Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

AN ARCHITECTURE OF A KNOWLEDGE-BASED SIMULATION ENGINE

Madhav Erraguntla
Texas A&M University
College Station, TX

Perakath C. Benjamin
Knowledge Based Systems, Inc
College Station, TX

Richard J. Mayer
Texas A&M University
College Station, TX

ABSTRACT

This paper describes the architecture of a *Knowledge-Based Simulation Engine* (KBSE). It summarizes ongoing efforts at Knowledge Based Systems Laboratory, Texas A&M University and Knowledge Based Systems Inc., College Station, TX, in developing knowledge based simulation. The KBSE provides support for: 1) developing a simulation model from a description of the system and the user concern in the form of a question to be answered, and 2) analyzing the simulation output. The support for output analysis includes assistance for statistical analysis, explaining a set of observations, and generating alternative ways to improve performance/solve a problem. The approach used to develop KBSE was to integrate simulation expert knowledge with domain knowledge, commonsense reasoning techniques, and qualitative and quantitative simulation techniques. The Phase I KBSE automates the generation of simulation models from system descriptions. Phase II provides intelligent support for simulation output analysis. This paper presents a summary of the Artificial Intelligence (AI)-based algorithms and heuristics developed in Phase I.

1 INTRODUCTION

The increasing complexity of organized enterprises has enhanced the attractiveness of simulation modeling as a decision support tool. However, this powerful tool still remains beyond the reach of many application domain personnel. The use of simulation among practitioners has been hindered for several reasons, two of the most salient being: 1) the long lead time and the considerable effort required to build a simulation model, and 2) the extensive training and skill required for the effective design and use of simulation modeling techniques.

Recent advances in the area of simulation modeling have focused on improving simulation modeling languages. These advances have attempted to reduce the semantic gap between a simulation model design and the corresponding execution simulation program. They represent important advances for improving the productivity of simulation modelers, but do little to aid the non-simulation trained decision maker. The research presented here addresses this problem directly by providing automated support for the simulation design activities from a description of a system and an user concern in the form of a question to be answered. The motivation for our research is the observation that domain experts often find it convenient to communicate their knowledge by relating an ordered sequence of activities that forms a description of "how things work" in their domain. These descriptions, often unstructured, contain much of the knowledge needed to reason about the problem at hand (Benjamin et al., 1993).

Output analysis is an important, but often neglected part of simulation. Research in the use of AI for analyzing simulation output has gained momentum in the past decade. Most of the research has focused on the statistical aspects of output analysis (Deslandres, 1991, Sargent, 1986). Researchers in the use of AI for statistical analysis have developed stand-alone software systems that assist statistical analysis (Pregibon, 1984). Most of the research focuses on the tactical analysis activities and fail to address the strategic aspects of simulation output analysis. Recent work by (Prakash, 1989) attempts to develop intelligent support for simulation model redesign using a knowledge based approach. The *Qualitative Reasoning* (QR) approach to simulation result interpretation was proposed by (Mayer, 1988). This research further investigates the utility of the QR approach and rapid transfer of the research to an implementation architecture. In this effort, we address the issue of providing knowledge based assistance to the user in this process of gathering information about the system, where simulation is the mechanism through which information is obtained. Our support for output analysis thus includes not only the statistical analysis but also knowledge-based reasoning to support the information gathering process.

2 APPROACH

The goals of KBSE are: 1) to support the design of a simulation model from a description of the system and

the user concern in the form of a question to be answered, and 2) to support the interpretation of simulation output. This section describes our approach to accomplish each of these goals. Figure 1 shows the simulation activities supported by the KBSE.

The Simulation model Design Process

1) *Capture system description*: Domain experts often find it convenient to communicate their knowledge by relating an ordered sequence of activities that form a description of how things work in their domain. These descriptions, often unstructured, contain much of the knowledge needed to reason about the problem at hand. The first step in the KBSE is to capture these system descriptions.

2) *Capture user concern*: The application domain personnel (henceforth referred to as the 'user') often wants to use simulation to solve some perceived or imagined problem. The second step in the KBSE is to capture these user concerns.

3) *Establish model boundaries*: This activity involves selecting the processes and objects of interest in the model. Elements of the system providing information that assist in accomplishing the modeling goals will be included inside the model boundary; those elements that do not contribute in an informational way towards attaining the goals are excluded.

4) *Translate user concerns to simulation parameters*: In general, the user concerns are broad in their scope and can't be answered directly. These broad concerns need to be broken down into detailed issues that could be addressed by simulation. The KBSE approach involves breaking the user concerns into sub-questions, performance metrics, and units of measurements. This step is a domain knowledge acquisition process.

5) *Determine abstraction level*: This activity involves determining the degree of detail (or abstraction) the model should have. The model should be detailed enough to draw valid conclusions, and abstract enough so as to reduce model development time and simulation run time.

6) *Generate executable model*: The final step in automated model generation is to generate executable simulation code. In the present version of KBSE, the models are generated in WITNESS™.

Activities 3, 4, and 5 together constitute the *conceptual model design process*. Thus, conceptual model design involves establishing the model boundaries, translating user concerns to simulation parameters, and determining the level of abstraction.

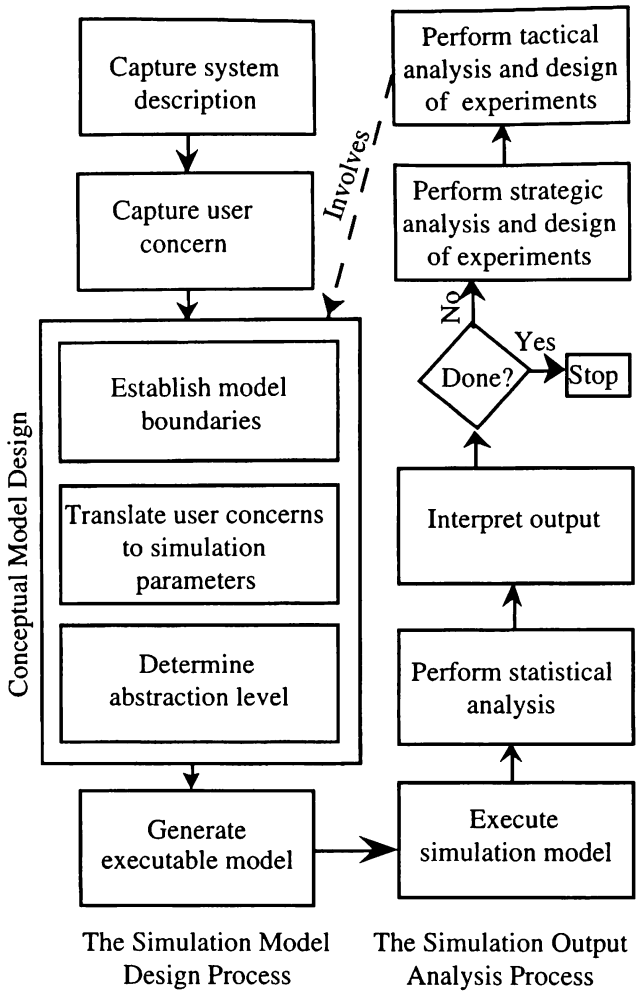

Figure 1 : Simulation Process

Output Interpretation and Analysis

1) *Perform Statistical analysis*: This activity involves statistical analysis of the output of the simulation runs such as confidence interval determination, etc.

2) *Interpret output*: This step involves knowledge based reasoning about the simulation output. Various hypothesis are proposed to explain why the system behaves in the way it does. For example, the user might be interested in determining system capacity. If, after the initial simulation runs, it is believed that the capacity is low, various hypothesis such as: a) the capacity is low because of low utilization, b) the capacity is low because of high breakdown rate, etc. will be proposed. These hypothesis will be tested in the subsequent stages for validity.

3) *Perform strategic analysis and design of experiments*: In this stage, the validity of the various hypothesis proposed in the previous step are determined. Wherever possible this will be done through Qualitative Reasoning (QR) (DeKleer, 1984, KBSI, 1991, KBSI, 1992,

Kuipers, 1987). In the cases where it is not possible to do so, additional simulation models are designed to test the validity of these hypothesis.

4) *Perform tactical analysis and design of experiments*: This step involves generation of additional simulation models, in case it was decided to use simulation to test the validity of the various alternate hypothesis proposed to explain the behavior of the system. This step involves interaction with the simulation model generator as shown in Figure 1.

As stated earlier, automated support for simulation model design was implemented in the Phase I KBSE. Support for output analysis will be implemented in Phase II. The various algorithms/heuristics developed and implemented in Phase I are presented in Section 3. An outline of our approach to some of the activities in the Phase II of KBSE are presented in Section 4. The issues in Section 4 are still in a research stage and are yet to be implemented. We present the broad approach we have in mind.

3 KBSE MODULES

3.1 System Description Capture

For the unstructured descriptions to be useful, we need a systematic methodology that: 1) can be used to capture the descriptions, 2) is formal enough to support some validation of the descriptions, and 3) captures these descriptions in a manner that facilitates automated reasoning. The Air Force IDEF3 Process Description Capture Method was used to facilitate the capture of structured system descriptions (Mayer, 1991) (A brief tutorial on IDEF3 is included in the Appendix).

3.2 User Concern Capture

The users very often wants to use simulation to solve some perceived problem. While they know what they want, in general, they are not sure if their concerns can be addressed by simulation. Hence, the first step in knowledge-assisted simulation model development is to assist the users in deciding whether their concerns can be solved through simulation or not. We propose to do this by capturing the domain knowledge about the set of concerns that can be answered through simulation. Based on interviews with domain experts, we identified the concerns given in Table 1 as some of the typical concerns that are relevant to the manufacturing personnel and that could be answered through simulation. The user concern capture is a domain knowledge acquisition activity. Based on the feedback of the customers and users, this set will be continuously refined. Not only is such a set informative to the users in deciding whether their concerns can be addressed by simulation, it is also the driving force for the entire knowledge-based simulation engine. The captured user concerns help establish the objectives of the simulation exercise. The automated simulation engine we are developing will present this list to the user. The user can just browse through this list and select the relevant questions. On the basis of the selections and the IDEF3-based description of the system, a simulation model will be automatically designed. The user can then run the model and generate data that will (hopefully) enable answering the questions.

Table 1: Sample User Concerns for Production Systems

1. Will I be able to meet my production requirements?
2. Will I be able to meet my due-date commitments?
3. Are my operational costs going to be too high?
4. What are the strategic decisions I need to make to maximize my productions and/or profits?
5. What should be my strategic production planning decisions?

3.3 Boundary Setting

3.3.1 Problem Description

A preliminary activity in developing a simulation model is determining the model boundaries. This activity involves selecting the processes and objects of interest in the model. This implies filtering out of irrelevant information. The ability to identify dependency links between items in the model and the modeling goals is vital to this process. Elements of the system providing information that assist in accomplishing the modeling goals will be included inside the model boundary; those elements that do not contribute in an informational way towards attaining the goals are excluded. For example, when modeling a shop floor control system with the goal of determining machine utilization, the machining and set-up activities might be relevant because they affect the utilization level. The material procurement process will be outside the boundaries of the model because it has no impact on utilization. We present in this Section a boundary fixing heuristic for the manufacturing domain, based on IDEF3 descriptions.

3.3.2 Boundary Setting Algorithm

The boundary setting algorithm we present here is based on the concept of an *upstream* process (Lin, 1990). A process *A* is said to be the upstream process of a process *B* if the output from *A* is an input (either directly or after some intermediate processes) to the process *B*. It is possible to determine the upstream processes of a given process in IDEF3 by tracking down the links and referents. The boundary setting algorithm we propose is based on this concept of upstream processes.

1) If the performance metric of interest is of global scope (like capacity of the entire plant), include the entire description.
2) If the performance metric of interest is of local scope (like capacity of a specific machine), consider the processes at the machine and all the upstream processes. Let's call this set the *set of relevant processes.*
3) For each resource in each process included in the relevant process set, if any other process uses the same resource, include that process also in the relevant process set.
4) For each process included in Step 3, add all the upstream processes.
5) For each process included in Steps 3 & 4, repeat 3 & 4 until no further process is included in the relevant process set.

The relevant process set is the set of processes of interest for the concern that needs to be answered. All the extraneous process are automatically deleted from the description or highlighted to the user for removal from the description before simulation model development. This makes the simulation model concise and will result in savings in simulation run time.

3.4 Translate User Concerns into Simulation Parameters

This step involves translating the broad concerns of the user into detailed parameters that can be measured through simulation. The user is often interested in broad issues such as the ones identified in Section 3.2. These broad issues can't be answered directly (by simulation or any other analytical means in general) and need to be broken down into detailed sub-questions. For each of these sub-questions, the performance measures and units of measurements then need to be determined. The process of determining performance measures starting from a set of user concerns is a domain knowledge-intensive reasoning process. In KBSE, all this domain knowledge is explicitly captured through discussions with domain experts and KBSE users. Just like the domain knowledge in Section 3.2, this domain knowledge will also be constantly updated based on the user and domain expert feedback. Users are guided through these mappings until they are able to translate their concerns into detailed issues that can be addressed by simulation. As soon as the user selects a concern from Table 1, all the relevant sub-questions will be displayed on a dialog box. Once the user selects all the relevant sub-questions, the appropriate performance measures and units of measurement will be displaced for selection. In this manner, the user's broad concerns are translated into detailed performance metrics. Thus, translating user concerns into simulation parameters consists of hierarchical decomposition and concept mappings (Figure 2). Some of the mappings that we developed for the manufacturing domain are shown in Table 2. Only a part of the domain knowledge we captured is presented in this paper. Additional details can be obtained from the authors. User concerns can be grouped into three categories:

1) *Concerns that can be answered directly by model execution.* These are the questions that can be answered in one execution of the model (with a suitable number of runs to get the desired confidence interval). These type of questions most often involve determining some system performance metrics such as capacity, utilization, etc.

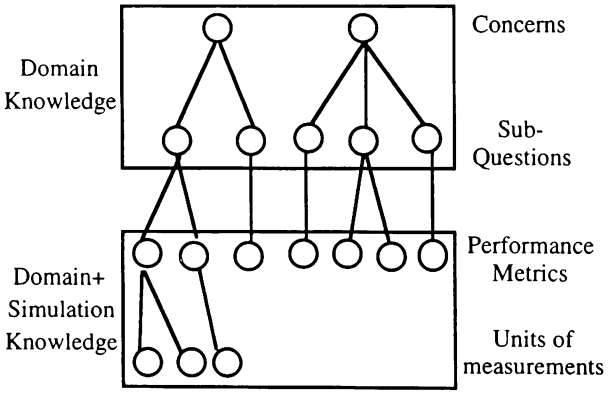

Figure 2. Hierarchical Decomposition and Concept Mapping Technique

2) *Concerns that need comparison of results of different simulation models.* Sometimes users are interested in selecting between two or more alternatives. Such problems involve comparing the results obtained from different simulation executions.
3) *Concerns that involve optimization.* Sometimes users are interested in finding the optimum of some system parameter setting. For example, they might be interested in questions such as, "How many resources do I need?" and "What is my best make-buy ratio?" Answer to such questions need an

optimum seeking heuristic in addition to performance evaluation at each setting.

We found that user Concerns 1, 2, 3 in Table 1 are of the first type, where as Concerns 4 and 5 are of the second and third types in the above classification. Concerns of the first type will be of primary interest in Phase I of KBSE. We have developed optimization seeking heuristics that integrate simulation and Response Surface Methodology (RSM) techniques (Benjamin, 1991) for concerns 3 and 4. These will be implemented in Phase II of KBSE.

Table 2. Concern->Sub-Question->Performance Metrics->Unit of Measure Mappings

Concern	Sub-questions	Performance Metrics	Units
Will I able to meet my	What is my capacity?	a. Capacity of the entire unit b. Capacity of a particular m/c	output/time process time
production requirements?	What is my utilization?	a. Avg. machine utilization b. Avg. labor utilization c. Specific machine utilization d. Specific labor utilization	% utilization % idle % set up % breakdown

3.5 Abstraction Level Determination

3.5.1 Problem Description

Once we allow the user to model at multiple levels of abstraction, the natural questions that the user faces are "what processes need to be decomposed?" and "to what level should I decompose?" If the user has any preferences based on domain knowledge, experience, or intuition, we follow them. This section addresses the issue of providing the user with knowledge based assistance in the case where there aren't any default preferences.

3.5.2 Example

Consider the description of a production process in IDEF3 (Figure 3).

The questions that need to be answered include the following types: 1) what is the *ideal* level of decomposition, 2) should I decompose all the Units of Behavior (UOBs), 3) should I decompose one level or two levels, 4) should I decompose only UOB 4 and leave UOB 3 etc.

3.5.3 Philosophy

The ideal level of decomposition is determined in KBSE on the basis of the following:

1) Level of abstraction is determined based on the objective of the simulation, that is, based on what questions need to be answered.
2) Determination of the appropriate level of decomposition is an iterative process. We first create the simulation model at the highest level of abstraction and run the model. If the results are *satisfactory*, we are done. Otherwise, we use the knowledge gained during the execution of the model to determine which UOBs to decompose.

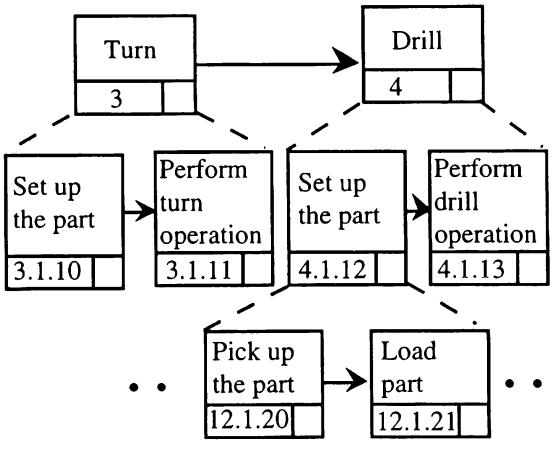

Figure 3. Description of a Production Process in IDEF3

3.5.4 Approach

Our approach to determining the appropriate level of abstraction is summarized as follows:

1) Build and run the model at the highest level of abstraction.
2) Based on the objective of the simulation, determine an *abstraction parameter*. Abstraction parameter is a parameter on the basis of which UOBs are decomposed. Right now, the abstraction parameters are limited to either time or cost. All the others can be reduced to some combination of these basic abstraction parameters.
3) While running the simulation model, apportion the abstraction parameter to each process. For example, if the question that needs to be answered is: "what is my

capacity?" the abstraction parameter could be process time. So we apportion the time the entity spends in the system to each of the processes.

4) If there is a single UOB contributing to more than (say) 30% to the abstraction parameter, then that UOB is the candidate for decomposition. For example, if we are interested in finding the capacity of the plant and we know that the part spends more than 30% of it's time in a particular process, it makes sense to decompose that process.

5) Once we have identified the set of processes that need decomposition, we decompose these processes in the next iteration and repeat the procedure. The stopping criterion is that there shouldn't be a single process contributing significantly (30%) to the abstraction parameter. This defines the *satisfactory* level of mixing of abstractions.

3.6 Generate the Executable Model

Once we have the detailed performance metrics and their units of measurement, the boundary of the model and the level of abstraction, generation of an executable model is a straightforward exercise. It will involve deciding issues such as how exactly to measure the performance metrics of interest, what variables to introduce, where to store the values, etc., as well as the actual translation of the description of the system to executable simulation code. In KBSE, the executable simulation code is generated in WITNESS™.

Table 3. Question, Abstraction Parameter, and Apportion Strategy Mappings

Question	Abstraction Parameter	Apportion To
1.a) Capacity of the entire unit	Time an entity spends in the system.	Processes
1.b) Capacity of a particular resource	Time an entity spends at the specific resource.	Processes occurring at that resource

4 OUTPUT INTERPRETATION AND ANALYSIS

In this Section, we present our overall approach to output interpretation and analysis.

4.1 Explaining Behavior Based on Qualitative Simulation Traces

An activation trace that is output from a qualitative simulation engine is an ordered sequence of instances of activities and events along with a history of changes that occurred to the state of the qualitative model. Such a trace will walk through different paths of an IDEF3 process flow diagram. The generated sequence of events will be governed by both the constraints implicit in the IDEF3 language (Mayer, 1991) and constraints about the real world processes described that are explicitly recorded in the IDEF3 description. Qualitative IDEF3 models are partial, requiring very loose "completion criteria." Their partiality is advantageous for activities such as IDEF3 description validation for assisting quantitative simulation model design [see (KBSI, 1991a)].

Our approach will be to analyze the information contained within the activation trace to generate an explanation for observed behavior. We will develop reasoning mechanisms that will enable the automated construction of such explanations. Initially, we will restrict the scope to generating explanations *in response to specific queries*. For example, suppose that the activation trace indicates that a particular machine in a manufacturing IDEF3 model is always idle. Suppose that the question to be answered is "Why is this machine idle?" Examining the "preconditions" for starting processing on the machine showed a constraint "if the part type is A and Operator #5 is available, then start processing." Further examination revealed that Operator #5 could never be available whenever parts of type A arrived at that machine. It is apparent that deriving this explanation requires reasoning with selected parts of the activation trace. It is also the case that generation of an explanation may require information in the original IDEF3 description that was not included in the model itself. In the process of designing even the qualitative model, only portions of the original system description are considered relevant.

4.2 Generating Alternative Solution/ Performance Improvement Strategies

This activity can be performed on two kinds of output: 1) summary statistics from qualitative simulations or 2) summary statistics from quantitative simulations. The main difference between an activation trace and the statistical simulation output is that the information in statistical output is both more summarized and more focused. Statistical outputs contain data that have been collected over time and compiled to focus attention on key performance aspects of the model. The interpretation of simulation output statistics will therefore require an

approach that is considerably different from that needed to interpret the activation trace.

We now present an example to illustrate our approach to generating hypotheses based on observed simulation results. Consider a production line with three machines in series (Figure 4). Suppose the manager of the line conducts a simulation study with the goal of determining capacity. Suppose that the mean capacity (measured in terms of production rate in #/day) is found to be 100. Suppose that this measured capacity is not acceptable and the manager would like to increase the capacity to 240. Thus, the next objective is to determine if and how to increase the line capacity as desired. The proposed system must provide support that will assist accomplishing the manager's goal. Using heuristic domain knowledge, the system might suggest one of the following strategies:

1) Increase the # of bottleneck resource(s).
2) Reduce the processing times of the bottleneck resource(s).
3) Combine strategies 1 and 2.

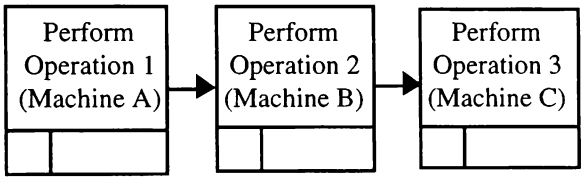

Figure 4. Example to Illustrate Intelligent Hypothesis Generation

We now describe how a QR approach can be used to modify the above strategies. Suppose it is known that there is a degree of imbalance in the line: the mean processing time for Machine A is three minutes, Machine B five minutes, and Machine C four minutes. The imbalance in the line (a structural problem) is the cause for the bottleneck at Machine B and also the cause of lower production and the accumulation of inventories at Machine B. This additional structural knowledge will help decide the actual levels of the resources re-allocation and/or changes in processing time (for example, it may not be wise to reduce the processing time of Machine B below four minutes to maintain balance between Machines B and C). To further illustrate the use of structural information, suppose that the UOB Operation 2 (Machine B) has a decomposition; that is, a description at a finer level of detail. A strategy to reduce the processing time will be to examine the constraints within each of the processes within this decomposition and determine the effect of changing parameters that affect these constraints. In summary, the approach we suggest is to derive such "performance improvement" strategies by the automated analysis of the *system structures* starting from the IDEF3 descriptions.

5 CONCEPTUAL ARCHITECTURE OF KBSE

Due to space constraints the section on the conceptual architecture of KBSE is removed from the final draft. The original, complete draft can be obtained from the authors.

6 SUMMARY

This paper documented the conceptual architecture of a Knowledge-Based Simulation Engine (KBSE). The KBSE provides support for: 1) developing a simulation model from a description of the system and the user concern in the form of a question to be answered, and 2) analyzing the simulation output. The algorithms and heuristics that were encoded in the KBSE were described. The research illustrates the integration of AI, qualitative reasoning techniques, and discrete-event simulation methods. The research provides significant benefits to the simulation research and user communities.

REFERENCES

Benjamin P. C. 1991. *Towards a New Method for the Design of Robust Systems using Computer Simulation Experiments*. Ph.D. Dissertation, Texas A & M University, College Station, TX.

Benjamin P.C., F. Fillion, R. J. Mayer, and T. M. Blinn. 1993. Intelligent Support For Simulation Modeling: A Description-Driven Approach. *Proceedings of the 1993 Summer Simulation Conference*: 273-277.

Deslandres, V. and H. Pierreval. 1991. An Expert System Prototype Assisting the Statistical Validation of Simulation Models. *Simulation*, 56: 79-89.

De Kleer, J. and J. S. Brown. 1984. A Qualitative Physics Based on Confluences. *Artificial Intelligence* 24: 7-83.

Knowledge Based Systems, Inc. 1991. *Knowledge-Based Assistant for Simulation Model Generation from IDEF3 Descriptions*. National Science Foundation Phase I SBIR Final Report, College Station, TX.

Knowledge Based Systems Inc. 1992. *Knowledge-Based Assistant for Simulation Model Generation from IDEF3 Descriptions*. National Science Foundation Phase II SBIR Grant No. III-9123380.

Kuipers, B. 1987. Qualitative Simulation as Causal Explanation. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17: 432-444.

Lin, Min-Jin. 1990. *Automatic Simulation Model Design from a Situation Theory Based Manufacturing System Description*. Ph.D. Dissertation, Texas A & M University, College Station, TX.

Mayer, R. J. 1988. *Cognitive Skills in Modeling and Simulation*. Ph.D. Dissertation, Texas A&M University, College Station, TX.

Mayer, R. J., C. P. Menzel, and P. S. D. Mayer. 1991. IDEF3: A Methodology for Process Description. *Final Technical Report, Integrated Information Systems Evolution Environment Project*. United States Air Force AL/HRGA, Wright-Patterson Air Force Base, OH.

Prakash, S. and R. E. Shannon. 1989. Intelligent Back End of a Goal Directed Simulation Environment for Discrete-Part Manufacturing. *Proceedings of the 1989 Winter Simulation Conference*, 883-891.

Pregibon, D. and W. A. Gale. 1984. REX: An Expert System for Regression Analysis. *COMSAT 1984: Proceedings in Computational Statistics*, 227-236.

Sargent, R. C. 1986. An Exploration of the Possibilities for Expert Aids in Model Validation. *Modeling and Simulation Methodology in the AI ERA*.(M. S. Elzas, T. I. Oren, and B. P. Zeigler, eds.) 279-297.

ACKNOWLEDGMENTS

This research was partially funded by NSF (Grant No. III-9123380).

APPENDIX A: OVERVIEW OF IDEF3

The IDEF3 Process Description Capture Method provides a mechanism for collecting and documenting processes. IDEF3 captures precedence and causality relations between situations and events in a form natural to domain experts by providing a structured method for expressing knowledge about how a system, process, or organization works. IDEF3 captures the behavioral aspects of an existing or proposed system. Captured process knowledge is structured within the context of a *scenario*, making IDEF3 an intuitive knowledge acquisition device for describing a system. IDEF3 captures all temporal information, including precedence and causality relationships associated with enterprise processes. There are two IDEF3 description modes, process flow and object state transition network. A process flow description captures knowledge of "how things work" in an organization, e.g., the description of what happens to a part as it flows through a sequence of manufacturing processes. The object state transition network description summarizes the allowable transitions an object may undergo throughout a particular process. Both the Process Flow Description and Object State Transition Description contain units of information that make up the system description. These model entities, as they are called, form the basic units of an IDEF3 description. The resulting diagrams and text comprise what is termed a "description" as opposed to the focus of what is produced by the other IDEF methods whose product is a "model."

The IDEF3 term for elements represented by boxes is a *Unit Of Behavior* (UOB). Each UOB can have associated with it both "descriptions in terms of other UOBs" and a "description in terms of a set of participating objects and their relations". We refer to the former as *decompositions* of a UOB and the latter as an *elaboration* of a UOB. IDEF3 provides this capacity by allowing multiple decomposition of the same UOB.

BIOGRAPHIES

MADHAV ERRAGUNTLA received his Master's degree in Industrial Engineering from the National Institute for Training in Industrial Engineering in 1989. Currently, he is a Ph.D student in the Industrial Engineering department at Texas A&M University. His research interests are knowledge representation and reasoning, simulation, planning, qualitative reasoning, manufacturing, and operations research.

PERAKATH C. BENJAMIN received his Master's degree in Industrial Engineering from the National Institute for Training in Industrial Engineering in 1983. He received his Ph.D in Industrial Engineering from Texas A&M in May 1991. He is currently Vice President (Innovations and Engineering) at Knowledge Based Systems, Inc. (KBSI). He is responsible for managing and providing technical input to research and development projects.

RICHARD J. MAYER received a Master of Science in Industrial Engineering from Purdue University in 1977. In 1988, he received a Ph.D in Industrial Engineering from Texas A&M University. From 1984 to 1989, he was Project Manager and Principal Investigator on thirty-nine funded research efforts in the Knowledge Based Systems Laboratory. He founded Knowledge Based Systems, Inc. in 1988 and has received funding for applications in engineering design assistance, systems analysis and concurrent engineering methods and tools. Currently, he is an Associate Professor of Industrial Engineering at Texas A&M University, College Station, Texas.