Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

THE USE OF GRAPHICAL MODELS IN MODEL VALIDATION

ROBERT G. SARGENT
Department of Industrial Engineering
and Operations Research
Syracuse University
Syracuse, NY 13244

ABSTRACT

The use of graphical models for model specification and in modelling is increasing rapidly. This paper discusses the use of these graphical models in model validation.

1. INTRODUCTION

Validating a model is an integral and important part of model development. Model validation is usually defined to mean "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" (Schesinger et al. 1979) and is the definition used here. Substantiating that a model is valid is a process which consists of performing tests and evaluations within the (total) model development process to determine whether a model is or is not valid. (For an expository on model validation, see, e.g., Gass (1983) and Sargent (1985) and for a recent bibliography, see Balci and Sargent (1984).)

Research has related model validation to specific steps of the model development process. We will relate model validation to a simplified revision of the modelling process as shown in Figure 1 (Sargent 1985). The *problem entity* is the system (real or proposed), idea, situation, policy, or phenomena to be modelled; the *conceptual model* is the mathematical/logical/verbal representation (mimic) of the problem entity developed for a particular study; and the *computerized model* is the conceptual model implemented on a computer. The conceptual model is developed through an *analysis and modelling phase*, the computerized model is developed through a *computer programming and implementation phase*, and inferences about the problem entity are obtained by conducting computer experiments on the computerized model in the *experimentation phase*. *Conceptual model validity* is defined as determining that the theories and assumptions underlying the conceptual model are correct and that the model representation of the problem entity is "reasonable" for the intended use of the model. *Computerized model verification* is defined as ensuring that the computer programming and implementation of the conceptual model is correct. *Operational validity* is defined as determining that the model's output behavior has sufficient accuracy for its intended purpose or use over the domain of the model's intended application. *Data validity* is defined as ensuring that the data necessary for model building, model evaluation and testing, and conducting the model experiments to solve the problem are adequate and correct.

In this paper, we discuss the use of graphical models in model validation. Graphical models are discussed in section 2 and their use in model validation in section 3. Section 4 is the summary.

2. GRAPHICAL MODELS

The use of graphical models for model specification and in modelling is rapidly growing. This is occurring primarily because new interactive computer graphic software systems for developing graphical models of systems are rapidly becoming available at reasonable costs. The reason that these new interactive graphic software systems are rapidly becoming available is that the cost of developing interactive computer graphic software systems for low cost computer hardware has recently become relatively inexpensive. These new graphic software systems are being developed for and being used on personal computers, workstations, and graphic devices (terminals) that tie into mainframe computers. Other reasons for the rapidly growing use of graphical models are their use reduces the amount of time and effort required to develop models and new types of graphical models are being developed.

Graphical models have different characteristics. They frequently are directed graphs which have different types of nodes that are connected by directed arcs (or edges). Graphical models can be computer language dependent or independent and they can be used manually on paper and on computers if computer graphic software systems have been developed for them. Interactive computer graphics software systems can have different types of graphical interfaces for using graphical models: the icon/window approach, the menu/form approach, and a combination of both approaches. Preprocessors exist in some of the interactive computer graphic software systems to generate computer codes from the graphical models. (For a discussion on graphical programming of simulation models, see Browne, Dutton, and Neuse 1986.)

Graphical models are themselves not new. For example, block diagrams of integrators, summers, etc. for the use in modelling on analog computers, block diagrams of GPSS (Schriber 1974) for discrete event simulation, and flow diagrams of DYNAMO (Pugh 1970) for industrial dynamics modelling have been used manually for numerous years. What is new are the wide variety of new graphical models becoming available and the (new) interactive computer graphic software systems being developed for graphical models.

One group of graphical models which can be used in model validation is those for functional decomposition. An example from this group is DeMarco diagrams used in structural analysis (DeMarco 1978). The diagrams form a hierarchical set, with lower levels expanding in more detail, on the functions and data of the upper level. The diagrams are numbered in a manner which shows their location in the heirarchy. This decomposition can continue, if desired, down to algorithms and to pseudo code. Figure 2 gives two (generic) functional levels as an example. Interactive computer graphic software systems containing graphical models for functional decomposition are available.

Another group of graphical models that can be used in model validation is those for discrete event simulation. This is a large group which has considerable variety. We will only discuss a representative sample. Two types of graphical models which have existed for a number of years are block diagrams and (process) networks. Examples of block diagrams are the GPSS Block Diagrams (Schriber 1974) and the SIMAN Block Diagrams (Pegden 1985). Examples of networks are SLAM networks (Pritsker 1984) and INSIGHT networks (Roberts 1984). These graphical models are computer language dependent and were (initially) developed to be used manually in developing models of specific systems. Recently, interactive computer graphic software systems have been developed for some versions of these graphical models and some of them contain preprocessors to automatically generate the computer code. A new graphical model called an event graph has recently been developed for simulation models that have the event world view (Schruben 1983 and Sargent 1985). This graphical model is language independent and is the first one for simulation models that use the event world view. Currently event graphs have to be developed manually as no software systems exist for them; however, some experimental software systems for them are under development. Figure 3 contains an event graph model of simple flexible manufacturing system from Sargent (1985). The nodes are the events and the edges show under what conditions additional events and what events are to be generated. Another new graphical model is the Information Processing Graph (IPG) that is used in the interactive graphic software system called GPSM (Graphical Programming of Simulation Models) that can, if desired, automatically create software for PAWS, the Performance Analysis Workbench System (Browne, Dutton, and Neuse 1986 and Information Research Associates 1985). Figure 4 contains an IPG of a simple computer system. The nodes are the resources and the arcs connect the resources to show the flow of the transactions among the resources. Icons and

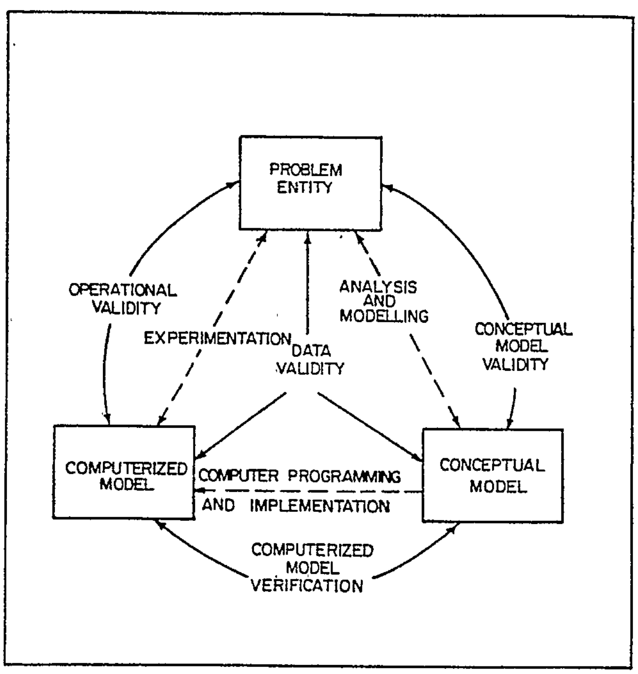

Figure 1. Simplified Version of the Modelling Process

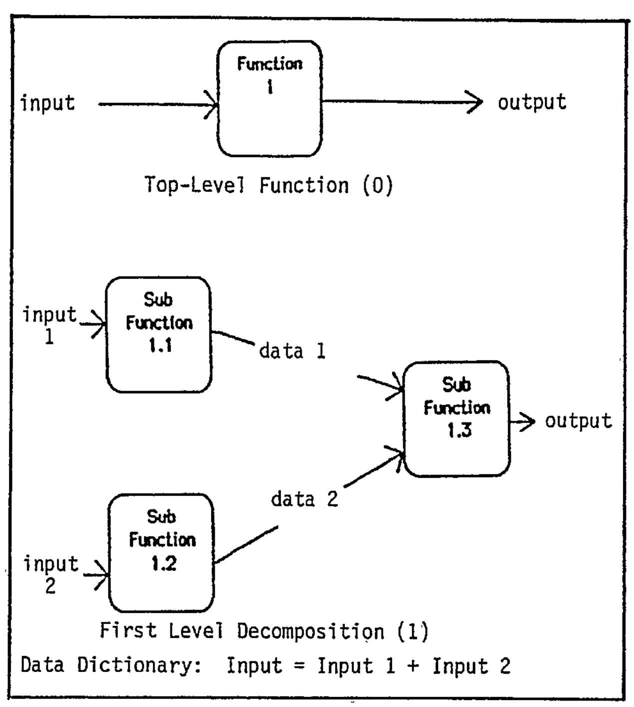

Figure 2: DeMarco Diagram for Functional Decomposition

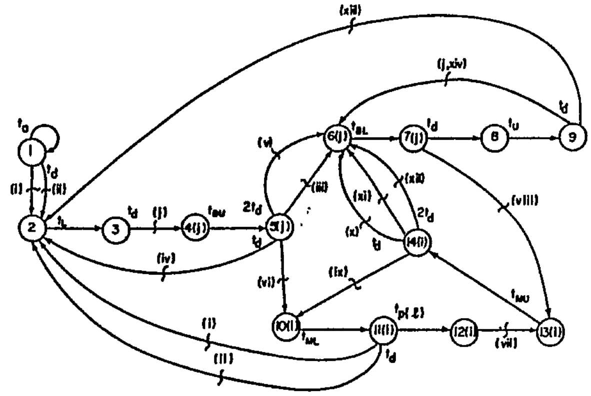

Figure 3. Event Graph of a Simple Flexible Manufacturing System

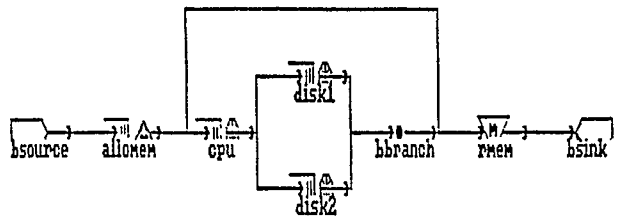

Figure 4. IPA of a Simple Computer System

a mouse are used to build an IPG and pop-up menus and windows are used for labeling and providing additional information about the nodes and arcs, e.g., queue discipline and service time distributions.

The primary purpose of this section is to discuss the graphic models being referred to in this paper. No attempt is being made to present a survey of graphical models. Graphical models other than those in the two groups just presented can be used in model validation, e.g., the block diagrams mentioned above for use in developing models for analog computers. Computer animation is not a graphical model (at least as we are using the term) and thus it is excluded in this work. (Computer animation is useful in model validation; for a discussion on the use of animation in model validation, see, for example, Sargent 1985 and Standridge 1986.)

3. MODEL VALIDATION

Model validation is a process that continues throughout model development. It involves substantiating that a model is valid. Graphical models can help in this substantiation. Specifically, graphical models can be most helpful with conceptual model validity which was defined in Section 1.

Different types of representation are available for conceptual models. These include graphical models, textual descriptions, mathematical equations, flowchart models, and computer codes including simulation language codes. Conceptual models can and frequently do have different levels of representation. The levels of representation can be from a high level down to a detailed level. A high level representation will, for example, identify and show the relationships among subsystems or identify and show the relationships among the high level system functions. A detailed or low level model will contain the details of the model logic including its algorithms. The common practice in modelling is to start at a high level of representation and develop lower and lower levels of representation until a detailed level of representation is obtained. The number of levels used depend upon the complexity of the system being modelled. Different types of representation can be used for different levels of representation. (Do not confuse the levels of representation of a model with the amount of detailed *contained in a model*, i.e., its fidelity or accuracy.)

Conceptual model validity includes the determination that the (conceptual) model representation of the problem entity or system is "reasonable" for the intended use of the model. This determination of "reasonableness" includes determining if an appropriate model structure has been used, if appropriate detail and aggregate relationships have been used for the model's intended purpose, and if the correct logic, algorithms, and mathematical and causual relationships have been used. The primary validation technique currently used to determine if the model representation is reasonable for the model's intended use is face validation. Face validation is having an expert or experts on the problem entity (e.g., the system) evaluate the model to determine if they believe it is reasonable and correct for its intended purpose. Face validity occurs after the model has been developed by the modeller and may be used for different levels of model representation.

The type of representation used for conceptual models affects the amount of effort and time required for face validation. Graphical models rank at or near the top of the different types of representation when they are ranked on the minimum amount of effort and time required to perform face validation. The primary reason for this is that the understanding of a conceptual model is usually relatively straight forward and clear when appropriate graphical models are used. The saying that a picture is worth a thousand words applies here. Different levels of representation of a conceptual model may use different types of graphical models. For example, higher level representations may use DeMarco diagrams and a low level representation may use event graphs.

Davis (1986) has recently suggested an alternative way to use system experts in (conceptual) model validation. He suggests using seminar gaming to simultaneously do model specification and model validation. Model specification is the developing of a conceptual model that can be used to program a computerized model. Different levels of model specification can be developed which is similar to different levels of representation discussed above. In seminar gaming, experts in the system simultaneously develop and validate each level of the model with the modeller. This contrasts with face validation where the modeller initially develops the model and then the experts determine whether the model is or is not valid.

The use of graphical models in seminar gaming should be extremely effective. We have already discussed that graphical models are usually the desired type of representation for face validation of conceptual models. Graphical models are also preferred in most cases for model specification; in particular, when an interactive computer graphical software sysem can be used for developing the model specifications (Browne, Dutton, and Neuse 1986). Thus, graphical models should work extremely well when system experts and modellers together do conceptual model specification and validation simultaneously.

Graphical models can also help in model validation in other ways. As stated above, graphical models usually provide a better understanding of conceptual models with less effort than the other types of representation. Thus, graphical models should be helpful in operational validity and in providing higher confidence in models of nonobservable systems. In operational validity, different configurations of the model and numerous values of the model's parameters are usually used; for example, in performing sensitivity analysis, degenerate tests, and extreme-condition tests (Sargent 1985). Graphical models should help (1) in conveying and understanding the relationships among the configurations, parameter values, and model behaviors and (2) reduce the effort required to perform operational validity if an interactive computer graphic software system is being used that can easily modify the model's configuration and parameter values.

If a system is unobservable (i.e., data cannot be collected on its behavior), comparisons cannot be made between model and systems behaviors and therefore high confidence in the model usually cannot be achieved (Sargent 1985). It is therefore *extremely* important that the conceptual model be *thoroughly* understood, that conceptual model validity be performed as *well* as possible, and that the model's behavior be *thoroughly* understood in operational validity to obtain as much confidence as possible in the model. Graphical models should be more helpful in accomplishing this than the other forms of model representation.

A topic related to model validation is model credibility. Model credibility is concern with having models and their results accepted by the (potential) users. Graphical models should be helpful in model credibility for the same reasons they are useful in model validation.

4. SUMMARY

The rapid growth in the use of graphical models for model specification and in modelling is beneficial to model validation. The primary reason is that graphical models is usually the preferred type of representation of conceptual models for both conceptual model validity and operational validity. As more interactive computer graphic software systems are developed that allow models to be easily developed, the model's computer code to be automatically generated, the computer codes easily run, and the model's results to be automatically displayed, it will be easy to ignore model validation. It should be remembered that high confidence usually cannot be obtained in a model unless a comparison of the model's and system's input-output behavior has been made for different sets of experimental conditions.

ACKNOWLEDGEMENT

This work was sponsored by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700.

REFERENCES

Balci, O. and Sargent, R.G. (1984). A Bibliography on the Credibility, Assessment, and Validation of Simulation and Mathematical Models. *Simuletter 15, 3*, 15-27.

Browne, J.C., Dutton, J.E., and Neuse, D.M. (1986). Introduction to Graphical Programming of Simulation Models. In: *Proceedings of the 1986 Summer Computer Simulation Conference*, Reno, Nevada. Society for Computer Simulation, 32-37.

Davis, E.A. (1986). Use of Seminar Gaming to Specify and Validate Simulation Models. In: *Proceedings of the 1986 Winter Simulation Conference* (J. Wilson, J. Henriksen, and S. Roberts, eds.). Washington, D.C.

DeMarco, T. (1978). *Structured Analysis and System Specification.* Yourdon, Inc., New York, New York.

Gass, S.I. (1983). Decision-Aiding Models: Validation, Assessment, and Related Issues for Policy Analysis. *Operations Research 31*, 601-631.

Information Research Associates (1985). Graphical Programming of Simulation Models: Introduction and Technical Summary. Austin, Texas.

Pegden, C.D. (1985). *Introduction to SIMAN*. Systems Modeling Corporation, State College, Pennsylvania.

Pritsker, A.A.B. (1984). *Introduction to Simulation and SLAM II.* Second Edition. Halsted Press, New York, New York.

Pugh, III, A.L. (1970). *DYNAMO II User's Manual.* The MIT Press, Cambridge, Massachusetts.

Roberts S.D. (1984). Simulation with INSIGHT. In: *Proceedings of the 1984 Winter Simulation Conference* (S. Sheppard, U.W. Pooch, C.D. Pegden, eds.) Dallas, Texas. 23-32.

Sargent, R.G. (1985). An Expository on Verification and Validation of Simulation Models. In: *Proceedings of the 1985 Winter Simulation Conference* (D. Gantz, G. Blais, S. Solomon, eds.). San Francisco, California. 15-22.

Sargent, R.G. (1985). Event Graph Modelling for Simulation With an Application to Flexible Manufacturing Systems. Working Paper #85-005, Department of Industrial Engineering and Operations Research, Syracuse University, Syracuse, New York 13244 (Revised December 1985).

Schlesinger, S., et al. (1979). Terminology for Model Credibility. *Simulation 32*, 103-104.

Schriber, T.J. (1974). *Simulation Using GPSS*. John Wiley and Sons, New York, New York.

Schruben, L.W. (1983). Simulation Modeling with Event Graphs. *Communications of the ACM 26*, 957- 963.

Standridge, C.R. (1986). Animating Simulation Using TESS. *Computers and Industrial Engineering 10*, 121-134.

AUTHOR'S BIOGRAPHY

ROBERT G. SARGENT is a Professor of Industrial Engineering and Operations Research and a member of the Computer and Information Science Faculty at Syracuse University. Dr. Sargent has served the Winter Simulation Conferences in several capacities, including being a member of the Board of Directors for ten years, Board Chairman for two years, General Chairman of the 1977 Conference, and Co-editor of the 1976 and 1977 Conference Proceedings. Professor Sargent was Department Editor of Simulation Modeling and Statistical Computing for the *Communications of the ACM* for five years, has served as Chairman of the TIMS College on Simulation and Gaming, and has received service awards from ACM, IIE, and the Winter Simulation Conference Board of Directors. He currently is an ACM National Lecturer, a member of the Executive Committee of the IEEE Computer Society Technical Committee on Simulation, and a Director-at-large of the Society for Computer Simulation. Dr. Sargent received his education at the University of Michigan. His current research interests include model validation, simulation methodology, simulation applications, performance evaluation, and applied operations research. Professor Sargent is a member of AIIM, the New York Academy of Sciences, Sigma Xi, ACM, IIE, ORSA, SCS, and TIMS, and is listed in Who's Who in America.

Robert G. Sargent
Department of Industrial Engineering
and Operations Research
Syracuse University
Syracuse, New York 13244-1240
(315)423-4348