Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

GOAL DRIVEN SIMULATION INTELLIGENT BACK ENDS: A STATE OF THE ART REVIEW

Louis A. Molina
Carlos Gandarillas
Martha A. Centeno

Department of Industrial and Systems Engineering
College of Engineering and Design
Florida International University
Miami, Florida 33199, U.S.A.

ABSTRACT

Goal driven simulation (GDS) seeks to automate many of the output analysis and experimental design tasks of a simulation study. Theoretically, its use allows the reallocation of the simulation expert to other tasks. GDS capabilities include determining parameters to change, suggesting a rate of change, and testing these changes against a pre-established set of goals. Realizing GDS, however, requires the integration of techniques such as object oriented design, knowledge based systems, and neural nets. Before achieving this integration, there are still several issues to resolve including the type of interaction these techniques would have among themselves. This paper explores several of the issues concerning the realization of goal driven simulation systems, their impact on the simulation modeling methodology, how GDS works, and the need for its development.

1 INTRODUCTION

Simulation is the process of designing a computer model of a real system, and conducting experiments with this model, to understand the behavior of the system and/or evaluating various strategies for the operation of the system (Centeno and Shannon, 1995). Activities involved in a simulation study require the modeler to posses knowledge in various disciplines, such as modeling, probability and statistics, computer programming, and process analysis (Alsugair and Chang, 1994). In many instances, the modeler finds herself or himself not an expert in one or two of these areas; thus, the scope of the study is limited, or its delivery date gets extended. Of the various tasks in a simulation study, the experimentation and output analysis are the ones that lead to a real solution, but they are also among the most repetitive and time consuming tasks after model verification (Ford and Schroer, 1987). Unless the modeler has extensive experience with the type of systems being modeled, s/he find her/himself testing a large set of alternatives before a desired performance goal is achieved.

Goal driven simulation (GDS) incorporates the use of a knowledge based expert system in conjunction with a simulation language to achieve an optimal solution, within a set of desired objectives. Thus, GDS would reduce the time needed for experimentation and analysis, and it would allow the shifting of some modeler resources to other tasks in the simulation study. The latter has led us to believe that GDS research shall experience a significant growth during the next decade.

The need for GDS has been reflected in works in different fields such as the analysis of flexible manufacturing systems (FMS) (Kopacsi and Kovacs, 1993), tutoring of simulation users (Touran, 1990), manufacturing of electronics (Ford and Schroer, 1987), and the general work by Shannon, Mayer and Adelsberger (1985). These works clearly describe that two of the areas of the simulation modeling process that need great improvement are the design of experiments and output analysis areas. Traditional simulation languages fall short of providing adequate support; however, these authors point out that the integration of knowledge-based techniques with traditional tools will lead to the development of an "ideal" simulation modeling environment.

Advances in hardware and simulation software have allowed simulation modeling to be utilized in very diverse sectors. In fact, many simulation software companies offer packages targeted to specialized areas within industry. However, such expansion has not been paralleled by similar advancements in automated output analysis (Centeno and Shannon, 1995). It is not until recently that some simulation packages have begun to include capabilities for the analysis of data the simulation model uses and/or produces.

In Section two, we discuss some of the reasons why GDS is needed. In Section 3, we review various research efforts in this area and discuss a framework that we believe contains the minimum elements required for a GDS simulation system. In Section 4, we discuss the impact that GDS development will have on the simulation modeling methodology itself.

2 THE NEED FOR GDS

To better understand the need for GDS, it is necessary to first understand what the current simulation methodology can do. In this context, simulation modeling is a process that involves the designing, building, verification, and validation of a model, with which the modeler will experiment to explore a variety of "*what*-if" scenarios. Simulation can provide estimates for a variety of measures of performance, and it can evaluate the effects of changes to the operating conditions. However, traditional simulation modeling methodology can not optimize, it can not describe or deal with system characteristics that have not been previously described to it, and it can not solve problems just provide information from which solution may be inferred. Thus, it is clear that the power of this technique could greatly benefit from technique that will enable it to optimize and seek automatically for solutions.

The GDS philosophy offers a framework in which a variety of data modeling and knowledge managing techniques could be incorporated. GDS will search for the appropriate set of inputs that would yield measures of performance within a desired interval; thus, enabling simulation optimization. The GDS approach is only restricted by the knowledge contained in the knowledge base. Further, under GDS, the simulation environment would automatically run multiple alternatives, over a wider search space, using smarter search techniques, while taking into consideration the objectives of the study. Thus, the model user will have appropriate support when doing the analysis of several alternatives to obtain trends or patterns of the system's performance.

GDS also requires that domain knowledge be part of the knowledge base. This knowledge enables the GDS to support the modeler even when the complexity of the system under study grows because it will guide him or her on how to modify parameters to achieve desired goals.

In summary, the need for the GDS approach is paramount because

- ✓ traditional simulation methodology can not optimize.
- ✓ the complexity of simulated systems has grown.
- ✓ solutions tend to be dependent on the expertise of the model user.

Current commercial simulation languages do not support optimization methodologies based on domain knowledge that relieves the analyst from multiple scenarios. SimRunner©, a package that works with ProModel, performs optimizations on models for this language (Akbay, 1996). However, this is not domain knowledge based search, but statistical searches over a multidimensional parameter field. In this sense, the package falls short from providing the user with a pseudo-intelligent search.

Similarly, other packages such as SIMAN/ARENA support only tools that aid the analyst in performing the study of each scenario. However the capability of creating an integrated model and knowledge base is not found in the current versions. GPSS and AUTOMOD excel in other areas but fall short in the optimization process.

3 GOAL DRIVEN SIMULATION

In traditional simulation analysis, the user conducts experiments by varying inputs to the model or by changing the model itself. On the other hand, goal driven simulation (GDS) drives itself to achieve a set of desired goals (Shannon and Prakash, 1990). Under GDS, the user inputs the simulation model, its capabilities, his priorities, and goals of the simulation study. Goals are the desired value(s) for one or more of the systems measures of performance. Target values to be met may include average time in the system, resource utilization, and average number in queue. GDS would determine what parameters should be changed, and test these parameters, to ensure that they lead to the desired goals.

Research in the area of goal driven simulation has focused on two major areas. These are 1) building new simulation systems altogether, and 2) building hybrid systems. The first area calls for a redefinition and expansion of the simulation modeling methodology to incorporate the strengths of other disciplines such as artificial intelligence (AI); however, in many instances, the resulting simulation system has re-invented the wheel in the sense that they have had to duplicate what traditional simulation packages do well in order to attain minimal value added. This situation has been mostly caused by the incompatibility of AI-based tools with more traditional programming languages. Fortunately, software developments are taking us to a point in which robust AI-based tools can truly be combined with powerful traditional simulation languages.

The second area has seen a little bit more success because it calls for the addition of modules to traditional simulation packages, so as to have a simulation modeling environment that operates under the GDS philosophy.

The addition of intelligent back-ends (IBE) responsible for assessing the outputs of the simulation model against the user-provided goals seems to be a very promising alternative for GDS (Shannon and Prakash, 1990). Such back ends could operate in one of two ways:

- ✓ _Series_: The IBE is triggered as soon as the simulation model has finished generating the data of the first experiment. Then, the IBE assesses such output against the goals. If the outputs fall within the prescribed range, then the experiment is complete; otherwise, the IBE will utilize its knowledge base to decide how to modify the input parameters, so as to steer the simulation model towards the desired goal. Once the model provides new outputs, the IBE repeats the process for as long as needed.
- ✓ _Parallel_: Both the back end and the simulation model are running simultaneously. The simulation model produces a series of traces from which estimates for the measures of performance are continuously updated. The IBE is attentive to such estimates and, as soon as it can determine that the values of these estimates are heading outside the target, it will halt the simulation run, modify the input parameters, and re-start the simulation run.

In both cases, the IBE should have the capability of recognizing a situation where the goal(s) can not be met, no matter how the IBE changes the input parameter set, so as to avoid for the process falling in an "infinite loop."

In the parallel mode, the GDS environment must be capable of on-line triggering of the inference engine of the IBE at strategic points in time. The simulation model would still process events using the *next-event* time advance approach, but it will have a *special* event (that occurs at "some" points in time) which call the IBE engine right before the simulation clock is advanced. Thus, allowing for an *on-line* modification of input parameters (Farimani-Toroghi and Peck, 1990)

The evolution of expert systems tools has made it possible to embed knowledge-based systems (KBES) into other software. Similarly, data and knowledge representation techniques have also advanced (e.g. object oriented concepts are readily available in many languages). These advancements make the second of the two areas of GDS more promising. In fact, these advancements should help us move forward from the efforts preceding GDS (such as that by Bengu and Haddock (1986), Haddock and O'Keefe (1990)) to fully functional GDS environments.

It is worth noting, however, that at the heart of GDS is the incorporation of analysis and optimization techniques that are glued in such a way that the IBE evaluates various alternatives and relates them to the study objectives. Tompkins and Azadivar (1995) propose the use of genetic algorithms to solve the optimization of some simulation models.

Based on our previous discussion, a GDS system is formed by three main components: 1) A knowledge-based expert system, 2) an *open* simulation language, and 3) an object oriented programming language. Each component is composed of a various modules depending on its role of each component. Figure 1 gives a schematic representation of a GDS framework.

The simulation language for such a system would be preferably an object oriented system. This allows the description of the different parts of the model by means of properties that can be easily modified. This allows tangible changes in the representation of the system by removing and adding complete objects (Farimani-Toroghi and Peck, 1990). Among existing simulation languages, SIMAN and SLAM have proven to be capable of interacting with external tools (Ford and Schroer, 1990). However, with the advent of shared platforms and enhanced communications protocols, most software developed using object oriented design and an object oriented programming language (OOPL) would be able to be linked to other packages.

The contents of the KBES include historical runs, priority rules, statistical knowledge, capability rules, and model representations. These modules can be defined as follows

- ☑ _Historical runs_ enable the collection of historic information of the system. The results from the modification of parameters and the output from the simulations are transformed into information expressed as cause-and-effect rules that could be use in later consultations.

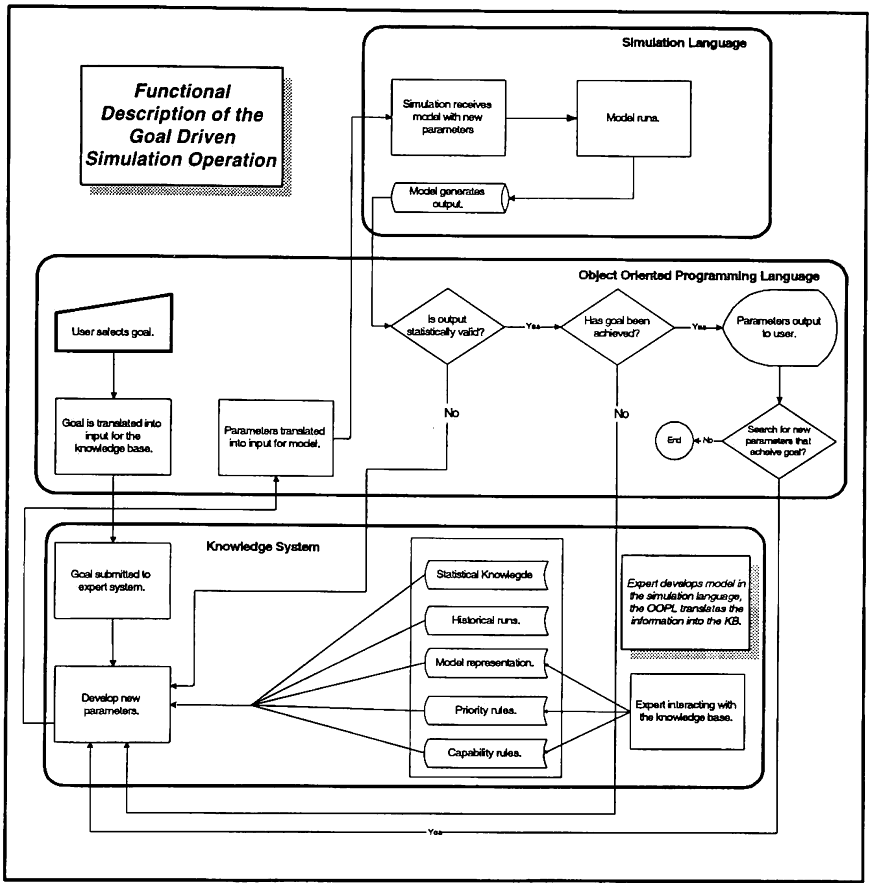

Figure 1: Proposed Framework for a GDS System

- ☑ *Priority rules* bound the search space according to a user given ranking of the input parameters. For example, a priority rule could guide the IBE to maximize the utilization of cheaper resources before increasing the capacity of the more expensive ones.
- ☑ *Statistical Knowledge* is needed to assess the closeness of outputs to the target, as well as to identify the changes required.
- ☑ *Capability rules*, or system capacity rules, define the allocation schemes of available system resources. These type of rules limits the search space to values that are realistic to the system under study.
- ☑ *Model representations* describe the model in terms of operational rules, including precedence relationships, entity flows, and any physical constraints fixed to the system. Works in this area include that by Benjamin et al. (1994) and Farimani-Toroghi and Peck (1990). The first work presents the application of an Air Force module call IDEF3 where descriptions of the model were successfully captured. The second work uses the production rule schema to represent some of the operational characteristics of the system.

Under GDS, the process is initiated by the user who seeks to satisfy a goal, bound by the restrictions previously encoded in the knowledge base. A model of the system is coded in the simulation language. It is then verified and validated. The goal requirements are passed to the knowledge base, which in turn would be accessed by the KBES to generates the model's parameters. These parameters are fed to the model via the OO user interface, and the simulation runs are triggered. Once this is obtained, runs can be executed to determine if the goals have been met. The historical component stores the results of the different runs in order to base the decision making process on actual performance of the model under different parameter settings. This removes the analyst from a great deal of repetitive calculation.

The use of a reasoning mechanism in GDS provides several advantages. The use of model specific information can describe the model using cause and effect relationships that relate the inputs to the outputs of the model. These inputs can then be manipulated to obtain a goal. The use of this approach, is what differentiates GDS from statistical methodologies or exhaustive studies of different scenarios.

4 IMPACT OF GDS ON THE SIMULATION METHODOLOGY

The simulation modeling methodology as described by Banks et al. (1996) would undergo a fundamental change if GDS were to be implemented. The study of several modified models would no longer be necessary to obtain the desired parameters. By avoiding the output analysis, and driving the simulation intelligently to the optimum scenario, there is no longer need for excessive hours of computational effort. The steps in a simulation study that would be impacted are experimental design and analysis of outputs.

4.1 The Experimental Design Stage

Under the experimental design stage, the user will not need to test the alternatives that possibly satisfy the desired goal. Instead, the user would input the goal and the system will provide a solution, or solutions, that satisfy the goal. Before any runs can be made, the interface would translate these goals into input for the KBES, which will establish what measures of performance need to be tracked to initiate the search towards the goal. This will initiate the search for a valid solution based on the information in the knowledge base. No longer will the model user be confronted with the situation in which a complete set of production runs are done just to find out during the analysis stage that a variable was left out of the data collection process.

4.2 Production Runs and Analysis

After the model is run, and given that the results of the model are statistically significant,, the outputs are analyzed and compared to the goal. If they meet or exceed the requirements of the goal, they are reported to the user. The user may then decide to have the system search for other parameters which yield the desired goal. In the case where the goal has not been achieved, the information is translated into historical knowledge for the expert system, triggering a new search for parameters. The new parameters are integrated into the simulation model for new runs to be performed.

4.3 The new role of the simulation model

The use of GDS not only will improve the cycle for studies, but will also change the notion of the model as a disposable entity. By creating a framework around the model, it becomes a dynamic entity. It will be allow modifications that are reflected in the knowledge base. This provides a tool that can continuously be monitored to reflect the performance of the real system. As changes in the parameters are varied, the system will provide feedback on expected performance. The investment in such a system will also allow the users to modify any physical representation of the model, that may be planned in the future.

5 SUMMARY

GDS are geared towards minimizing the need of human interaction during the analysis of output information; thus, reducing the risk of error. The traditional simulation methodology requires the analyst to perform what-if analysis. This trial and error approach requires the user to perform repetitive tasks such as statistical analysis of output. The focus of the modeler is shifted from the statistical output analysis to capability analysis. Capability analysis would encompass the study of all possible values for the assignment of resources within the system. In addition, users other than the modeler would have access to a system that would yield answers to "what-if" questions, without requiring from them an in-depth technical knowledge to develop a complete simulation study.

However, to realize GDS, further research is needed in the areas of classification of the types of systems to simulate, domain knowledge acquisition, and the integration of heterogeneous tools that may be rooted in heterogeneous platforms.

REFERENCES

Alsugair, A. and D. Y. Chang. 1994. A Goal Driven Approach to Discrete Event Simulation. In *Proceedings of the First Congress Held in Conjunction with A/E/C Systems*. 1: 515-522.

Akbay, K. 1996. Using Simulation Optimization to Find the Best Solution. IIE Solutions. 28(5):24-29.

Banks, J., J. S. Carson, II, and B. L. Nelson. 1996. *Discrete-Event System Simulation*. New Jersey: Prentice Hall.

Bengu, G. and J. Haddock. 1986. A Generative Simulation Optimization System. *Computers in Industrial Engineering*. 10(4): 301-313.

Centeno, M. A. and R. E. Shannon. 1995. Integrating AI and Databases in Application to Simulation. *Advances in Databases and Artificial Intelligence* 1: 311-332.

Erraguntla, M., P. C. Benjamin and R. Mayer. 1994. An Architecture of a Knowledge-Based Simulation Engine. In *Proceedings of the 1994 Winter Simulation Conference,* eds. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 673-680..

Farimani-Toroghi, S. and S. Peck. 1990. A Framework for Combined Simulation and Knowledge Based Systems. In *Expert Systems, Artificial Intelligence, Neural Networks, Vision, Manufacturing Modeling and Simulation, Proceedings of the Annual Pittsburgh Conference*, 903-907. SCS, San Diego, California.

Ford, D. R. and B. J. Schroer. 1987. An Expert Manufacturing Simulation System. *Simulation* 48: 193-200.

Haddock, J. and R. M. O'Keefe. 1990. Using Artificial Intelligence to Facilitate Manufacturing Systems Simulation. *Computers in Industrial* Engineering. 18: 275-283.

Kopacsi, S. and G. L. Kovacs. 1993. A Knowledge Based Simulation System for Flexible Manufacturing. In *Proceedings of the IEEE International Conference on Robotics and Automation,* 858-863. IEEE, Los Alamitos, Ca.

Lenat, D. B. 1995. Artificial Intelligence. *Scientific America.* 273 : 80-82.

Murray, K. J. and S. Sheppard. 1988. Knowledge-Based Simulation Model Specification. *Simulation* 50 : 112-119.

Romero, N. R. and L. Interrante. 1944. An Expert System Using Simulation Measures of Effectiveness. In *IEEE International Conference on Systems, Man, and Cybernetics* 3 : 2201-2206.

Schena, G. D. 1994. Objective-Driven Simulation as Tool for Plant Design- Application to Cyclone Selection with Cost-Minimization Function. *Transactions. Section C, Mineral Processing and Extractive Metallurgy* 103 : C155-C162.

Shannon, R. E., R. Mayer, and H. Adelsberger. 1985. Expert Systems and Simulation. *Simulation* 44 : 275-284.

Shannon, R. E. and S. Prakash. 1990. Goal driven Simulation Systems. In *1990 International Industrial Engineering Conference Proceedings.* 345-350.

Tibbitts, B. R. 1993. Flexible Simulation of a Complex Semiconductor Manufacturing Line Using a Rule-Based System. *IBM Journal of Research and Development* 37 : 507-521.

Tompkins, G. and F. Azadivar. 1995. Genetic Algorithms in Optimizing Simulated Systems. In *Proceedings of the 1995 Winter Simulation Conference,* eds. C. Alexopoulos, K. Kang, W.R. Lilegdon, and D. Goldsman, 757-762.

Touran, A. 1990. Integration of Simulation with Expert Systems. *Journal of Construction Management* 116 : 480-493.

AUTHOR BIOGRAPHIES

MARTHA A. CENTENO is an assistant professor in the Department of Industrial and Systems Engineering at Florida International University. She received a B.S. degree in Chemical Engineering from ITESO University (Mexico), an M.S. degree in Industrial Engineering from Louisiana State University in 1985, and a Ph.D. degree in Industrial Engineering from Texas A&M University in 1990. Her research interests are in the areas of intelligent simulation, smart data analysis, and engineering education. Dr. Centeno is a member of Alpha Pi Mi, IIE, INFORMS, SCS, ASA, and ASEE.

CARLOS GANDARILLAS is a graduate student at Florida International University, where he is pursuing a Master of Science degree in Industrial Engineering. His research interests are in the area of smart simulation. He is expected to graduate in August of 1997. He received a Bachelor of Science degree in Industrial and Systems Engineering from Florida International University in 1995. He is a member of Alpha Pi Mu.

LOUIS A. MOLINA is a graduate student at Florida International University, where he is pursuing a Master of Science degree in Industrial Engineering. His research interests are in the area of goal driven simulation and knowledge acquisition, from databases, for statistical modeling. He is expected to graduate in August of 1996. He received a Bachelor of Science degree in Industrial and Systems Engineering from Florida International University in 1994 and is a member of Tau Beta Pi and IIE.