Guest Editorial:

Object-Oriented Simulation: Aspects of the Model Development Lifecycle

John A. Miller
University of Georgia

1 INTRODUCTION

Recently, object-oriented programming has been increasingly applied to simulation, using languages such as C++ and ModSim II. The benefits of utilizing this programming paradigm are being realized in the simulation area as they have been in the software industry in general. It is, however, somewhat misleading to imply that simulation is following this general trend in software development. In the mid 1960’s a pioneering simulation languages called Simula-67 was developed. Motivated by the need to more fully model entities or objects in the real-world, the designers of Simula-67 introduced the core concepts that now form the object-oriented paradigm. Simula-67 also strongly influenced the designs of Smalltalk and C++, the two languages which are primarily responsible for the enormous popularity of object-oriented programming today. Therefore, it is only natural for object-oriented programming to be important in simulation, as that is where it originated.

The central theme of this Special Issue is the use of object-oriented concepts throughout the lifecycle of software/model development. In software engineering, object-oriented approaches to requirements analysis (OOA), design (OOD) and programming (OOP) are meeting with success. Addressing software engineering issues, providing environments, and utilizing CASE tools for the application area of simulation are important concerns if the potential of the object-oriented paradigm is to be fully realized in simulation. The papers in this Special Issue make distinct contributions to this understanding. The first group of papers uses graphical design specifications to automate the development/execution of simulation models. The next group of papers discusses mechanisms for maximizing resuability. The last group is concerned with the backend of the lifecycle, namely software/model testing.

2 MODEL EXECUTION FROM DATA FLOW DESIGNS

The first pair of papers addresses the difficult problem of executing design specifications. Both papers use data flow concepts from software engineering augmented with Petri net characteristics. The paper entitled, "An Object-Oriented Simulation Procedure for Automating Transformation Schema Execution" is by Christopher J. Coomber from Deakin University in Australia.

This paper shows how a transformation schema (an enhanced data flow diagram) can be rendered in an automatable form by employing concepts from object-oriented design, and how the resulting system of objects can be executed using an object-oriented simulation strategy.

The paper entitled "The Object Flow Model for Discrete Event Simulation" is by Lissa F. Pollacia from Northwestern State University, and Lois M.L. Delcambre from the Oregon Graduate Institute. This paper has more of a database orientation. Using their approach, objects can be retrieved from a database, flow through an object flow diagram triggering events, and be stored back in the database.

3 MODEL REUSABILITY AND BEHAVIOR SHARING

The second pair of papers presents alternative means of providing resuability and behavior sharing. They discuss how design techniques and language features can be used to maximize opportunities for sharing and resue. The paper entitled "The DEVS Formalism: Reusable Model Specification in an Object-Oriented Framework" is by Tag Gon Kim from the Korea Advanced Institute of Science and Technology. Kim uses the mechanism of inheritance under strict hierarchies coupled with the discipline imposed by the DEVS formalism to achieve high reusability.

The paper entitled "Actor Tools for Discrete Event Simulation" is by Alain Senteni from the Universite de la Reunion, Sylvain Giroux and Guy Lapalme from the University of Montreal, and Patrick Salle from the Universite Paul-Sabatier. These authors use the more flexible mechanism of delegation to achieve the same ends.

4 SIMULATION TESTBEDS

Using object-oriented technology allows powerful and flexible testing environments to be developed. New modules can more easily be incorporated into a system and tested. The final paper in this Special Issue is entitled "An Object-Oriented Simulation Testbed for Exercising Knowledge-Based Systems", and is by Michael L. Hilton from the University of South Carolina and Craig S. Anken from the US Air Force Systems Command. It discusses how these aims are achieved in AAITT, a testbed developed by the military for command and control applications.

5 SUMMARY

The papers in this Special Issue have examined important issues in the model development lifecycle. These five papers were selected from eleven submissions, and have benefited from the improvements suggested by the referees. I would like to thank the authors for their contributions, and especially to thank the referees (listed below) for performing their vital but relatively thankless service. Finally, special thanks go to the Editor-in-Chief of the International Journal in Computer Simulation, George W. Zobrist, for making this Special Issue possible.

6 REFEREES

Marc Abrams	Virginia Tech
Osman Balci	Virginia Tech
Martha A. Centeno	Florida State University
Jia-Bing Cheng	Pennsylvania State University
Deborah A. Davis	Systems Modeling Corp.
John W. Fendrich	Bradley University
Paul A. Fishwick	University of Florida
Martin D. Fraser	Georgia State University
Jorge Haddock	Rensselaer Polytechnic Institute
Manjunath Kamath	Oklahoma State University
Krys J. Kochut	University of Georgia
Enrique V. Kortright	Nicholls State University
Dennis S. Mok	Bellcore
Walter D. Potter	University of Georgia
Herb Schwetman	MCC
Andrew F. Seila	University of Georgia
Jeffrey W. Smith	University of Georgia
Philip A. Wilsey	University of Cincinnati