Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

SIMULATION MODEL DESIGN

Paul A. Fishwick

Dept. of Computer & Information Science
University of Florida
Bldg. CSE, Room 301
Gainesville, FL 32611

ABSTRACT

In this state of the art talk, we will present a structure for defining and categorizing simulation model designs. In the past, simulation researchers have created categories for discrete event simulation: event, process and activity; however, there are problems with this breakdown. First, the major problem is that the taxonomy based on these three sub-types deals with only discrete event methods. Discrete time methods including a spatial decomposition of a physical system (cellular automata, L-Systems) or a continuous model are not included. Second, the terms "event," "process" and "activity" create a division among classes of simulation languages, rather than a division based on model design. The term "process," for example, is really a level of abstraction higher than "event" and is not *orthogonal* to "event." The structure that we present in this talk is more comprehensive and provides simulationists with a unified framework that is independent of the terms *discrete* and *continuous*.

1 OVERVIEW

Simulation is a tightly coupled and iterative three component process composed of 1) *model design*, 2) *model execution* and 3) *execution analysis* as shown in Fig. 1 along with the relevant sub-areas and book chapter numbers in a book which has just been published in the Fall of 1994 (Fishwick 1995). The bold lines in Fig. 1 are to show our emphasis in the text: *model design* and *model execution*. The third area of *execution analysis* already has broad coverage in simulation and is not covered in the book or this state of the art talk. Also, in this talk, we will cover model design and not algorithms for model execution. Perhaps the hardest general problem in simulation is determining the exact method that one should use to create a model. After all, where does one begin? Just as the discipline of software engineering has emerged to address this question for software, in general, model-

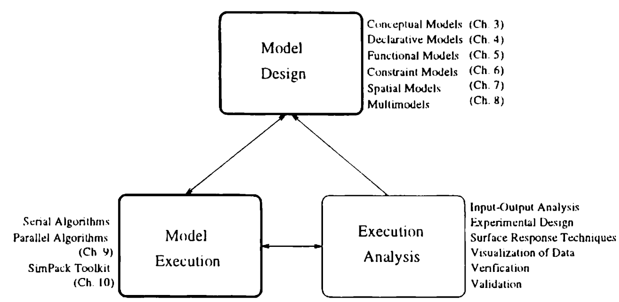

Figure 1: The study of computer simulation.

ers also have a need to explore similar issues: *how do we engineer models*? While there are many modeling techniques for simulation, we are often in a quandary as to which model technique to use, and under what conditions we should use it. Our approach is depicted in Fig. 2 along with the associated chapter references where each modeling method is defined. For the talk, we will proceed to briefly discuss the model types. A more complete written treatment is provided in Fishwick (1995).

2 MODEL TYPES

There are five basic model types, and one complex model type which includes abstraction levels, each composed of one of the basic types. All model types are now discussed.

2.1 Conceptual Models

Conceptual models represent the first phase in any modeling endeavor. All static and dynamic knowledge about the physical system must be encoded in some form which allows specification of interaction without necessarily specifying the dynamics in quantitative terms. Semantic networks (Woods 1975) present one way of encoding conceptual semantics; however, we have chosen object-oriented design networks (Booch 1991; Rumbaugh, Blaha, Premerlani, Frederick, and

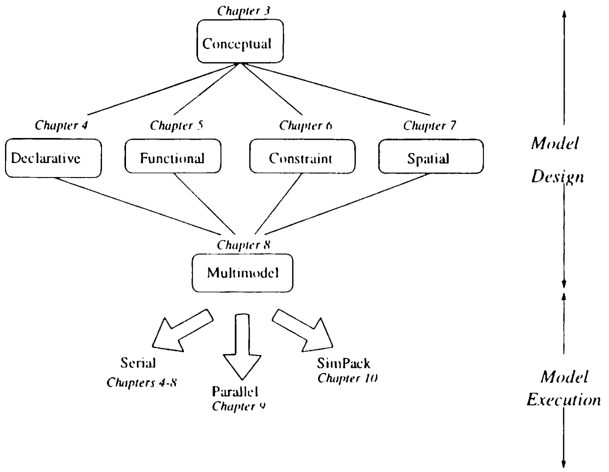

Figure 2: Model engineering progression.

Lorenson 1991) which have more formal treatment. The ultimate conceptual model is one based on database technology, such as an object-oriented database, capturing all facets of the physical system.

2.2 Declarative Models

These models permit dynamics to be encoded as state-to-state or event-to-event transitions. The idea behind declarative modeling is to focus on the structure of state (or event) from one time period to the next, while de-emphasizing functions or constraints which define the transition. Models such as finite state automata (Hopcroft and Ullman 1979), Markov models, event graphs (Schruben 1983) and temporal logic models (Moszkowski 1986) fall into the declarative category. Declarative models are state-based (FSAs), event-based (event graphs) or a hybrid (Petri nets (Peterson 1981)).

2.3 Functional Models

Functional models represent a directional flow of a signal (discrete or continuous) among transfer functions (boxes). When the system is seen as a set of boxes communicating with messages or signals, the functional paradigm takes hold. The use of functional models is found in control engineering (Ogata 1970; Dorf 1986) (with continuous signals) as well as queuing networks for computer system model design (MacDougall 1987). Some functional systems focus not so much on the functions, but more on the variables. Such models include signal flow graphs, compartmental models (Jacquez 1985), and Systems Dynamics (Roberts, Andersen, Deal, Garet, and Shaffer 1983).

2.4 Constraint Models

There are two types of constraint models: *equational* and *graph-based*. Constraint models are models where a balance (or constraint) is at the heart of the model design. In such a case, an equation is often the best characterization of the model since a directional approach such as functional modeling is insufficient. Equational systems include difference models, ODEs and delay differential equations. Graphical models such as bond graphs (Breedveld 1986; Karnopp, Margolis, and Rosenberg 1990) and electrical network graphs (Raghuram 1989) are also constraint based.

2.5 Spatial Models

If a system is spatially decomposed as for cellular automata (Wolfram 1986; Toffoli and Margolus 1987), Ising systems, PDE-based solutions or finite element models, then the system is being modeled using a spatial modeling technique. Spatial models are used to model systems in great detail, where individual pieces of physical phenomena are modeled by discretizing the geometry of the system. Spatial models are "entity-based" or "space-based." Entity-based spatial models focus on a fixed space where the entity dynamics are given whereas space-based focus on how the space changes by convolving a template over the space at each time step. PDEs are space-based where the template defines the integration method. L-Systems (Prusinkiewicz and Lindenmeyer 1990) are entity-based since the dynamics are based on how the organism grows over a fixed space.

2.6 Multimodels

Large scale models are built from one or more abstraction levels, each level being designed using one of the aforementioned more primitive model types. The lowest level of abstraction for a system will probably use a spatial model whereas the highest level may use a declarative finite state machine. Intermediate levels will often use functional and constraint techniques. Models which are composed of other models are termed *multimodels* (Fishwick and Zeigler 1992; Fishwick 1992; Fishwick 1993). By utilizing abstraction levels, we can switch levels during the simulation and use the abstraction most appropriate at that given time. This approach gives us multiple levels of explanation and is computationally more efficient than simulating the system at one level.

REFERENCES

Booch, G. 1991. *Object Oriented Design*. Benjamin Cummings.

Breedveld, P. C. 1986. A Systematic Method to Derive Bond Graph Models. In *Second European Simulation Congress*, Antwerp, Belgium.

Dorf, R. C. 1986. *Modern Control Systems.* Addison Wesley.

Fishwick, P. A. 1992. An Integrated Approach to System Modelling using a Synthesis of Artificial Intelligence, Software Engineering and Simulation Methodologies. *ACM Transactions on Modeling and Computer Simulation.* (submitted for review).

Fishwick, P. A. 1993. A simulation environment for multimodeling. *Discrete Event Dynamic Systems: Theory and Applications 3*, 151–171.

Fishwick, P. A. 1995. *Simulation Model Design and Execution: Building Digital Worlds.* Prentice Hall.

Fishwick, P. A. and B. P. Zeigler. 1992. A Multimodel Methodology for Qualitative Model Engineering. *ACM Transactions on Modeling and Computer Simulation 2*(1), 52–81.

Hopcroft, J. E. and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages and Computation.* Addison Wesley.

Jacquez, J. A. 1985. *Compartmental Analysis in Biology and Medicine* (2nd ed.). University of Michigan Press.

Karnopp, D. C., D. L. Margolis, and R. C. Rosenberg. 1990. *System Dynamics.* John Wiley and Sons.

MacDougall, M. H. 1987. *Simulating Computer Systems: Techniques and Tools.* MIT Press.

Moszkowski, B. 1986. *Executing Temporal Logic Programs.* Cambridge: Cambridge Press.

Ogata, K. 1970. *Modern Control Engineering.* Prentice Hall.

Peterson, J. L. 1981. *Petri Net Theory and the Modeling of Systems.* Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Prusinkiewicz, P. and A. Lindenmeyer. 1990. *The Algorithmic Beauty of Plants.* Springer-Verlag.

Raghuram, R. 1989. *Computer Simulation of Electronic Circuits.* John Wiley.

Roberts, N., D. Andersen, R. Deal, M. Garet, and W. Shaffer. 1983. *Introduction to Computer Simulation: A Systems Dynamics Approach.* Addison-Wesley.

Rumbaugh, J., M. Blaha, W. Premerlani, E. Frederick, and W. Lorenson. 1991. *Object-Oriented Modeling and Design.* Prentice Hall.

Schruben, L. W. 1983. Simulation Modeling with Event Graphs. *Communications of the ACM 26*(11).

Toffoli, T. and N. Margolus. 1987. *Cellular Automata Machines: A New Environment for Modeling* (2 ed.). MIT Press.

Wolfram, S. 1986. *Theory and Applications of Cellular Automata.* Singapore: World Scientific Publishing. (includes selected papers from 1983 - 1986).

Woods, W. A. 1975. What's in a Link: Foundations for Semantic Networks. In D. Bobrow and A. Collins (Eds.), *Representation and Understanding.* Academic Press.

AUTHOR BIOGRAPHY

PAUL A. FISHWICK is an Associate Professor in the Department of Computer and Information Science and Engineering at the University of Florida. He received the BS in Mathematics from the Pennsylvania State University, MS in Applied Science from the College of William and Mary, and PhD in Computer and Information Science from the University of Pennsylvania in 1986. He also has six years of industrial/government production and research experience working at Newport News Shipbuilding and Dry Dock Co. (doing CAD/CAM parts definition research) and at NASA Langley Research Center (studying engineering data base models for structural engineering). His research interests are in computer simulation modeling and analysis methods for complex systems. Dr. Fishwick created the `comp.simulation` newsgroup (Simulation Digest) in 1987 and moderated the group until December 1993. He is a senior member of the IEEE and the Society for Computer Simulation. Dr. Fishwick was chairman of the IEEE Computer Society technical committee on simulation (TCSIM) for two years (1988-1990) and he is on the editorial boards of several journals including the *ACM Transactions on Modeling and Computer Simulation, IEEE Transactions on Systems, Man and Cybernetics, The Transactions of the Society for Computer Simulation, International Journal of Computer Simulation*, and the *Journal of Systems Engineering.*