Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

WEB-BASED SIMULATION

Paul A. Fishwick

Department of Computer & Information Science & Engineering
University of Florida
Gainesville, Florida 32611, U.S.A.

ABSTRACT

The state of the art talk which accompanies this entry in the WSC proceedings presents a working definition for "web-based simulation" as well as a methodology for constructing model repositories, connected to the web. The methodology, Object-Oriented Physical Modeling (OOPM), defines a formal approach to constructing both natural and artificial systems using an extension of the classical object-oriented framework as defined in software engineering and programming language design. The extension is one where class attributes and methods can take on *models* as values. The end result of OOPM design is a model repository, which is integrated with the web and made available to others on the Internet so that models can be constructed in a "plug and play" fashion.

1 INTRODUCTION

The World Wide Web (often just referred to as "the web") represents a fertile area in which to perform computer simulation research. It only makes common sense to imagine that by combining the web with computer simulation, that the simulation community can achieve an integration that will have an key impact on future simulation research. There are several directions that one can take in creating a marriage between the web and the simulation field (Fishwick 1996). Two directions involve 1) parallel and distributed model execution, and 2) distributed model repositories. Both of these avenues are fruitful. We have narrowed our focus to the area of distributed model repositories since there has been less research in this area than in the more mature field of distributed simulation (Fujimoto 1990; Lin and Fishwick 1996). Also, the concept of model repository lends itself to the study of how to organize model information. Since the web is also concerned with how to effectively organize information, this appears to be a reasonable way to blend the web with simulation.

The web defines a networked hypermedia approach to storing information. Search engines exist to help a user browse or perform a topical search. In simulation, information is generally focused on physical objects. These physical objects, whether they are humans, milling machines or a container of fluid, have attributes and exhibit behaviors. If we are to permit a situation where physical object information is as freely available as hypermedia to remote users on today's web, then we need to 1) formalize this information, 2) provide a way to integrate to today's web-based information, and 3) effect mechanisms for searching and browsing models. In the talk, we address these three issues as follows:

1. *Formalizing Model Information:* Object-Oriented Physical Modeling (OOPM) (Fishwick 1997) defines a formal approach to capturing physical knowledge in a form that extends the object design principles specified in the fast-growing area of object design within software engineering and programming language design (Booch 1991; Rumbaugh et al. 1991). Some of the current object-oriented design methodology requires modification to support physical modeling. Moreover, there does not currently exist a clearly-defined method of capturing physical knowledge in an object-oriented modeling framework even though many of the object-oriented "nuts and bolts" exist to help structure the method. The OOPM methodology satisfies the requirement of development of a theoretical framework for physical modeling, while allowing for legacy code insertion and user-defined dynamic model and multimodel types (Fishwick 1995).

2. *Integrating with Today's Web:* Since the web is a network of multimedia documents, we must provide a way of integrating OOPM with the web. This is done by a simple mechanism: permitting class and object attributes to be of a

type *Web*. *Web* defines a class, where object instances reflect HTTP addresses. In this fashion, documentation is seen as an attribute of an object. Within a web document, a conceptual model may be inserted as a basic URL type: `model`. To retrieve a conceptual model of a six-cylinder automobile engine from Detroit, the following URL would be accessed: `model://models.general-motors.com/engine6cyl.mod`. This permits a tightly-coupled, interwoven effect between the existing web and a conceptual model containing classes and relations among classes.

3. *Model Browsing and Searching:* Models need to be stored on the Internet in an easily accessible fashion. A little over a year go, we began construction on the Moose Model Repository (MMR) to achieve this storage requirement. The repository comes with two methods of access. The first method is a protocol which allows one to retrieve and store model information in connection with the repository. The second method is an independent definition file (IDF) of a model. The IDF file is a flat ASCII file that serves to capture all information in the MMR. The IDF file can be used for both import and export purposes, allowing simple transfer and hand-editing of models where necessary. In a nutshell, the MMR represents a conceptual model. The conceptual model defines a set of classes and relations among classes, as well as objects created from these classes. Each object has attributes and methods, as do all objects. Attributes are either variables or static models, and methods are either code methods or dynamic model methods (Fishwick 1997).

MOOSE

Based on the OOPM methodology, we have constructed an ongoing prototype implementation called MOOSE (Multimodeling Object-Oriented Simulation Environment) (Cubert and Fishwick 1997). MOOSE represents a computer implementation of OOPM, while including a graphical user interface for conceptual and dynamic model creation and editing. The architecture for the model repository in MOOSE is illustrated in figure 1. Figure 1 includes the following terms:

- MOOSE Model Repository (MMR): this is defined by the MMR protocol needed for searching, setting and retrieving conceptual models and their components. The schema for the MMR is not currently defined so that it is free to be implemented as a flat file, series of files or a database.

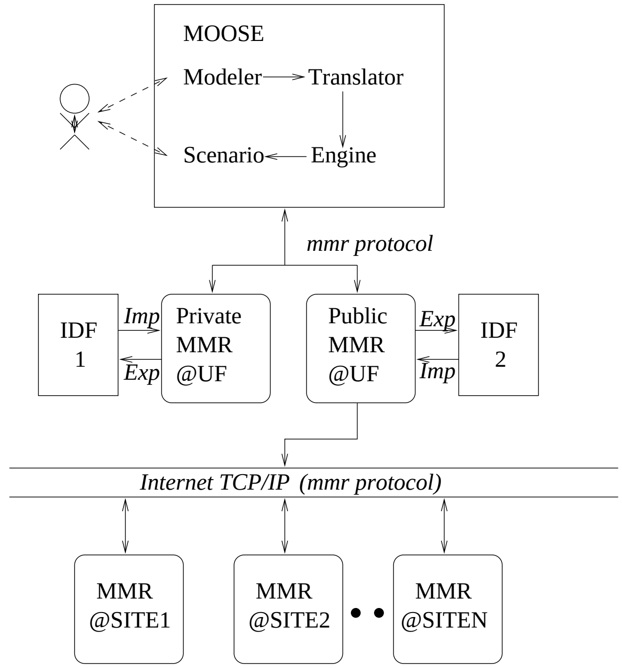

Figure 1: Model Repository Architecture

- MMR Protocol: this defines the contract between a client program and the MMR repository, which acts as a server processing requests.
- Modeler: the interactive graphical user interface (GUI) which permits the user to begin with a conceptual model and proceed to fill in attributes and methods for classes. Object creation is also facilitated.
- Translator: the process of translating the model definition to an arbitrary implementation language, such as C++.
- Engine: the simulator that takes the output from the translator and attaches an event-driven code that simulates the conceptual model with all of its subordinate classes, objects and their dynamic model methods and code methods.
- Scenario: the module that presents simulation output to the user.
- MMR@SITE: an MMR can be private or made publicly available for external site access. "UF" refers to the University of Florida, where an initial MMR is under construction.

- IDF: Independent Definition File, which can be imported or exported. This allows for an ASCII file representation of an MMR.

2 FUTURE WORK

Work on MOOSE and the model repositories is ongoing and represents a large time investment. Our purpose has been to allow users a way to create multimodels in a graphical manner, thereby exploiting model types with which they are already comfortable. There is no need to drop the current approach to modeling and conform to a singular model type. The model repository will lead to a prototype that allows modelers to treat models as interchangeable units. Just as the web search engines currently index text and multimedia objects, we envision that search engines can be built to index models by their functions, class type or some other characteristic.

ACKNOWLEDGMENTS

I would like to acknowledge the graduate students of the MOOSE team for their individual efforts in making MOOSE a reality: Robert Cubert, Tolga Goktekin, Gyooseok Kim, Youngsup Kim, Kangsun Lee and Jin Joo Lee. We would like to thank the following funding sources that have contributed towards our study of modeling and implementation of a multimodeling simulation environment for analysis and planning: (1) Rome Laboratory, Griffiss Air Force Base, New York under contract F30602-95-C-0267 and grant F30602-95-1-0031; (2) Department of the Interior under grant 14-45-0009-1544-154 and the (3) National Science Foundation Engineering Research Center (ERC) in Particle Science and Technology at the University of Florida (with Industrial Partners of the ERC) under grant EEC-94-02989.

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

Cubert, R. M. and P. A. Fishwick. 1997. Moose: An object-oriented multimodeling and simulation application framework. *Simulation*.

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

Fishwick, P. A. 1996. Web-based simulation: Some personal observations. In *1996 Winter Simulation Conference*, pp. 772–779. Society for Computer Simulation International.

Fishwick, P. A. 1997. A visual object-oriented multimodeling design approach for physical modeling. *ACM Transactions on Modeling and Computer Simulation*. Revised article submitted for review.

Fujimoto, R. M. 1990, October). Parallel Discrete Event Simulation. *Communications of the ACM 33*(10), 31–53.

Lin, Y.-B. and P. A. Fishwick. 1996. Asynchronous Parallel Discrete Event Simulation. *IEEE Transactions on Systems, Man and Cybernetics 26*(4), 397–412.

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

AUTHOR BIOGRAPHY

PAUL A. FISHWICK is an Associate Professor in the Department of Computer and Information Science and Engineering at the University of Florida, and will be serving as General Chair of the 2000 Winter Simulation Conference. 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 founded the `comp.simulation` Internet news group (Simulation Digest) in 1987, which now serves over 15,000 subscribers. He has chaired workshops and conferences in the area of computer simulation, and will serve as General Chair of the 2000 Winter Simulation Conference. He 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*. Dr. Fishwick's WWW home page is `http://www.cise.ufl.edu/~fishwick` and his E-mail address is `fishwick@cise.ufl.edu`.