1981 Winter Simulation Conference Proceedings
T.I. Ören, C.M. Delfosse, C.M. Shub (Eds.)

SIMULATION MODEL MANAGEMENT: RESOLVING THE TECHNOLOGICAL GAPS

Richard E. Nance
Ahmed L. Mezaache
C. Michael Overstreet
Department of Computer Science
Virginia Tech
Blacksburg, Virginia 24061

Model management poses requirements and responsibilities that extend throughout the life cycle of a simulation model. Recent publications have identified major problems in cost and time overruns, which are traceable to deficiencies in project and sponsor management. Beginning with the division of the simulation model life cycle into seven phases, we define "model management" and develop the requirements for a Model Management System (MMS). The functional description of a MMS focuses on those phases that jointly characterize the model development effort. Recent research in simulation model development is described, and particular emphasis is given to the approach taken with the Conical Methodology.

1. Identifying the Technological Gaps

Recognition of modeling as the basic tool of problem solving has continued throughout the life of operations research and management science. That modeling is a fundamental requirement in the use of simulation as a problem-solving technique is undeniable. However, equally undeniable is the lack of attention to problem solving and modeling as fundamentals in the teaching of mathematical sciences (applied mathematics, statistics, operations research, management sciences, and computer science). The tendency to treat modeling as an art and problem solving as almost an "unteachable" is clearly recognized in the recent interesting paper of Woolley and Pidd [1].

Throughout the past two decades, the mathematical sciences have found their techniques and modeling approaches the subject of criticism at various times. In the past five years the "computerized model", irrespective of the underlying technique, has drawn particularly indicting criticism. A GAO study of Federal computerized models categorized their deficiencies as primarily stemming from management shortcomings. These shortcomings have led to high cost, long development times, and limited utility; and the mismanagement is attributed frequently to the sponsoring agency as well as the developing agency [2]. The conclusions of the GAO report caused individuals and groups both within and outside the Federal model development and user communities, to review their approaches to the modeling activity. This research initiated with just such a review.

2. Understanding the Contributing Causes

An immediate tendency is to attribute the inadequacies of computerized models to the widely claimed difficulties in software development. To use software development inadequicies as a scapegoat is simply to ignore both the conclusions of the GAO study and the clearly apparent gaps in the teaching of modeling and problem solving. The analogy between software development and the model development processes is a good one, and unfortunately it lends some undeserved credibility to the claim.

Nor do we subscribe completely to the tempting explanation that problems with computerized models indicate the same trap that enveloped the programming community during the past decade. This trap is the delusion that very large (or complex) models are developed by the extrapolation of the same techniques applicable to producing small (or simple) models. No doubt, the delay in recognizing this fallacy has been a contributing cause, but it does not suffice to completely explain the problems.

We adopt the view that a computer program can be a model representation. Despite the fact that few descriptions of the program development process

THE SYSTEM

Either physical or real, the system and the study objectives provide the reference for the model and the modeling task.

PHASE 1: CONCEPTUAL MODEL

The conceptual model is that model which exists in the mind of the modeler. The form of the conceptual model is influenced by the system, the perceptions of the system held by the modeler (which are affected by the modeler's background and experience and those external factors affecting the particular modeling task), and the objectives of the study.

PHASE 2: COMMUNICATIVE MODEL

A model representation which can be communicated to other humans can be judged or compared against the system and the study objectives by more than one human. Several communicative models could be constructed during a study, each derived from a preceding communicative model (following the first) or different conceptual models. Entity cycle diagrams are examples of communicative models.

PHASE 3: PROGRAMMED MODEL

A programmed model is a model representation that admits execution by a computer to produce simulation results. It is a communicative model from which experimental results are obtained. SIMSCRIPT or SIMULA programs are examples of programmed models.

PHASE 4: EXPERIMENTAL MODEL

The programmed model and the executable description of the test environment (the experimental frame of Zeigler [ZEIGB76]) form the experimental model.

PHASE 5: MODEL RESULTS

The results phase includes the outcome from a single execution of the experimental model or those results produced to satisfy a single test scenario, which might require several model executions with different input value specifications, structural changes, etc.

PHASE 6: INTEGRATED DECISION SUPPORT

Integrated decision support extends the experimental domain to multiple scenario executions, indexed and accessible either by automatic, manual, or combined analysis so as to permit the recognition of behavioral features or trends unspecified in the individual tests. The recognition of untested but interesting test scenarios is possible. Analysis permits the extrapolation or prediction of untested scenarios based on prior results.

PHASE 7: MODIFIED MODEL

Modification represents a significant change in the model definition and specification from the original. The change might be caused by an extension of function of the model or restatement of the study objectives.

Fig. 1

can be recognized as similar to the model development process, we are in complete agreement with Lehman [3], who holds this view. We also find Lehman's taxonomy of programs to be instructive in characterizing some of the problems in model development. Taking this view, we hold the perspective of model development and usage as controlled, disciplined, computer assisted, and planned. The primary difficulty with computerized models is traceable to the lack of discipline and control in the development stages and a reliance on nonexistent or outdated planning and project management aids. In other words, difficulty stems from the missing ingredient: "model management."

The notion of managing the model development might sound strange. Ten years ago it would have sounded ludicrous. But when a simulation model is used to develop a national blueprint for energy policy [2] or when the simulation project costs extend into the millions of dollars as is now the case, the challenge is properly described as "model management."

3. Drawing a Clearer Focus

The model development process and the methodological approach to simulation modeling and experimentation have drawn increasing attention over the past three years. No doubt, our own ideas have profited from the published works of Zeigler [4;5], Ören [6;7], Elzas [8], Mathewson [9;10], and the DELTA project [11].

Our perspective of model management and its support through a Model Management System (MMS) begins with the definition of the model life cycle. Since the definition of the software life cycle served as an important precursor to a more complete understanding of software development technology, we believe that the model life cycle is the proper point to begin correction of the difficulties cited above.

We describe the model life cycle as comprised of seven phases, which are presented in Figure 1. While the transition of a modeling effort from one phase to another is unlikely to be clearly defined, the existence of each of the first five phases seems assured. The later two phases are, or rather should be, realizable in especially large, costly projects. Explicit recognition of the seven phases spanning the useful life of a simulation model assists in explaining and appreciating the objectives of model management and the consequent requirements of a MMS.

Phases 1-5 represent the effort most commonly called "model development." By examining the descriptions of each phase in simulation model development, one can begin to perceive the inadequacies of the tools provided for simulation modeling. The difficulties in model validation, are better illustrated by recognizing that unless model varification is carefully done in each phase succeeding the conceptual model, relation of the experimental model to the conceptual model is nearly impossible. The final two phases, integrated decision support and the modified model, are evident in very large projects but rarely made explicit. However, good model management requires their recognition and inclusion as explicit phases.

4. A Model Management System

4.1 Definition.

> A Model Management System (MMS) is a set of tools that assist in the efficient creation and use of an effective model whose application is expected to extend in scope and time beyond the original study objectives.

The assistance provided by the MMS begins with the conceptual model and extends throughout the succeeding six phases. The MMS provides an organization and integration of data; an accessible data base of prior modeling efforts categorized by multiple logical schema; and, through its monitoring and data gathering ability, creates information needed for the planning of future modeling and simulation projects. More detail on the forms of assistance provided by the MMS are given in the following paragraphs.

4.2 Users of a Model Management System.

We characterize the MMS as supporting a modeling organization, that is a unit whose primary function is the development of and experimentation with models. Such a unit might be internal or external to a parent organization it might function in either a consultative or client role. We avoid these details for clarity, but they are not ignored in the scenario that follows.

The user group is made up of five user types:

(1) The organization manager, who supervises

(2) several project managers, who manage project teams made up of

(3) analysts, primarily responsible for the model definition and specification, the experimental design, the data definition and organization, and the presentation and interpretation of model results, who are supported by

(4) simulation software development managers. (The chief programmer position to use the terminology of Mills [12]), who conform the data definition and organization to the logical and physical requirements of a data base management system and computer system configuration, and instruct

(5) programmers, who develop the coded model representation in an executable language.

This categorization of users imposes five levels of responsibility in the model development activity. Not all units would necessarily have this many levels; in fact, the combination of certain levels is quite plausible and probably to be expected. However, this delineation seems reasonable and is helpful in the distinction of users for whom MMS requirements are intended.

NEEDS REQUIREMENTS OF A MODEL MANAGEMENT SYSTEM

OBJECTIVES OF MODEL MANAGEMENT	REQUIREMENTS OF A M.M.S.	USER
1. Given a problem that requires a model or models to reach a solution, produce an effective model with an efficient effort that concludes in a reasonable time.	1. (a) Provide access to previously developed models and submodels, data, experimental designs, model results, and implementation results.	All
	(b) Furnish tools for model development that: (i) speed the realization of a communicative model, (ii) assist in assuring model correctness, (iii) coordinate the experimental design from the outset, (iv) relieve the modeler of programming nuances in the production of the programmed model, and (v) enable model documentation to be an integral product of the development effort.	Analyst
	(c) Provide for the scheduling of milestones monitoring of progress, and control of costs in the production of a model.	Organization Manager Project Manager
	(d) Assist in the realization of a model representation in an executable language.	Programmer
2. Permit the use of models to range from long-term policy formulation and strategic planning to short-term (quasi-real-time) decision making. (To use Kiviat's terminology: model directed management.)	2. (a) Enable direct interaction of managers with models.	Organization Manager
	(b) Promote the understanding of models at different levels--high level for "top" managers, intermediate level for analysts, and detailed level for analysts and programmers	All
	(c) Promote a progressive learning of model details guided by the manager/user.	All
3. Enable information to be obtained from completed, in-progress, or planned modeling efforts to meet: (1) legal or jurisdictional requirements, or (2) prediction/planning needs.	3. (a) Organization of a database for prior modeling projects that can be interrogated by status, resources used, and other characteristics related to model description and prediction/planning.	All Project Manager Analyst
	(b) Representation of submodels, models, experimental design, experimental outcomes, and model modifications.	All

Table 1

THE NEEDS REQUIREMENTS AND FUNCTIONAL DESCRIPTION OF A MODEL MANAGEMENT SYSTEM UTILIZING THE CONICAL METHODOLOGY

REQUIREMENTS FOR APPLYING THE CONICAL METHODOLOGY	FUNCTIONS OF A MODEL MANAGEMENT SYSTEM UTILIZING THE CONICAL METHODOLOGY
1. Interactive environment for model development.	1. (a) Dialogue (User Interface) (b) Editor (c) File handling (storage, retrieval, etc.) (d) Tutorial on system
2. Model development capability.	2. (a) Model definition using a SMSDL. (b) Model specification using a SMSDL. (c) Production of stratified (multi-level) documentation (d) Tutorial on CM
3. Primitive representation (removing SPL conditions) that reveals (captures) the <u>object description</u> and the <u>object relationships</u> consistent with the objectives of the modeling study.	3. (a) Translation to primitive representation
4. Assistance in determining model correctness--verification of model.	4. (a) Diagnosis of primitive representation (measures) (i) completeness (ii) consistency (iii) relative complexity
5. Monitoring of the model development process.	5. (a) Accounting (b) Control (management definition of milestones) (c) Database management
6. Test and execution for verification and validation.	6. (a) Execution of submodels (b) Execution of the model
7. Data definition, storage, analysis.	7. (a) Database management (b) Statistical analysis (c) Data translation
8. Modification and reuse of models.	8. (a) Model archival (includes representation) (b) Database management

Table 2

The existence of a client organization utilizing the modeling unit is also possible. This client organization would be expected to have corresponding levels of modeling responsibilities. Thus, the requirements of a MMS intended to serve users in the modeling unit would also serve users at the corresponding level in the client organization.

4.3 Objectives of Model Management.

This section advances our objectives of model management from which we construct the requirements of a model management system and the consequent functional description. We have tried to reduce the number of objectives; consequently, each objective is rather comprehensive in scope. The elaboration and expansion takes place in the requirements definition and the subsequent functional description. This is in keeping with the topdown design approach to which we subscribe.

The best model is the least cost model that accomplishes the objectives of the study. This rather trite statement leads to the following objective of model management.

> Objective One. Given a problem that requires a model or models to reach a solution, produce an effective model with an efficient effort that concludes in a reasonable time.

The second objective recognizes the considerable differences in scope and investment in models that are used for short duration and in models which are intended for major investments or extensive use.

> Objective Two. Permit the use of models to range from long term policy formulation and strategic planning to short time (quasi real-time) decision making.

This objective paraphrases the goal of model directed management set by Kiviat [13] several years earlier.

The difficulty in estimating time and cost for software development is acknowledged and has received much attention. Modeling and model development seem to add another layer of difficulty to the task. What is needed is accurate data taken from ongoing projects concerning the efforts and particular activities contributing to time and cost overruns. The third objective relates to this need.

> Objective Three. Enable information to be obtained from completed, in progress, or planned modeling efforts to meet: (1) legal or jurisdictional requirements, or (2) prediction/planning needs.

4.4 Requirements of a Model Management System.

Working from the objectives of model management, we have composed the needs requirements of model management shown in Table 1. The particular user level affected by each requirement is identified. Note that a MMS has as its components: (1) a data base management subsystem, (2) an extensive dialogue module providing the vehicle for producing a communicative model from a conceptual model, (3) a software development subsystem, (4) a documentation production subsystem, (5) an experimental analysis subsystem, (6) a knowledge based development subsystem, and (7) an internal monitoring and accounting subsystem. These subsystems provide one functional partitioning of a MMS, but it is not necessarily the partitioning that best defines the task of creating such a system. We view the functional partitioning as a crucial step in the eventual realization of a MMS because the interfaces among subsystems can provide a designers nightmare due to numerous interconnections or a users white elephant because of excessive overhead.

5. The Conical Methodology in a Model Management System

A recent report [14] describes the conical methodology (CM), intended for model development in discrete event simulation. Earlier papers [17,15] have alluded to the CM and argued the need for such a methodological approach. The cited report [14] describes the context in which simulation model development now exists, explains the need met by the CM in relation to other approaches, sets forth the definitions forming the foundation of the CM, and illustrates its use in developing a single model. The report concludes with an incomplete critique of the CM and the model representation produced by it.

Our intent in this paper is not to repeat much of what is included in [14]. Rather, we wish to sketch the relationship between the Conical Methodology and the requirements imposed by it on a Model Management System as well as its contributions to a MMS.

Table 2 presents the requirements for applying the CM and the functional capability of a MMS utilizing the methodology. This tabular presentation clearly illustrates that the CM is a methodology, and its implementation is that which truly supports the model development process. The crucial components of a CM implementation are: 1) a primitive representation providing a description of objects and object relationships devoid of the usual syntactic and semantic influences of higher level languages, and (2) a simulation model specification and documentation language (SMSDL) by which model definition and specification can proceed from the conceptual through the communicative phase to a primmitive representation. The characteristics of a SMSDL are described in an earlier [15]. The recent work of Frankowski and Franta [16] propose a SMSDL based on the process oriented world view.

Please note that the CM supports only the model development phases of the simulation model life cycle. The MMS must provide functions for support of phases 6 and 7, and the relationship between these phases and the product of the model development phases must be properly structured in the functional partitioning of a MMS. At this time we do not have a clear view of the most beneficial functional partitioning to guide the further work in the creation of a MMS.

6. Summary

"Model management" is an accurate description of the challenge inherent in current larger, complex simulation experiments. Computer assistance is essential in meeting this challenge, but a Model Management System must address all phases of the model life cycle. The Conical Methodology serves as one guide in structuring the model development tasks (phases 1-5 of the life cycle) in a more axiomatic fashion. However, the implementation of the Conical Methodlogy depends on the solution of three major problems:

(1) the construction of a primitive representation that enables model diagnostics,

(2) the definition of a Simulation Model Specification and Documentation Language and the production of a language translator, and

(3) the creation of an innovative, powerful dialogue system, well designed for the model development requirements.

Our research is currently addressing all three problems, but the first continues to be emphasized.

References

1. Wooley, R.N. and M. Pidd. "Problem Structuring--A Literature Review," Journal of the Operational Research Society, 32 (3):1981, 197-206.

2. U.S. Government Accounting Office, "Ways to Improve Management of Federally Funded Computerized Models," LCD-75-111, Washington, D.C., August 23, 1976.

3. Lehman, M.M. "Programs, Programming and the System Life Cycle," Report No. 80/6, Department of Computing Science, Imperial College, London, April 15, 1980.

4. Zeigler, Bernard P. "Modeling and Simulation Methodology: State of the Art and Promising Directions," Simulation of Systems '79, L. Dekker et. al. (eds), North Holland, 1980.

5. Zeigler, Bernard P. "Concepts and Software for Advanced Simulation Methodologies," Simulation with Discrete Models: A State-of-the-Art View, Oren, et. al. (eds), IEEE, 1980.

6. Ören, Tuncer I. and Bernard P. Zeigler, "Concepts for Advanced Simulation Methodologies," Simulation, 32 (3):March 1979, 69-82.

7. Ören, Tuncer I. "Computer-Aided Modeling Systems (CAMS)," Plenary Address, Simulation '80 Symposium, Intertaken, Switzerland, June 25-27, 1980.

8. Elzas, M.S. "What Is Needed for Robust Simulation?" Methodology in Systems Simulation, B.P. Zeigler, et. al. (eds.), North Holland, 1979, 57-91.

9. Mathewson, S.C. "Computer Aided Simulation Modeling and Experimentation," Proceedings of the Eighth Australian Computer Conference, 1978, 9-13.

10. Mathewson, S.C. "Integrated Computer Simulation Modeling," (privately communicated), 1979.

11. Holbaek-Hanssen, E., P. Handlykken and K. Nygaard. "System Description and the DELTA Language," Report No. 4 (Publication no. 523) second printing, Norwegian Computing Center, Oslo, 1977.

12. Mills, H. "Chief Programmer Teams, Principles, and Procedures," IBM Federal Systems Division Report FSC 71-5108, Gaithersburg, Maryland, 1971.

13. Kiviat, Philip J. "Model Directed Management," address at the Virginia Computer Users Conference, Blacksburg, Virginia, March 1977.

14. Nance, Richard E. "Model Representation in Discrete Event Simulation: The Conical Methodology," Technical Report CS81003-R, Department of Computer Science, Virginia Tech, March 15, 1981.

15. Nance, Richard E. "Model Representation in Discrete Event Simulation: Prospects for Developing Documentation Standards," Current Issues in Computer Simulation, N. Adam and A. Dogramaci (eds), Academic Press, 1979, 83-97.

16. Frankowski, Elaine N. and W.R. Franta, "A Process Oriented Simulation Model Specification and Documentation Language," Software--Practice and Experience, 10 :1980, 721-742.

17. Nance, Richard E. "The Feasibility of and the Methodology for Developing Federal Documentation Standards for Simulation Models," Final Report to the National Bureau of Standards, Department of Computer Science, Virginia Tech, 26 June 1977.