*Proceedings of the 1993 Winter Simulation Conference*
*G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)*

# MULTIMODELING AS A UNIFIED MODELING FRAMEWORK

Paul A. Fishwick

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

## ABSTRACT

In this tutorial talk, we will present the technique of *multimodeling* which permits one to create large, structured models with many levels of abstraction — from high levels of granularity down to continuous state models. Multimodeling, in a nutshell, endows the simulationist with the capability to blend different model types together to form hybrid models. To model a milk sterilization facility, for instance, might involve a finite state machine or Petri net at the high level to model phase transitions associated with transporting the milk, heating the milk and packaging. Lower level functional block models refine the FSA or Petri net states to define relationships in a lower level phase space with variables representing temperature, pressure, volume and position. Multimodeling offers a unified view toward modeling with well known model types instead of promoting a singular model type which is insufficient to model large scale systems at many levels (Fishwick 1994).

## TUTORIAL OVERVIEW

This overview is adapted from a recent article (Fishwick 1993) on multimodel implementation in Simpack (Fishwick 1992b). Modeling "in the large" has always been a central topic in computer simulation. Approaches in past work have concentrated on using functional coupling and hierarchical decomposition to alleviate the problems associated with complex, cumbersome models. These techniques have been partially successful but they have lead to the manifestation of simulation languages and methodologies that grow to fit the needs of new applications and extensions. Our approach —multimodeling— advocates the use of existing, well known model types (such as Petri networks, automata, Markov models and block models) within the same model structure: a multimodel. Multimodeling (Fishwick 1991; Fishwick and Zeigler 1992; Fishwick 1992a) is the process of engineering a model by combining different model types to form an abstraction network or hierarchy. If we begin to understand a physical system by creating a model, we often find that the model is too limited; the model will answer only a very limited set of questions about system behavior. It is necessary, then, to create many models and link them together, thereby maintaining a multi-level view of a system while permitting an analyst to observe system output at several abstraction levels. Even apparently simple physical systems can exhibit remarkly complex behavior in terms of all models necessary to completely define the system. A significant problem with single level models is that, not only do they attempt to answer a narrow set of questions, but the models are made to be used only by a select number of people. It would be convenient if we could create models that can provide answers to children in an educational setting, as well as to scientists. To enlarge our class of answerable questions, we must combine models together in some seamless fashion.

The previous research in combined modeling has fueled the study of multimodels, but our approach to modeling is significantly different in that we build models for large-scale systems by employing those modeling techniques that have been proven useful for a specific abstraction level. Instead of using only Petri nets or only automata, and then extending these types (i.e., colored Petri nets, augmented automata), we mix and match types to form a multimodel. We do not subscribe to the idea that, in order to model ever increasingly complicated systems, one must continue to extend a singular modeling approach. Instead, a blending of approaches is warranted. The multimodel approach uses SimPack (Fishwick 1992b) tools depending on the model types present in the abstraction hierarchy. *SimPack* is a collection of C and C++ libraries and executable programs for computer simulation. In this collection, several different simulation algorithms are supported including discrete event simulation, continuous simulation and multimodel simulation. The purpose of the *SimPack* toolkit is to provide the user with a set of utilities that illustrate the basics of building a working simulation from a model description. Most simulation packages cover one of two areas: discrete event or continuous. Discrete event methods cater to those performing modeling of queuing networks, flexible manufacturing systems and inventory practices. Continuous methods are normally associated with block diagram methods for control and system engineering. Some available software can perform both types of simulation; however, bulk support is usually available in only one form. As systems become large and complex, the analyst will require simulation software that can support a wide variety of model types. We provide a set of C and C++ tools that accommodates a direct translation from graph-based modeling approaches into callable routines; we do not force the user to think in terms of a single overall language for all simulation applications. Instead, we believe that most systems will contain model components whose types are different. The perceived need to have an "all in one" simulation language does not match most real world problems where a set of well-tested model types has developed naturally. The *SimPack* emphasis on "diversity" in modeling makes it a natural candidate in which to construct multimodels.

## REFERENCES

Fishwick, P. A. 1991. Heterogeneous Decomposition and Coupling for Combined Modeling. In *1991 Winter Simulation Conference*, pages 1199 - 1208, Phoenix, AZ.

Fishwick, P. A. 1992a. An Integrated Approach to System Modelling using a Synthesis of Artificial Intelligence, Software Engineering and Simulation Methodologies. *ACM Transactions on Modeling and Computer Simulation*, 2(4):307–330.

Fishwick, P. A. 1992b. Simpack: Getting Started with Simulation Programming in C and C++. In *1992 Winter Simulation Conference*, pages 154–162, Arlington, VA.

Fishwick, P. A. 1993. A Simulation Environment for Multimodeling. *Discrete Event Dynamic Systems: Theory and Applications*, 3:151–171.

Fishwick, P. A. 1994. *Simulation Model Design and Execution*. Prentice Hall. To be published as a textbook.

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

## AUTHOR BIOGRAPHY

**PAUL A. FISHWICK** is an associate professor in the Department of Computer and Information Sciences 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. He is a senior member of the IEEE and the Society for Computer Simulation. He is also a member of the IEEE Society for Systems, Man and Cybernetics, ACM and AAAI. 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*.