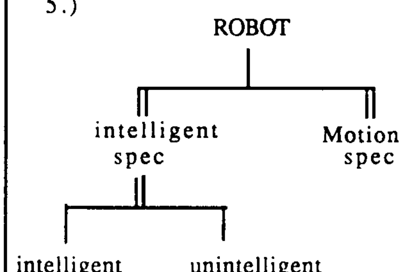
3)



4.)



5.)



n.) Store the model at Global KB

Figure 3 KAR with FRASES for Robot Model Construction the forward-reasoning and backward-reasoning strategies are the basic schemes. The typical examples of them are the OPS system and the MYCIN shell (Waterman, 1986). At the present time, we focus on a model generation by exploring the best reasoning strategy using the FRASES representation.

3.1 Automatic Generation of Selection Rules

A set of selection rules is defined for representing selection constraints imposed on an entity set to select appropriate model objects from taxonomic relations of FRASES (Rozenblit and Huang, 1987). With the frame characteristics, the members of an entity set have attributes and own values. The members of the entity set can be then distinguished by these values. One of the characteristics of frame representation is that the selection of instances is based on the objects' slot values. Possessing the features of frame representation, FRASES implicitly contains the selection rules. The inferencing mechanism should regard these features as a set of selection rules. The implicit selection rules will reduce the complexity of the knowledge base as well as facilitate future refinements of the rules by simply changing the values of objects' slots.

3.2 Rule Indexing

It may be infeasible to find the rule that matches a given situation by systematically checking each rule in a knowledge-based system with a large number of production rules. We add an additional index slot for each EIF frame. All the rules associated with an EIF frame have the same index. The purpose is to find a much more limited set of applicable rules and thus improve the search efficiency. This method reduces the number of rules that need to be considered as compared with an exhaustive consideration of each rule. This approach is especially beneficial in a hierarchical knowledge base since the *searching area* usually occurs in the neighborhood of the current state. There are actually two numbers in the rule index. One indicates the degree of breadth. The other indicates the degree of depth. They are used for forward-reasoning and backward- reasoning strategies respectively.

3.3 Opportunistic Inferencing

Top-down and bottom-up strategies have often been employed in problem solving. In MODSYN, we have incorporated a backward-reasoning strategy with a bottom-up instantiation (Rozenblit and Huang, 1989). The alternative is to employ a forward-reasoning strategy with top-down instantiation process for model generation. For example, a blackboard model has been proposed and successfully applied in Hearsay-II system by using both reasoning strategies alternatively (Erman et al., 1980). The central issue of dealing with real-word problem is "What pieces of knowledge should be applied, when and how?" (Nii, 1986). In other words, pieces of knowledge should "know" when they should apply and how they could apply (forward or backward reasoning). This is called opportunistic reasoning. The blackboard model is a typical example of opportunistic problem solving by using several highly structured queues. Our efforts are focusing on exploring an opportunistic reasoning for design model generation by integrating both reasoning strategies we have explored so far.

One possible method to implement this kind of reasoning is to use two extra working memories to store traversing status and fired-rule status. For example, when backward chaining is being used, the inference engine may check the fired-rule status. If there are only a few rules that have not been fired, then forward reasoning may be applied. We can consider both reasoning chains as linked lists. As shown in Figure 4, the blocks stand for candidate rules. In part a) of the figure, forward chaining begins with the premises of some block. Then a subset of next candidate rules is fired. The process will stop after exploring all candidate rules and will form a singly-linked list. Backward reasoning is the same as forward reasoning except that it begins with the conclusion parts of some candidate rules. The purpose of opportunistic reasoning is to create a doubly-linked list. The "reasoning direction" is not restricted to forward or backward. In order to achieve the reasoning, the traversing status (or pointer status) and fired-rule status are needed as shown in part c) of Figure 4.

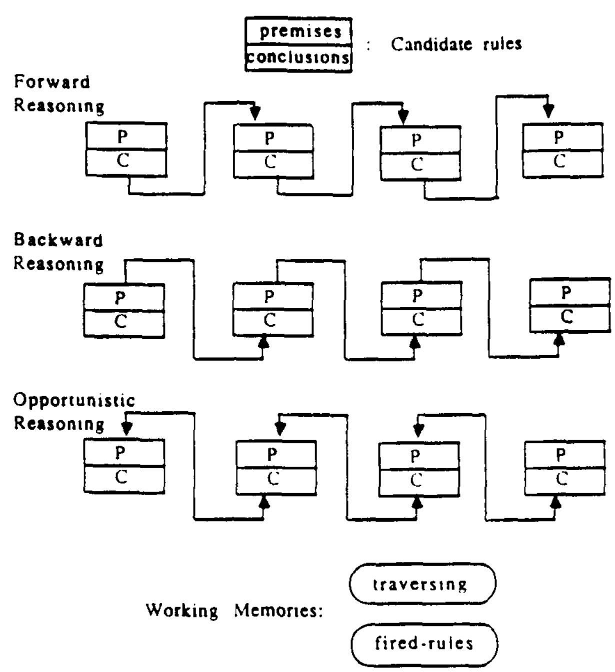


Figure 4 Comparisons of Reasoning Chains

4. EXECUTABLE SIMULATION MODEL CONSTRUCTION

The model construction process can be divided into three subtasks:

- Construction of design models.
- Construction of experimental frames.
- Synthesis of design models and experimental frames.

The construction of experimental frames based on the *Atomic Frame* concept has been developed in our previous work. For implementation details, readers are referred to [Hu 1989; Rozenblit and Hu 1989].

The atomic frame concept can be extended to serve as the basic scheme for construction of simulation models. Once the design specification characterizing the input/output requirements, performance requirements, and technical constraints is given, the rule-based model constructor analyzes and extracts all the required atomic frames (e.g., a FIFO/LIFO/Priority queue, a Up/Down-Counter, etc.) for construction of a *Generic Model* that will logically fit the needs of model specifications. Here *generic* stands for simulation language-independent. The generic models are then translated into simulation models based on the target language employed. This enables the system to adapt to different simulation environments by adding grammars of simulation languages.

After models and experimental frames are generated, the system starts the synthesis of a simulator module. This is accomplished by acquiring coupling information of a composition tree. Formally, a pruned FRASES structure can be converted into a number of composition trees, each of which represents a model alternative. The coupling information on decomposition nodes together with the desired performance indices indicated on each entity node will direct the system to conduct a bottom-up model synthesis process.

The final synthesized model is then validated and simulated for performance analysis. Model refinements may be required if none of the synthesized models fit the performance requirements. This requires the application of *learning from experience* for refinement of the knowledge base. The overall model construction process is depicted in Figure 5.

5. RULE-BASED AUTOMATIC ANALYSIS

The simulation results must be collected and evaluated to see if the generated model is satisfactory. Two methods have been developed for the purpose: causal path analysis and rule-based diagnosis (Reddy et al., 1986). The former is derived from statistic concepts. The regression and correlation between variables are investigated. The latter uses simple rules to determine whether some variables' values should be changed or not.

The rule base proposed here contains the expertise of simulation domain analysts. Two kinds of actions will be taken if necessary after the analysis: a local refinement and model regeneration. If there are only few results which are not satisfactory, the rule base will be responsible to inference the possible update on input constraints. In other words, some attached variables' values (i.e. constraints) may need to be changed to maintain the consistency of behavior constraints. The inference engine takes the results of analysis as new facts to inference using Global KB. If there are only few specifications that need to be specified again, the VKB will suggest the user to replace some submodels. In the extreme case then the simulation results of the generated model are very far away from the goal, another consultation phase will be suggested.

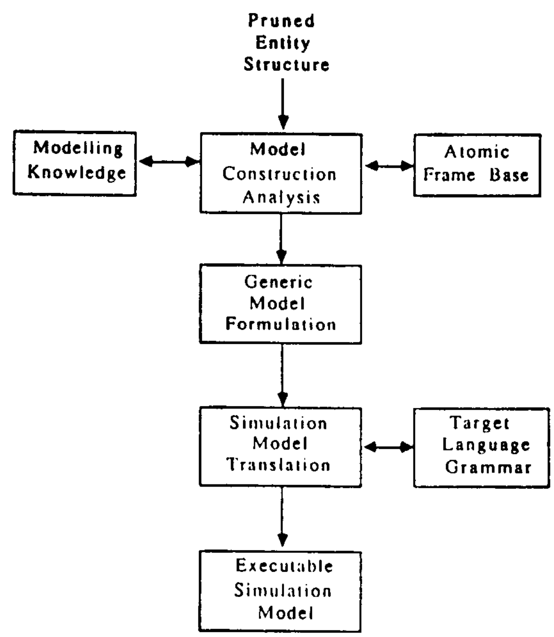


Figure 5 Model Construction Process

6. CONCLUSIONS

To unify the operations among the four KBs, a unique inferencing mechanism should be designed to drive them. A working memory will be shared by all KBs to maintain the consistency. When running the system, the user will be directed by Meta-Acquisition KB to create a family of models using his or expert's knowledge. The model represented in FRASES will be stored in the Global KB. Then, the system will enter the consultation phase. The system will question the user for constraints for a desired model. A set of specifications will be given by the user to generate the final model. The system will use the modeling knowledge from the Construction KB to generate an executable simulation model. Finally, the simulation results will be collected and evaluated by Validation KB. The operation process of knowledge management in model development is shown in Figure 6.

We have presented the concept design of an automatic model development using knowledge bases approach. The expected advantages over other approaches are the speed up the model construction. This speed up will be afforded by using the FRASES acquisition process and a global model database for facilitating the refinements.

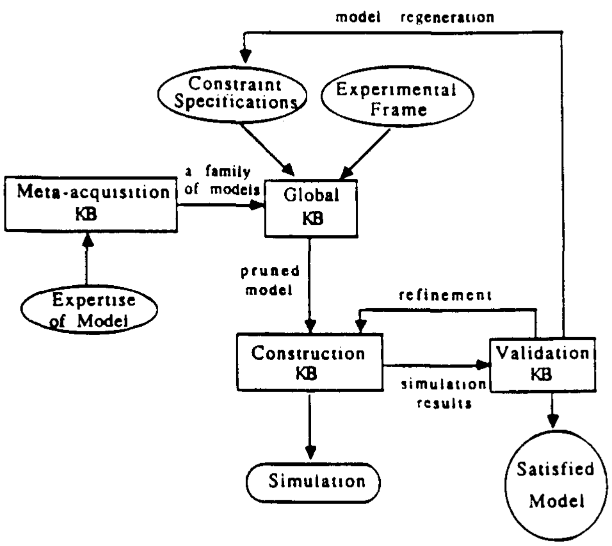


Figure 6 Operation Process of Knowledge Management in Model Development

REFERENCES

Balci, O. and Nance, R. E. (1987). Simulation model development enviroments: a research prototype. *J. Opl Res. Soc.* Vol. 38, No. 8, 753-763.

Balmer, D. W. and Paul, R. J. (1986). CASM-the right environment for simulation. *J. Opl Res. Soc.* 37, 443-452.

Doukidis, G. and Paul, R. J. (1985). Research into expert systems to aid simulation model formulation. *J. Opl Res. Soc.* Vol. 36, No. 4, 319-325.

Erman, D. L., Hayes-Roth, F., Lesser, V. R., and Reddy, D. Raj. (1980). The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Survey* 12:213-253.

Ericsson, K. A. and Simon, H. A. (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, 453-472.

Hu, J., Huang, Y. M., and Rozenblit, J. W. (1989). FRASES - A knowledge representation scheme for engineering design. *Advances in AI and Simulation*, SCS Simulation Series, vol.20, no.4, 141-146.

Hu, Jhyfang and Rozenblit, J. W. (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, 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.

Murray K. J. and Sheppard, S. V. (1988). Knowledge-based simulation model specification, *Simulation*, March, 1988, 112-119.

Nii, H. P. (1986). Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architecture. *The AI Magazine*, Summer, 1986, 38-53.

Olson, J. R. and Rueter, H. H. (1987). Extracting expertise from experts: methods for knowledge acquisition. *Expert Systems*, August 1987, Vol.4, No.3, 152-168.

Pan, N. (1989). A lisp-based shell for model structuring in system design. *Master Thesis*, Dept. of ECE, Univ. of Arizona, Tucson, Arizona.

Reddy Y. V. et al., (1986). The knowledge-based simulation system. *IEEE Software*, March, 1988, 26-37.

Ritchie, I. C. (1984). Knowledge acquisition by computer induction. *Proceedings of UNICOM Seminar*, London, England.

Rozenblit, J. W. and Hu, J. (1989). Experimental frame generation in a knowledge-based system design and simulation environment. *To appear in Modeling and Simulation Methodology: Knowledge System Paradigms*, (M. Elzas et. al., eds), North Holland, Amsterdam.

Rozenblit, J. W. and Huang, Y. M. (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 multifacetted modeling and system design, *ORSA Journal on Computing* (in review).

Waterman, D. A. and Newell, N. (1971). Protocol analysis as a task for artificial intelligence, *Artificial Intelligence*, 2, p.285.

Waterman, D. A. (1986). *A Guide to Expert Systems*, Addison-Wesley, Reading, Mass..

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

AUTHORS' BIOGRAPHIES

Yueh-Min Huang is a Ph.D. student in the Department of Electrical and Computer Engineering at University of Arizona. He received a B.S. in Engineering Science from National Cheng Kung University (Taiwan) in 1982, and M.S. in Electrical Engineering from University of Arizona in 1987. His current research interests include artificial intelligence, knowledge-based system design, and system modeling. He is a student member of Eta Kappa Nu, AAAI and IEEE Computer Society.

Yueh-Min Huang
Dept. of Electr. and Computer Engr.
The University of Arizona
Tucson, Arizona 85721
(602)621-3517

Jerzy W. Rozenblit is an assistant professor in the Department of Electrical and Computer Engineering at University of Arizona. He received his Ph.D. in Computer Science from Wayne State University in Detroit, in 1985. His research interests are in the areas of modeling and simulation, system design, and artificial intelligence. He is a member of ACM, IEEE Computer Society, and The society for Computer Simulation.

Jerzy W. Rozenblit
Dept. of Electr. and Computer Engr.
The University of Arizona
Tucson, Arizona 85721
(602)621-6177

Jhyfang Hu received his M.S. and Ph.D. degrees in Electrical Engineering from University of Arizona in 1986 and 1989. His research interests include artificial intelligence, modeling and simulation, and design automation. He is a member of Eta Kappa Nu and IEEE Computer Society.

Jhyfang Hu
Dept. of Electr. and Computer Engr.
The University of Arizona
Tucson, Arizona 85721
(602)621-3517