Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

DATABASES AND ARTIFICIAL INTELLIGENCE: ENABLING TECHNOLOGIES FOR SIMULATION MODELING

Martha A. Centeno
Charles R. Standridge

Industrial Engineering Department
Florida A&M University / Florida State University
Joint College of Engineering
P.O. Box 2175
Tallahassee, Florida 32316, U. S. A.

ABSTRACT

Over the last two decades, discrete simulation modeling has matured from *"the tool of last resort"* into one of the most flexible tools available. Simulation modeling is an experimental activity in which model behavior is observed to gain insights about the underlying system. Simulation modeling has evolved over the last 30 years by establishing sound theoretical foundations, by developing more powerful and *easy-to-use* tools, and by seeking the integration of these simulation tools with existing information systems. In this paper, we survey the ways in which Artificial Intelligence (AI) tools and databases have been used to develop enable both modeling support tools as well as new simulation modeling methodologies.

1. INTRODUCTION

Simulation modeling refers to the imitation, using a computer, of the behavior of a *"real"* world system, with the intent of observing the system under various experimental conditions. Many of today's systems are so complex that gaining an understanding of the multiple interactions among components cannot be supported by analytical tools. Thus, an effective way to analyze such a system is to devise an abstract model of it, simplify the model in such a way that superfluous system details are removed without loosing validity, and observe a simulation of the simplified model under the desired sets of experimental conditions. Since its beginning, the simulation modeling process has followed this approach (see Figure 1.)

A study of a system starts when the existence of a problem with the real system is noted, when it is not possible to experiment with the real world system, or when the system is under design. Management needs and expectations must be carefully assessed by the modeler who, in return must determine whether or not simulation is indeed an adequate tool for the analysis of the system under scrutiny. In other words, the modeler must establish a set of assumptions under which an analysis technique, or a combination of techniques, is applicable, feasible and sufficient.

When simulation modeling is used, the modeler will gather data and performs proper statistical analysis to support the study. When there is no data available, the modeler must define the inputs to the model about the system using rules of thumb and/or personal experience.

A conceptual model of a system must be devised and converted into a computer model. The modeler resorts to tools such as model generators, simulation packages, and so on, to make the transition less difficult and less time consuming. The digital model must be thoroughly verified and validated. The reliability of the digital model depends on the quality of the verification and validation processes. Verification entails assuring that the digital code of the model performs as expected and intended, and validation seeks to show that the model behavior represents that of the real-world system being modeled.

With a reliable and accurate digital model, the modeler proceeds to the experimentation phase. Statistical experiments are designed to meet the objectives of the study, and the model is observed and analyzed within the framework of multiple experimental conditions. Upon completion, the modeler prepares a set of recommendations into a management report which includes implementation and operations guidelines.

Providing adequate computer-based support for the SMP has been a non-trivial endeavor. Simulation modeling has relied heavily on the creativity and intuition of its users to carry out some of the tasks in the SMP, such as model abstraction and design of experiments. Furthermore, since the experiment is conducted using a computer, the effective use of it had required "guru" level computer programming skills. These facts had rendered simulation modeling the *tool of last resort*. However, such perception has changed over the years mainly due to two factors: 1) the recognition by the experts in the field that the conceptual stage of the SMP can benefit from quantitative and systematic procedures to stimulate the generation of alternate models, and 2) the incorporation of enabling technologies from various areas of computer science such as artificial intelligence (AI) and databases.

Up until 1960, all simulations were written in a general purpose programming language such as FORTRAN. Tocher and Oren (1960) changed the trend by recognizing that there were some common functions among the models being simulated. A very important aspect of this change was the emphasis given to the productivity of the modeler and the modeling process itself. Then, through GASP II [Pritsker & Kiviat 1969], first discrete event library, GPSS [Schriber 1974, Gordon 1975], a processs oriented language, and SIMULA [Dahl & Nygaard 1967, Eklundh 1979], an object oriented language, support for the idea of common functions was provided, as well as the means of simulating at a higher level of abstraction. During the 1980's, old and new tools began incorporating more capabilities, including databases and graphics. This resulted in powerful simulation tools, such as SLAM II, TESS [Standridge 1985], SIMAN/CINEMA [Pegden 1984, 1991], INSIGHT [Systech 1985], and domain specific packages, such as MAP/1 [Miner & Rolston 1984], and STAR*CELL [Steudel & Park 1987], SIMFACTORY [Russell 1983], WITNESS [Murgiano 1990], AutoMod II [AutoSimulations 1989], and COMNET II.5 [Mills 1988].

Paralleling these developments, new concepts and formalism were being developed, and the importance of simulation environments was further stressed, so as to support not only the coding and execution of the SMP, but also data management, model abstraction and definition. TESS [Standridge (1985)], for example, integrates model building, simulation, analysis, and presentation capabilities on top of a database.

These developments led to a rethinking of the role of simulation languages and packages. Many experts had considered a general purpose simulation language

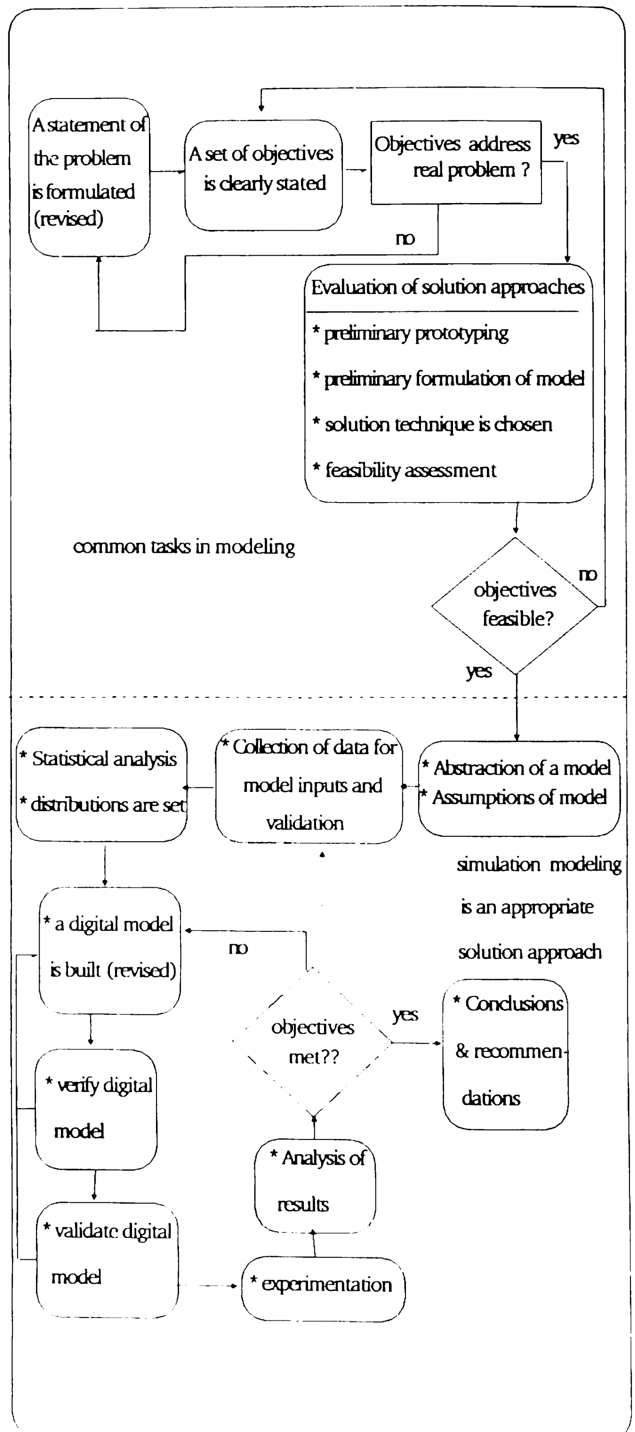

Figure 1: The Simulation Modeling Process

fundamental for simulation modeling. However, this raised questions about how fundamental the simulation language truly is? Could simulation modeling be done without the analysts being concerned with the programming language? Is it possible to reuse simulation models? Is it possible to connect existing information systems to the simulation modeling system? Is it possible to simplify the model construction process? Seeking answers to these questions has brought us to pursue the marriage of AI, data bases, and simulation to design and implement simulation environments.

In this paper, we survey the impact of AI and databases on ***discrete*** simulation modeling. This survey has been organized in two areas: 1) AI and Simulation, 2) Databases and Simulation. AI, databases and simulation as a triplet is discussed through some examples.

2. A.I. AND SIMULATION MODELING

Artificial Intelligence (AI) has been defined by many in different and controversial ways. One useful definition, however, is that given by Rich & Knight [1991]: "AI is the study of how to make computers do things which, at the moment, people do better." Therefore, AI attempts to produce machines that are able to *see* (vision), *speak* (speech), *talk* in a human language (natural language processing), and *reason* based on formal knowledge and past experiences (expert systems, neural nets). Problem solving is one of the main subjects of AI research. AI strategies for problem solving seek to provide a general representation of the problem and allow the computer to search for a solution within the boundaries set by the nature of problem.

Although AI and simulation are two disciplines which matured independently, they have developed a common domain: that of problem solving. The problem solving paradigm in simulation modeling is mainly a search; thus, it parallels the problem solving paradigm in AI (Table 1, Rolston 1988). The goal of the search is to find the "combination of parameter values that will optimize the response values and the controllable variables of the system."[Shannon 1984] AI, in particular knowledge based systems, may be involved with computer simulation not only in the model building process, but also in selecting from among solution methods, in organizing experiments, and in the analysis of the experimental results obtained.

The simulation community has studied carefully the potential of AI techniques in simulation [O'keefe 1986, Reddy 1987, Shannon 1984, Shannon, Mayer, Adelsberger 1985, Zeigler 1984 & 1987, Klahr 1984]]. It has become clear that these two fields can be combined in various ways, using natural language and/or knowledge based systems (KBS), as shown in Figures 2a through 2c.

One way of interaction is to embed a KBS within a simulation model (Figure 2a.) The model would interrogate the KBS as to whether the current solution satisfies the objectives set forth or not. The simulation model can be developed using the classical simulation methodology, with an interaction window to query the KBS at the end of the execution of each run.

Another way is to embed a simulation model within a KBS (Figure 2b). In searching for the solution of a given problem, the KBS would halt the search momentarily, it would query the given simulation model for a result value, and continue its search towards the goal. Again, the simulation model may be built using classical simulation methods with an interaction window as before; however, this time the simulation model has a passive role. It only executes when the KBS decides that it is necessary to do so. This approach to merge both fields allows for more than one solution method to be used if so required.

A novel approach has been to develop new simulation systems using AI-based techniques and tools (Figure 2c). This approach has been sought because the classical approach to simulation modeling can only answer questions of the *what-if* type. It can not answer questions of the type *why?*, *how?*, and *ever/never?*. In other words, under the classical approach to simulation, it is not possible to query why an entity behaved the way it did or how a particular solution was reached. [Erickson 1985,Rothemberg 1989]

It has also become apparent that in order to achieve reusability of models, or of model components, support tools must be designed and implemented using object orientation. This concept has been with the simulation community since the 1960's (SIMULA); in fact, SIMULA had a major impact in establishing the formal concepts of OO [Khoshafian & Abnous (1990)]. Although the OO paradigm has assumed a number of different names in the literature (e.g. *units* [Bobrow & Winograd 1984], *frames* [Fikes & Kehler 1985, Minsky 1975], *actors* [Klahr, Faught & Martin 1980], and *concepts* [Kung 1990]), the basic notion of an object is to organize and store pieces of information relating to a single concept into a single location; thus, it provides a way to describe systems in terms of its components (objects) rather than in terms of procedures. The net result is an improved comprehensibility of simulation experiments by model users and decision makers.

Most of the implementations of object orientation concepts involve great overhead processing, and they view the world through the object structure only, but there are other kind of knowledge that *does not fit* an

Table 1: Problem solving in AI and Simulation

Artificial Intelligence	Simulation Modeling
* Define a problem environment as a collection of states.	* Define a system in terms of objects and the objects characteristics.
* Define *start states* within the space to represent initial problem conditions.	* Establish a set of input parameters as the initial conditions of experiment.
* Define *goal states* that lead to acceptable solutions.	* Define the state space for each output variable so as to satisfy objectives.
* Define a set of operators to guide the changes from one state to another	* Build a digital model of the system.

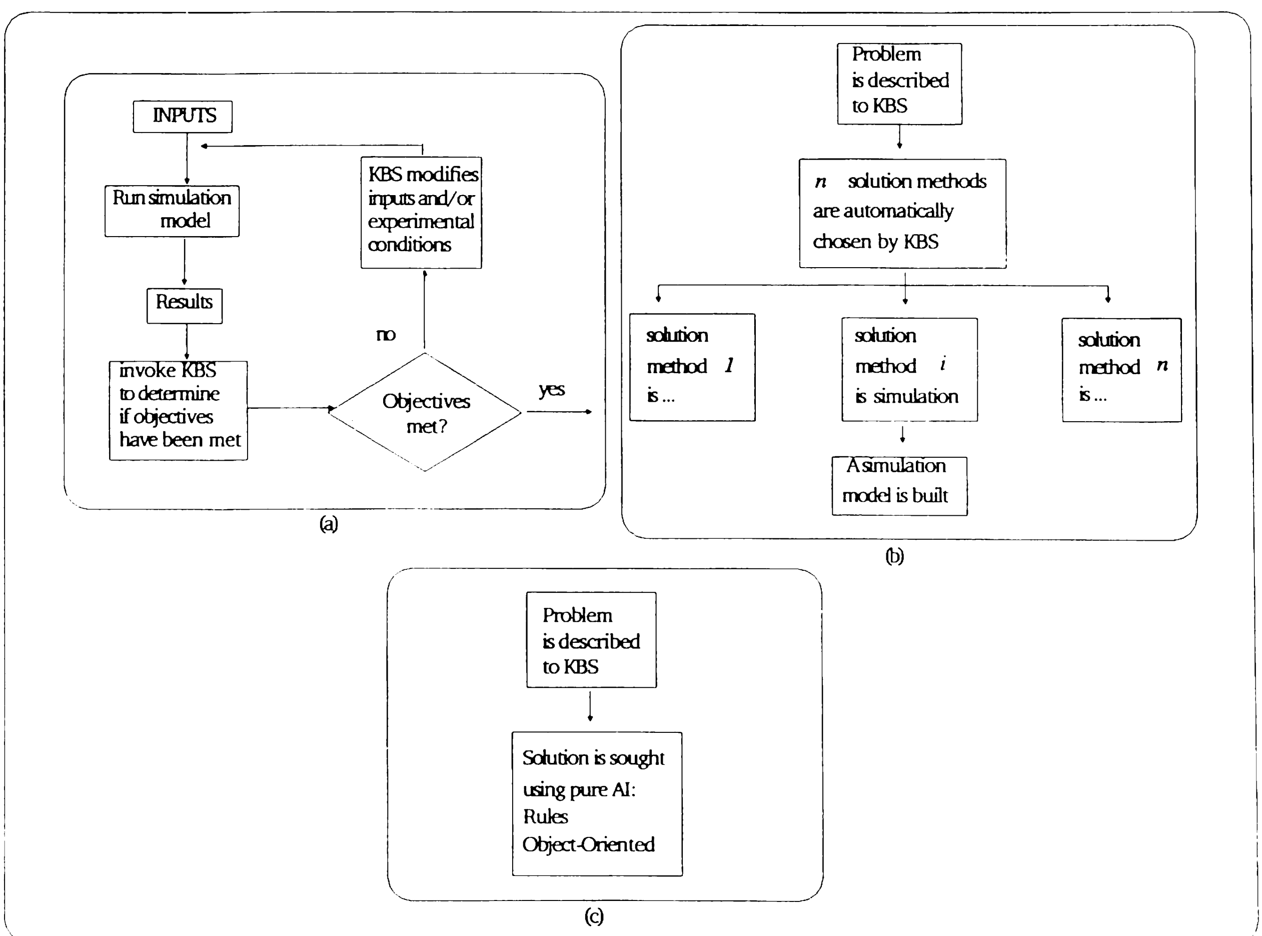

Figure 2: Approaches for merging AI and Simulation modeling

object; thus, other approaches have been explored. One of the most promising methodologies for intelligent simulation is the production rules approach. Rule-based systems use an "IF A, THEN B" framework for knowledge representation. Rules constitute a formal representation of policies, strategies or recommendations. The condition portion of a rule is often a series of predicates that test properties about the current state of the system. The action portion of a production rule then changes the current state of the system. See Genesereth & Ginsberg (1985) and Hayes-Roth (1985) for further treatment of this concept.

Rules are used for multiple purposes in simulation. There are rules to define the behavior or methods which are to be used by objects, rules to test the model for completeness and validity, and rules to drive the model towards goal achievement. When used in conjunction with objects, production rules are used to describe the steps that allow the assertion of new facts into the knowledge base.

Validation and verification can be supported by a series of production rules about completeness of the specified model and flow of entities (objects) through the system. For example, in a complex manufacturing facility with many stations and products made of many parts, it is fairly easy to leave out the definition or declaration of a sub-assembly component (object). The missing component can be detected by a rule such as

IF product P1 is to be assembled at station A1 AND
    object S1 is-part-of product P1 AND
    S1 is not initially located at A1 AND
    S1 is not in routed to A1 AND
    S1 is not the output of any station
THEN print "object S1 has not been defined" AND
    display all objects that are part of P1 AND
    request correction of P1 definition

Production rules are a promising approach to drive a model towards a specified goal. For instance, suppose a service facility (e.g. a medical clinic) is to be studied using simulation. One of the goals of the study is to determine the optimum number of doctors to have on duty, so that patients average waiting time is within an interval (a,b), where $a < b$ and $a,b > 0$. A possible rule to accommodate this goal may be

IF average waiting time is greater than B AND
    number of idle doctors is greater than 0
THEN add 1 to the number of busy doctors AND
    subtract 1 to the number of idle doctors AND
    continue

IF average waiting time is less than A AND
    the number of busy doctors is greater than one
THEN subtract 1 to the number of busy tellers AND
    add 1 to the number of idle doctors AND
    continue

As in the case of OOP, production rule systems have failed thus far to provide a global modeling capability for simulation studies. However, when combined with OOP, rule based systems have greatly enhanced simulation modeling. Zeigler (1987,1989) for example, has taken advantage of the similarity between OOP and discrete simulation formalism to develop an environment for model construction. Discrete models are built in a hierarchical modular manner, successively putting smaller systems together to form larger ones.

ROSS [Klahr, Faught & Martin 1980] is a rule oriented simulation system developed by the RAND corporation. ROSS is an interactive system implemented using LISP. ROSS was developed specifically for war gaming. Real world systems are modeled as objects. Messages are passed between objects, and IF-THEN rules describe the behavior of the objects. The user may halt the simulation at any time, modify the model, and continue the simulation.

KBS [Fox & Reddy 1982, Reddy & Fox 1982] is a knowledge based simulation system developed at Carnegie-Mellon. Like ROSS, it incorporates OOP to describe the real world. Unlike ROSS, it allows goals describing the performance criteria of model components to be attached to objects, and it informs the user whether the goals were met.

Pure AI-based simulation systems tend to execute too slowly to obtain statistically significant results in a reasonable time frame.[Futo & Gergely 1989] Therefore, other research efforts have taken the approach of developing intelligent, automatic programming interfaces for existing and reliable simulation tools. Out of necessity, these systems tend to be limited to a specific domain such as automated guided vehicle (AGV) systems [Brazier 1987], electronic assembly [Ford & Schroer 1987], flexible manufacturing systems [Haddock & Davis 1985, Mellichamp & Wahab 1987], and computer networks [Haigh & Bornhorst 1986, Murray 1986].

Serious knowledge based simulation systems deal with large amounts of data and, thus, need to possess the means to access data efficiently. The development of powerful database management systems provides an opportunity to add the needed data handling capabilities to *AI oriented* simulation.

3. DATABASES & SIMULATION MODELING

The idea behind database concepts is the separation of data manipulation and data organization as much as possible, so that data can be used by both the programmer and the non-programmer. This objective is also present in the development of environments for simulation modeling. A simulation modeling environment seeks to separate systems description, model definition, and model execution as much as possible, so that the model user may concentrate on the modeling aspects of the SMP rather than in the details of writing computer code.

Over the years the database field has evolved to a point in which commercial database management systems (DBMS) can be used as the center piece of a modeling environment. Three main data structures have been developed for databases: hierarchical, network, and relational.

Under the hierarchical model, a database "...consists of an ordered set of trees - more precisely an ordered set consisting of multiple occurrences of a single type of tree" [Tsichritzis & Lochovsky 1976]. A tree, in this context, consists of a root (parent) node from which one or more lower-level (child) nodes derive. From the simulation view point, this data model offers great flexibility and a natural schema for the description of systems to be modeled. However, most hierarchical DBMS require an in-depth knowledge of the physical storage of the records. Thus, merging a hierarchical DBMS to a simulation language has been a cumbersome task.

An evolution from the hierarchical model is the network data structure. As the hierarchical model, it consists of trees, each tree beginning with a parent node and having one or more children. Unlike the hierarchical model, it allows a child node to have more than one parent [Date 1985, Bachman 1969]. Therefore, a network database has two sets of trees: a set of records and a set of links.

A step forward has been the relational model. As proposed by Codd (1970,1990), it "...deals with tuples by means of their information content, not by means extraneous to the tuple such as tuple numbers, tuple identifiers or storage addresses." Furthermore, it requires the existence of a data sub-language based on applied predicate calculus. During the 1980's, the Structured Query Language (SQL) [Hursch & Hursch 1988] became the sub-language of choice. Through SQL and its extensions, the relational model allows the incorporation of semantic information through special constructs such as *associations*, *properties*, and *entities* [Codd 1979, Gardarin & Valduriez 1989], *classes* and *hierarchies* [Blaha et.al. 1988, Goldberg & Robson 1983, Hammer & McLeaod 1980].

The potential of this model for applications other than business has been discussed by many. The work done by Smith & Smith [1977] clearly depicts a relational model that can be manipulated to hold various types of entities, including hierarchy-like entities. Through the concepts of *aggregation*, (which refers to an abstraction in which a relationship between objects is regarded as a higher level object) and of *generalization*, (which refers to an abstraction in which a set of similar objects is regarded as a generic object), it is possible to represent systems that are hierarchical by nature.

Stonebraker, Anton & Hanson (1987) have suggested that the power of the relational model be enhanced to support objects. They proposed that a field in a database be allowed to have a value that is a collection of commands in the query language supported by the relational DBMS. Although this idea tends to violate the atomic property of each datum in relational theory, it seems to be the most appropriate solution to address situations where objects with unpredictable composition are needed.

The prior idea was introduced to the simulation domain by Ketcham (1986) with MBS, where he incorporated function names as part of the information in the hierarchical schema. He configured his custom made DBMS engine in such a way that the engine would look into a procedure field and then into the MBS library to match the content of the field against executable code in the library. MBS has undergone a great deal of refinements and has been renamed from MBS to IBIS [Ketcham Shannon & Hogg 1989].

Ketcham's initial work carried with it the drawbacks of the hierarchical model. To overcome them, Ghoshal (1988) took Ketcham's schema definitions and converted them to a relational-based scheme using the format given by Smith & Smith (1977). Miller & Weyrich, Jr. (1989) have worked on the design and implementation of a simulation environment that integrates a process-oriented simulation language and a database system with object-oriented extensions. Users may interact with the system by formulating SQL-like queries to retrieve information stored from previous simulations.

Centeno (1990) reports the design of an integrated simulation modeling environment (ISME) which utilizes ORACLE, as the RDBMS, SIMAN as the simulation model language, and C as the communication protocols development language.

In the conceptualization of ISME, several tools are vertically integrated through a common user interface. Thus, the full realization of such an environment is based on the design and development of an integrated management system (IMS) that will allow a smooth integration of a collection of software tools for designing, writing and validating simulation models, preparing model input data, analyzing model output data, and designing and carrying out experiments with the model.

Conceptually, ISME provides support throughout the entire modeling process in two ways. First, it provides a network of knowledge-based software tools to support activities in the modeling process, e.g. model conceptualization, solution method(s) selection, statistical analysis, and model building. Second, ISME supports various types of users by providing tools to satisfy individual user requirements, through an interaction mechanism that is natural to the user. For instance, the needs of a system configurer vary greatly from those of a corporate officer or line manager (decision makers.) The system configurer tailors user capabilities to a particular domain, and incorporates domain specific knowledge into the environments knowledge bases. Thus, the system configurer requires access to technical tools such as 4th generation languages and the DBMS itself. In this sense, the system configurer may want to swap back and forth from ISME's common interface to the interface provided by a particular tool. Decision makers, on the other hand, require a fast, easy-to-use, versatile interface that will allow them to access information generated by models, as well as financial data, in order to assess the set of recommendations set forth through the modeling activity. The authors are currently working on ISME and some variations of it.

Another effort is the one at VPI&SU [Balci, et.al. 1990]. They have been working on SMDE, a four layer simulation support environment that uses the OO paradigm as required in the Conical methodology [Nance 1987]. Under this framework a variety of tools are integrated to 1) offer cost effective support throughout the SMP, 2) increase modeler's productivity, and 3) improve the reliability of the simulation studies.

4. CONCLUSIONS

Simulation modeling is a powerful modeling technique that requires various kind of knowledge and tools. Formal simulation modeling methodologies and efficient and smart tools require merging formalisms and techniques from AI and databases. Extensive research has been conducted on the marriage of AI and simulation. The lessons learned lead to the conclusion that it is necessary to complement the power of both simulation and AI with databases so as to develop simulation modeling environments that nurture the entire SMP. Realization of these environments is a function of merging databases, AI and simulation in such a way that the overhead of three fields is minimized. Techniques and tools from AI should 1) guide the search process of the SMP, and 2) serve as a knowledge assistant to the modeler. Databases should hold data from which knowledge is derived by the AI-based tool. Simulation methods and tools act as the catalyst between AI and databases in the quest for comprehensive and smart simulation modeling environments.

REFERENCES

AutoSimulations, 1989, *AutoMod II User's Manual*, AutoSimulations, Inc., Bountiful, Ut.

Balci, O. & et. al., 1990, "Model Generation Issues in a Simulation Support Environment," *Proceedings of the 1990 WSC*, O. Balci, R.P. Sadowski, R.E. Nance (eds), 257-263

Bachman, C.W., 1969, "Data Structure Diagrams," *Journal of the ACM SIGBDP*, 1(2), summer 1969

Birtwistle, G.G. Lomow & B. Unger 1984, "Process Style Packages for Discrete Event Modeling: Using SIMULA's Class," *Simulation*, 1(1), 175-195

Blaha, M., et. al. 1988, "Relational Database Design Using Object-Oriented Methodology," *Communications of the ACM*, 31(4), 414-427

Bobrow, D. & T. Winograd 1984, "An Overview of KRL: A Knowledge Representation Language," Cognitive Science, 1(1), 3-46

Brazier, M.K., 1987, "Automatic Programming of AGVS Simulation Models," *Master's Thesis*, Industrial Engineering Department, Texas A&M University, College Station, Texas, 77843 December

Centeno, M.A., 1990, "Design of an Integrated Simulation Modeling Environment Using a Relational Framework," *Doctoral Dissertation*, Industrial Engineering Department, Texas A&M University, College Station, Texas, 77843 May

Codd, E.F., 1970, "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, 13(6), 377-387

Codd, E.F., 1979, "Extending the Relational Model to Capture More Meaning," *ACM Transactions on Database Systems*, December, 397-434.

Codd E.F., 1990, *The Relational Model for Database Management - Vr. 2*, Addison-Wesley, Reading, Ma.

Dahl, O.J. & K. Nygaard 1966, "Simula: An ALGOL Based Simulation Language," *Communications of the ACM*, Vol. 9, 671-678

Date, C.J., 1985, *An Introduction to Database Systems*, Addison-Wesley, Reading, Ma.

Eklundh, B. 1979, "SIMULA: A Way of Thinking," *Proceedings of the 1979 WSC*, H. Highland, M. Spiegel, & R. Shannon (eds), 11-20

Erickson, S.A. 1985, "Fusing AI and Simulation in Military Modeling," *AI Applied to Simulation*, Proceedings of the European Conference at the University of Ghent, 140-150

Fikes, R. & T. Kehler 1985, "The Role of Frame-Based Representation in Reasoning," Communication of the ACM, 28(9), 904-920

Ford, D.R. & B.J. Schroer, 1987, "An Expert Manufacturing Simulation System," *Simulation*, 48(5), 193-200

Fox, M. & Y. Reddy, 1982, "Knowledge Representation in Organizational Modeling and Simulation: Definition and Interpretation," *Proceedings of the 13th Annual Pittsburgh Conference on Modeling and Simulation*, April

Futo, I. & T. Gergely, 1989, *Artificial Intelligence in Simulation*, Ellis Horwood, London, England

Gardarin, G. & P. Valduriez 1989, *Relational Databases and Knowledge Bases*, Addison-Wesley, Reading, Ma.

Genesereth, M.R. & M.L. Ginsberg, 1985, "Logic Programming," *Communications of the ACM*, 28(9), 933-941

Ghoshal, D. 1988, "Framework for Database and Model base in an Information Based Simulation System," *Master's Thesis*, Industrial Engineering Department, Texas A&M University, College Station, Texas, 77843, August

Goldberg A. & D. Robson, 1983, *Smaltalk: The Language and its Implementation*, Addison-Wesley, Reading, Ma

Gordon, G., 1975,*The Application of GPSS V to Discrete Systems Simulation*, Prentice-Hall, Englewood Cliffs, New Jersey

Haddock, J. & R.P. Davis, 1985, "Building a Simulation Generator for Manufacturing Cell Design and Control," *Annual International Industrial Engineering Conference Proceedings*

Haigh, P.L. & E.M. Bornhorst, 1986, "A User Friendly Environment for Simulating Computer Systems," *Modeling and Simulation on Microcomputers*, C.C. Barnett (ed), 134-140

Hammer, M. & D. McLeod, 1980, "Database Description with SDM: A Semantic Data Model," *TODS*, 6(3), September 1980

Hayes-Roth, F., 1985, "Ruled-Based Systems," *Communications of the ACM*, 28(9), 921-932

Hursch, C.J. & J.L. Hursch, 1988, *SQL: The Structured Query Language*, TAB Book Inc., 1st edition, Blue Ridge Summit, Pa.

Ketcham, M.G., 1986, "Computer Simulation as a Decision Support Tool" *Doctoral dissertation*, Industrial Engineering Department, Texas A&M University, College Station, Tx 77843

Ketcham, M.G., R.E. Shannon & G.L. Hogg, 1989, "Information Structures for Simulating Modeling of Manufacturing Systems," *Simulation*, 52(2), 59-67

Khoshafian, S. & R. Abnous, 1990, *Object Orientation Concepts, Languages, Databases, User Interfaces*, John Wiley & Sons, Inc., New York

Klahr, P., W.S. Faught, & G.R. Martin, 1980, "Rule Oriented Simulation," *Proceedings of the 1980 International Conference on Cybernetics and Society*, IEEE, Cambridge, Ma, 350-354

Klahr, P., 1984, "Artificial Intelligence Approaches to Simulation," *Proceedings of the 1984 UKSC Conference on Computer Simulation*, 87-92

Kung C., 1990, "Object Subclass Hierarchy in SQL: A Simple Approach," *Communications of the ACM*, 33(7), July, 117-125

Mellichamp, J. & A. Wahab, 1987, "An Expert System for FMS Design," *Simulation*, 48(5), 201-208

Miller, J.A. & O.R. Weyrich, Jr. 1989, "Query Driven Simulation Using SIMODULA," *Proceedings of the 1989 Eastern Multiconference,* 22nd Annual Simulation Symposium, 167-181

Mills, R., 1988, "Telecommunication Network Analysis with COMNET II.5," *Proceedings of the 1988 WSC*, M. Abrams, P. Haigh, J. Comfort (eds), 162-164

Miner, R.J. & L.J. Rolston, 1984, "MAP/1 Tutorial," *Proceedings of the 1984 WSC*, Dallas, Tx. November, 59-62

Minsky, M. 1975 "A Framework for Representing Knowledge," *The Psychology of Computer Vision*, P. H. Winston (ed), New York, McGraw-Hill

Murgiano, C., 1990, "A Tutorial on WITNESS," *Proceedings of the 1990 WSC*, O. Balci, R.P. Sadowski, R.E. Nance (eds), 177-179

Murray, K.J., 1986, "Knowledge-Based Model Construction System: An Automatic Programming Approach to Simulation Modeling," *Doctoral Dissertation*, Dept of Computer Science, Texas A&M University, College Station, Texas, 77843

Nance, R.E., 1987, "The Conical Methodology: A Framework for Simulation Model Development," *Proceedings of the Conference on Methodology and Validation*, O. Balci (ed), SCS, San Diego, 38-43

O'Keefe, R., 1986, "Simulation and Expert Systems: A Taxonomy and Some Examples," *Simulation*, 46(1), 10-16

Pegden, C.D. 1984, *Introduction to SIMAN with V2.0 Enhancements*, Systems Modeling Corporation, Calder Square, P.O. Box 10074, State College, Pennsylvania, 16805-0074

Pegden, C.D., R.E. Shannon, & R.P. Sadowski, 1990, *Introduction to Simulation Using SIMAN*, McGraw-Hill, Inc.

Pritsker, A.A.B. & P.J. Kiviat, 1969, *Simulation with GASP II*, Prentice-Hall

Reddy, R., 1987, "Epistemology of Knowledge-Based Simulation," *Simulation*, 48(4), 162-166

Reddy, Y. & M. Fox, 1982, "Knowledge Representation in Organizational Modeling and Simulation: A Detailed Example," *Proceedings of the 13th Annual Pittsburgh Conference on Modeling and Simulation*, April

Reddy, Y., M. Fox & N. Hussain, 1985, "Automating the Analysis of Simulation in KBS," AI, Graphics and Simulation, G. Birtwistle (ed), 34-40

Rich, E. & K. Knight, 1991, *Artificial Intelligence*, McGraw Hill, New York, N. Y., 2nd Edition

Rolston, D.W., 1988, *Principles of Artificial Intelligence and Expert Systems Development*, McGraw-Hill Book Company.

Rothenberg, J., 1989, "Object-Oriented Simulation: Where do we go from here?," *A RAND Note*, N-3028-DARPA, October

Russell, E.C., 1983, *Building Simulation Models with SIMSCRIPT II.5*, CACI Products, La Jolla, Ca.

Schriber, J.J., 1974, *Simulation Using GPSS*, Wiley & Sons, New York

Shannon, R.E. 1984, "Artificial Intelligence and Simulation," *Proceedings of the 1984 WSC*, S. Sheppard, U. Pooch, & D. Pegden (eds), 3-9

Shannon, R.E., R. Mayer & H.H. Adelsberger, 1985, "Expert Systems and Simulation," *Simulation*, 44(6), 275-284

Shannon, R.E., 1986, "Intelligent Simulation Environments," *Intelligent Simulation Environments*, P.A. Luke & H.H. Adelsberger (eds), Simulation Series, SCS, San Diego, Ca., 17(1), 150-156

Smith, J.M. & D. Smith, 1977, "Database Abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems*, 2(2), 105-133

Standridge, C.R., 1985, "Performing Simulation Projects with The Extended Simulation System (TESS)," *Simulation*, 45(3), 283-291

Steudel, H. & T. Park 1987, "STAR*CELL: A Flexible Manufacturing Cell Simulator," *Proceedings of the 1987 WSC*, A. Thesen, H. Grant, W.D. Kelton (eds), 230-234

Stonebraker, M., J. Anton & E. Hanson 1987, "Extending a Database System with Procedures," *ACM TODS*, 12(3), 350-376

Systech, Inc., 1985, *INSIGHT User's Manual*, Indianapolis, Indiana

Tocher, K. & G. Oren 1960, "The Automatic Programming of Simulations" *Proceedings of the 2nd. International Conference on Oper. Research*, 50-68

Tsichritzis, D. & F. Lochovsky, 1976, "Hierarchical Data Base Management: A Survey," *ACM Computing Surveys*, 8(1), March

Zeigler, B.P., 1984, *Multifacetted Modelling and Discrete Event Simulation*, Academic Press, USA

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

Zeigler, B.P., 1989, "DEVS Representation of Dynamic Systems: Event-based Intelligent Control," *Proceedings of the IEEE*, 77(1), 72-80

OTHER RELEVANT REFERENCES

Boyce, R.F. & D.D. Chamberlin 1973, *Using a Structured English Query Language as a Data Definition Facility*. IBM Research Report RJ1318 #20559 December 10, 1973.

Boyce, R.F., et. al., 1975, "Specifying Queries as Relational Expressions: The Square Data Sublanguage," *Communications of the ACM*, 18(11), 621-628.

Centeno, M.A. & C.R. Standridge, 1991,"Modeling Manufacturing Systems: An Information-Based Approach," *24th Annual Symposium 1991, SCS Eastern Multiconference*, April 1-5, 1991, New Orleans, La.

Chamberlin, D.D. & R.F. Boyce, 1974, "SEQUEL: A Structured English Query Language," *Proc. ACM-SIGMOD Workshop on Data Description, Access, and Control*, 249-264.

Clema, J.K. 1980, "Managing Simulation Projects," Simulation with Discrete Models: A State-of-the-art View, T.I. Oren, C. Shub, & P. Roth (eds), SCS, San Diego, Ca.

Oren, T.I. & B.P. Zeigler, 1979, "Concepts for Advanced Simulation Methodologies," Simulation, 32(3), 69-82

MARTHA A. CENTENO is an assistant professor in the Department of Industrial Engineering at the Florida A&M University / Florida State University Joint College of Engineering. She received a B.S. degree in Chemical Engineering from ITESO (Mexico) in 1981, a M.S. degree in Industrial Engineering from Louisiana State University, and a Ph.D. degree in Industrial Engineering from Texas A&M University 1990. Her current research interests are in the areas of intelligent environments for systems analyses. Dr. Centeno is a member of ASA, Alpha Pi Mu, IIE, ORSA, and SCS

CHARLES R. STANDRIDGE is an associate professor in the Department of Industrial Engineering at the Florida A&M University / Florida State University Joint College of Engineering. He led the dvelopment of the Simulation Data Language (SDL) and of The Extended Simulation Support System (TESS) for Pritsker Corporation. His current research interest are in the development of modeling and analysis environments for manufacturing systems, in the use of animation as a modeling and analysis tool, and in the analysis of health care delivery systems.