Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

EXPERT SIMULATION SYSTEM BASED ON A RELATIONAL DATABASE

Robert E. Shannon

Department of Industrial Engineering
Texas A&M University
College Station, Texas 77843

Martha A. Centeno

Department of Industrial Engineering
FAMU/FSU College of Engineering
Tallahassee, Florida 32300

ABSTRACT

The presentation will describe the design and early implementation of a life-cycle oriented simulation environment based on a commercial relational database. The modules and database that make up the environment will be discussed. This paper presents the rationale for such a system and the current state-of-the-art.

1. TRENDS IN LANGUAGE DEVELOPMENT

As the years have passed, new simulation languages have been developed and old ones enhanced. Language developers have focused their attention on three objectives: (1) reduced model development time, (2) improved accuracy, and (3) improved communication. There is, in fact, a revolution taking place and an explosion of creative activity in simulation modeling and language development. This revolution has been brought on to a large degree by developments outside the simulation community. Among these has been the advent of the micro-computer, graphics, and the technologies of artificial intelligence and expert systems.

The ultimate goal of any programming language is to close the gap between what users conceptualize as a representation of a system and how they actually express that relationship in some executable form. The current phase is one in which the development of simulation software is in a significant transition period. The emphasis is upon ease of use and providing an *integrated simulation environment* rather than simply more powerful languages. Henriksen [1984] says, "An integrated simulation environment is a collection of software tools for designing, writing, and validating models; writing and verifying simulation programs (implementing models); preparing model input data; analyzing model output data; and designing and carrying out experiments with models." He further points out that an important difference between a truly integrated approach and that promoted by traditional simulation languages or specialized tools is feedback. The simulation environment should include a mechanism whereby results of the experiment phase are feed back to the model construction phase. It is through this feedback mechanism that task closure is achieved without unduly interrupting the simulationist's train of thought.

The current language development activity is motivated by two problems as well as the opportunity presented by the confluence of developments in several areas of technology. It is clear to all practitioners that doing simulation modeling studies tends to be a low productivity activity. Even expert practitioners are repeatedly surprised by how much effort is required to accomplish a useful result. The traditional modeling life-cycle is very labor intensive and time consuming. The computer is used mainly for execution of the model while the construction of the model, design of the experiments to be run, and analysis of the results are shouldered by the modeler. Furthermore, since most of the "intelligent" functions are performed by hand, a great deal of knowledge is discarded at the completion of each project. Most models are treated as "throw away" items as is the analysis required to solve the problem [Zeigler 1984; Ören 1986].

Giddings [1984] has argued that in order for significant changes in the man-power intensive nature of model development to be realized, the development process itself must be redefined. Attempts must be made to "conserve" effort by incorporating knowledge and expertise into the software. With this approach, as much work as possible will be shifted from the human to the machine, thus freeing the modeler for handling decisions that truly require the "human" touch.

Closely allied to the problem of low productivity is the fact that simulation modeling as practiced today requires a high level of training, and even then, is as much black art as science. Thus, there is a severe shortage of trained and experienced personnel. Clema [1980] observes that " ... we have a small number of artisans with proven track records in consulting, industry, academia, and government, who are generally successful, and a second group of practitioners, large in number, who perhaps have not acquired the necessary skills and experience to achieve the success desired by the customer."

This is not surprising since today, in order to use modeling correctly and intelligently, the practitioner is required to have expertise in a number of different fields. This generally means separate courses in probability, statistics, design of experiments, modeling, computer programming, and a modeling language. This translates to about 720 hours of formal classroom instruction plus another 1,440 hours of outside study (more than 1 man-year of effort) and that is only to gain the basic tools. In order to really become proficient, the practitioner must then gain real world, practical experience (hopefully under the tutelage of an expert). Adding time to learn about the systems being studied, further increases the human investment. As a result, the cost of modeling is often prohibitive because of the continuing reliance on expensive human analytical skills. The goal for the development of new modeling systems is to make simulation modeling less of a "black art" and possible for engineers, scientists, and managers to do modeling studies correctly and easily without such elaborate training.

In addition to the above problems the current increased development activity is motivated by the fact that there have recently emerged certain new opportunities each of which, if properly exploited, has the potential to exert an influence of historic proportions. The first of these is the desktop computing revolution. The emergence of desktop computers which can address 16MB or more of RAM memory, operating at 16 to 20MHz speeds, opens up the possibilities for putting fantastic power at the fingertips of managers and engineers. The micro-computer has freed them from the tyranny of the central computer services department. No longer must simulations be run on the third shift so as not to interfere with the payroll, accounting, etc. In addition, access to the micro-computer means that every organization, no matter how small, can now afford the hardware and software to perform simulation studies. Almost all of the most popular simulation languages are now available on micro-computers.

Yet another opportunity is presented by the remarkable progress in the field of knowledge representation and database management during the last decade, especially the explosive emergence of relational and object oriented database technology. Most serious expert systems (ES) and simulation modeling applications deal with large amounts of data and need to access that data efficiently. The development of excellent database programs and the evolution of sophisticated inquiry interfaces are natural adjuncts to data-hungry ES and modeling programs.

Finally, the progress being made in artificial intelligence (AI) technology opens the door for a rethinking of the simulation modeling process for design and decision support. The problem solving paradigm as currently practiced in simulation modeling studies is essentially a search. 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. The burden of conducting that search and integrating the results of model behavior into a coherent solution currently rests with the human user. Much of the work in the AI field on automated reasoning has dealt with the inference engine and how to use the facts and rules contained in the knowledge base to conduct an efficient search for a solution. These developments will play a vital part in the design of new goal seeking, knowledge based simulation systems.

2. EXPERT SYSTEMS

Beginning with the mid 1980's AI and Expert Systems technology gained a lot of attention from the simulation community because they seemed to offer many advantages over more traditional approaches to simulation. The potential of expert systems and AI in simulation has been explored by Shannon [1984, 1986, 1989], Shannon et al. [1985], Reddy [1987], and O'Keefe [1986] among others. Several languages based upon AI methodologies have been developed. Adelsberger and Neumann [1985] explored the use of PROLOG as a simulation language. ROSS developed at the RAND Corporation [Klahr 1984] and KBS developed at Carnegie-Mellon [Reddy et al. 1985], are examples of simulation languages based upon AI concepts.

Use of AI techniques for simulation environments can be found in Ketcham and Harhen [1986] who proposed an integrated network of expert systems to handle the decision making process in simulation studies. Zeigler [1987] has taken advantage of the compatibility between object oriented programming paradigms and discrete simulation formalisms to develop an environment for model construction. Guariso et al. [1989] designed and prototyped a knowledge based system that interactively generates simulation models.

Despite their potential, however, AI-based systems have been regarded by many as an expensive, sometimes inefficient solution. More recently, a number of researchers have taken the approach to developing intelligent front and back ends around existing commercial languages [e.g., Brazier and Shannon 1987; Haddock and Davis 1985; Murray 1986; among others]. Conceptually, merging the best of these two fields would yield a knowledge-based simulation environment in which a simulation model is synthesized by accepting a description of the model and consulting the appropriate knowledge base. This implies that generic descriptions of the objects to describe the system are stored in long term fashion within the computer for use whenever needed. The latter has brought a third player into the game: databases.

3. DATABASES

The use of databases, in conjunction with simulation modeling, began just recently. TESS [Standridge et al. 1985; Grant and Starks 1988] integrates simulation, data management, graphics, and animation on top of a network database. TESS represents a new generation of software that integrates model building, model execution, analysis and presentation of results. Ketcham [1986, 1987, Ketcham et al. 1989] developed MBS which has undergone several refinements and has been renamed from MBS to IBIS. IBIS is based upon a hierarchical database which makes it a little hard to use. Each application program must know exactly how the data is stored, plus the user must define the schema of each record type needed to describe the system to be modeled. In addition, most corporate database management systems today are relational rather than hierarchical.

The concept of the relational data model, as proposed by Codd [1970], is simple, yet robust. A relational database is "a database that is perceived by its users as a collection of tables (and nothing else but tables)," [Date 1986]. The relational model does not imply nor does it address specific physical approaches for data storage and retrieval. Merging simulation and the relational model was explored by Yancey [1987], who investigated the idea of a relational database management system centered environment to properly collect and manipulate data generated by simulation runs.

Ghoshal [1988] took Ketcham's [1986] schema definitions and converted them to relational-based schema using the format given by Smith and Smith [1977]. Ghoshal showed that you could superimpose a hierarchical structure upon a relational database by applying the concepts of "aggregation" which refers to an abstraction in which a relationship between objects is regarded as a higher level) and of "generalization," (which refers to an abstraction in which a set of similar objects is regarded as a generic object) to pure relational tables.

The concepts of generalization and aggregation in databases are the equivalent of classes and objects in simulation and in AI. Aggregation is easily achieved through the normalization process. These two concepts, generalization and aggregation, provide the means to treat normalized tables as objects representing different levels of abstraction.

4. CONCLUSIONS

Simulation, AI, and the relational data model are just beginning to be studied as a triplet. Centeno [1990] describes an integrated simulation environment that is under development which is based on ORACLE™, a commercial relational database management system. The Integrated Simulation Modeling Environment (ISME) uses the relational database as the only means to represent the objects in a simulation model, the relationships between them and some elements of the user interface.

The flexibility, robustness, and elegance of the relational model have caused it to be widely used in industry and it is available over the entire gamut of computer hardware. Furthermore, not only does the use of the relational model allow the development of a combined ordinary database and simulation system, but it offers a suitable framework for the construction of a knowledge base for a KBS.

REFERENCES

Adelsberger, H.H. and G. Neumann (1985), "Goal Oriented Simulation Modeling Using PROLOG," In *Proceedings of the 1985 SCS Conference on Modeling and Simulation on Micro-Computers*, San Diego, CA, 42-47.

Brazier, M.K. and R.E. Shannon (1987), "Automatic Programming of AGVS Simulation Models," In *Proceedings of the 1987 Winter Simulation Conference*, A. Thesen, H. Grant, and W.D. Kelton, Eds. IEEE, Piscataway, NJ, 703-708.

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

Clema, J.K. (1980), "Managing Simulation Projects," In *Simulation with Discrete Models: A State-of-the-Art View*, T.I. Ören, C. Shub, and P. Roth, Eds. SCS, San Diego, CA.

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

Date, C.J. (1986), *An Introduction to Database Systems, Volume 1*, Fourth Edition, Addison Wesley, Reading, MA.

Ghoshal, D. (1988), "Framework for Database and Modelbase in an Information Based Simulation System," M.S. Thesis, Industrial Engineering Department, Texas A&M University, College Station, TX.

Giddings, R.V. (1984), "Accommodating Uncertainty in Software Design," *Communications of the ACM 27*, 5.

Grant, M.E. and D.W. Starks (1988), "A Tutorial on TESS: The Extended Simulation Support System," In *Proceedings of the 1988 Winter Simulation Conference*, M.A. Abrams, P.L. Haigh, and J.C. Comfort, Eds. IEEE, Piscataway, NJ, 136-140.

Guariso, G.H., H. Martin, and H. Werthmer (1989), "An Intelligent Simulation Model Generator," *Simulation 53*, 2, 57-66.

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

Henriksen, J.O. (1984), "Discrete Event Simulation Languages: Current Status and Future Directions," In *Proceedings of the 1984 Winter Simulation Conference*.

Ketcham, M.G. (1986), "Computer Simulation as a Decision Support Tool," Doctoral Dissertation, Industrial Engineering Department, Texas A&M University, College Station, TX.

Ketcham, M.G. and J.G. Harhen (1986), "Expert Systems and User Decisions in Simulation Studies," In *Languages for Continuous System Simulation*, F.E. Cellier, Ed. SCS, San Diego, CA, 44-49.

Ketcham, M.G. (1987), "IBIS: Information-Based Integrated Simulation: The Design of a Simulation Model Base for Manufacturing," In *Simulation in CIM and Artificial Intelligence: Proceedings of the European Simulation Multiconference*, J. Retti and K. Wichmann, Eds. SCS, San Diego, CA, 26-30.

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

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

Murray, K.J. (1986), "Knowledge-Based Model Construction System: An Automatic Programming Approach to Simulation Modeling," Ph.D. Dissertation, Department of Computer Science, Texas A&M University, College Station, TX.

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

Ören, T.L. (1986), "Knowledge Bases for an Advanced Simulation Environment," In *Intelligent Simulation Environments*, SCS *Simulation Series 17*, 1, 16-22.

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

Reddy, Y.V., M. Fox, and N. Hussain (1985), "Automating the Analysis of Simulation in KBS," In *AI, Graphics, and Simulation*, G. Birtwistle, Ed. SCS, San Diego, CA, 34-40.

Rozenblit, J.W., T.G. Kim, and B.P. Zeigler (1988), "Towards an Implementation of a Knowledge-Based System Design and Simulation," In *Proceedings of the 1988 Winter Simulation Conference*, M. Abrams, P. Haigh, and J. Comfort, Eds. IEEE, Piscataway, NJ, 226-230.

Shannon, R.E. (1984), "Artificial Intelligence and Simulation," (keynote address), In *Proceedings of the 1984 Winter Simulation Conference*, S. Sheppard, U.W. Pooch, and C.D. Pegden, Eds. IEEE, Piscataway, NJ, 3-9.

Shannon, R.E. (1986), "Intelligent Simulation Environments," In *Proceedings of the Conference on Intelligent Simulation Environments*, P.A. Luker and H.H. Adelsberger, Eds. SCS *Simulation Series 17*, 1, 150-156.

Shannon, R.E., R. Mayer, and H.H. Adelsberger (1986), "Expert Systems and Simulation," *Simulation 44*, 6, 275-284.

Shaw, M.L.G. and B.R. Gaines (1986), "A Framework for Knowledge-Based Systems Unifying Expert Systems and Simulation," In *Proceedings of the Conference on Intelligent Simulation Environments*, P.A. Luker and H.H. Adelsberger, Eds. SCS *Simulation Series 17*, 1, 38-43.

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

Standridge, C.R., D.K. Vaughan, and M.L. Sale (1985), "A Tutorial on TESS: The Extended Simulation System," In *Proceedings of the 1985 Winter Simulation Conference*, D.T. Gantz, G.C. Blais, S.L. Solomon, Eds. IEEE, Piscataway, NJ, 73-79.

Yancey, D.P. (1987), "Database Management Systems Can Provide Way to Manage Information Generated in a Computer Simulation Program," *Industrial Engineering 19*, 5, 50-53.

Zeigler, B.P. (1984), "Multifaceted Modeling Methodology: Grappling with the Irreducible Complexity of Systems," *Behavioral Science 29*, 169-178.

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