Preface

Reusable simulation software! Now that's the edge needed by the simulation technologist in today's competitive environment. At present, large research organizations and industrial corporations are global in structure. As a result, units or modules of simulation code developed in the United States must be *reusable* in other parts of the world to realize this competitive cost edge. And, in the development of large systems, building a "breadboard" or engineering model is expensive, especially if it doesn't work as specified. When we simulate a large system (or any system, for that matter) on computers (or parallel processors), the system can be validated before we build it. Better yet, the simulated system can be "perturbed" to see how it reacts to real-world disturbances or out-of-tolerance conditions (lightning strikes, power outages, loss of signal, or lack of critical parts on the production line, for example). In order to perform large-scale simulations, our simulation techniques must be *adaptable* to real world situations. However, some philosophical analyses should be planned prior to application because, once a simulation program is in place, it must produce repeatable results. As we all know, "the only thing in this world that changes is change itself." So our simulation software must also be *maintainable*.

Reusability, adaptability, maintainability—that's the ticket in today's global economy. These standard software object-oriented design procedures can be applied to the development of simulation environments for separation of physical, control, and information elements of a system. *Object-Oriented Simulation: Reusability, Adaptability, Maintainability* is a valuable reference resource for the simulation developer who desires to work on the *leading edge* of simulation technology. The book was written

for those professionally involved in computer simulation research as well as for designers, developers, producers, scholars, and managers. It will also benefit those preparing proposals for simulation techniques and those involved in research projects. It is particularly well-suited as an ancillary text for courses in object-oriented simulation and simulation methodology. The book contains specially-edited chapters on object-oriented simulation. Although such papers are presented in journals, special issues, and at conferences, it is difficult to find a single source that provides the foremost papers on the subject. In addition to presenting well-known software engineering techniques that are required in the design of a simulation environment, the information is presented in an object-oriented environment. Illustrated examples display the latest advances in object-oriented simulation techniques, which can help developers save countless research and development hours and increase productivity levels. The reader will gain a solid understanding of the concept of Object-Oriented Simulation (OOS), various applications utilizing the OOS environment, multilevel modeling and reusability in an OOS environment, object-oriented simulation testbeds, object-oriented concurrent programming for discrete event or behavioral simulation, and the use of object-oriented concepts throughout the lifecycle of software/model development. A particularly valuable feature is the variety of new simulation techniques described in the final three chapters.

An introductory survey is presented in Chapter 1 to offer the reader a good understanding of the basic concepts of OOS. Chapter 2 presents simulation languages based on C++. In Chapter 3, the Object Flow Model (OFM) and its contribution to simulation and conceptual modeling of database applications is examined. Discrete-event simulation is covered in Chapter 4, with a description of a methodology for developing reusable simulation models in an object-oriented framework. Chapter 5 discusses the issues involved in attempting to provide a multilevel simulation modeling capacity within object-oriented frameworks. In Chapter 6, a detailed account of a particular software integration environment, the Advanced AI Technology Testbed (AAITT) is reported. Chapter 7 reports further research on automating transformation schema execution. Then Chapter 8, building on the background gained from the previous seven chapters, presents a progression of steps for the application of object-oriented techniques to discrete-event or behavioral simulation problems, starting from simple objects (or actors) up to whole ecologies of reflective agents.