The Java Simulation Handbook –

Simulating Discrete Event Systems with UML and Java

Bernd Page, Wolfgang Kreutzer
(main authors)

Coauthored by Björn Gehlsen, Johannes Göbel,
Gunnar Kiesel, Nicolas Knaak, Julia Kuck, Tim Lechler
Ruth Meyer, Gaby Neumann, Volker Wohlgemuth

Berichte aus der Informatik

Bernd Page,
Wolfgang Kreutzer

The Java Simulation Handbook

Simulating Discrete Event Systems
with UML and Java

Shaker Verlag
Aachen 2005

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the internet at http://dnb.ddb.de.

Printed in Germany.

ISBN 3-8322-3771-2
ISSN 0945-0807

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • eMail: info@shaker.de

Introduction

Computer simulation is an important tool for modelling and analysing a complex system. Its applications range widely; from the natural to the engineering sciences, from the social sciences to economics, from medicine to studying environmental concerns. Within computer science, simulation has been used in such diverse areas as chip design, system and communication network performance analysis, database design, and for the study of operating systems and other software. When applied to commercial problems, computer simulation offers a flexible and highly successful method to explore and optimize information and material flows in an enterprise.

Designing and implementing suitable algorithms and programs for such a wide range of applications, however, remains a challenging task; particularly since it seems highly desirable to base models on unified and transferable architectures, and since usable and powerful modelling software can be a key ingredient to the success of a simulation study.

Simulation is one of the earliest applications of computing technology. In addition to mastery of the respective domains (e.g. production, computer, telecommunication, or logistics systems), its application demands competence in a large number of areas, such as system analysis, system design, statistical methods, experimental design, software design and programming. Because of its wide ranging applications, its high degree of practical relevance, and its demand for a unique blend of skills in both system design and implementation, simulation plays also an important and easily motivated part in teaching academic computer science programmes. Good software is needed to support exploration of the whole spectrum of simulation development, from system analysis to model implementation and experimentation. Programming a simulation model, in particular, places high cognitive demands on the model designer. By offering prepackaged functionality to support specific modelling contexts, object-oriented frameworks help to master this complexity. Good modelling frameworks ensure that many *programming aspects* of a simulation study are already taken care of, so that a model designer's mind becomes free to concentrate on *problem-specific aspects* of a system under investigation.

For many years the Faculty of Informatics at the University of Hamburg has been involved in developing software for discrete event simulation frameworks, hosted in different programming languages. These frameworks package core functionality for implementing discrete event simulations and therefore leave students more time to analyse, design, experiment with, and evaluate relevant system models. Studying their architecture also aids understanding of important structural aspects and control patterns in simulation software.

Prompted by Java's increasing popularity as a programming tool for both teaching at universities and use in industrial practice, we have, for a number of years, designed and

implemented a Java-based discrete event simulation framework called DESMO-J (Discrete Event Simulation and MOdelling in Java). We plan to continue maintaining and improving DESMO-J as public domain software under the GNU Lesser General Public License. From 2000 to 2004 a close cooperation with the University of Canterbury has been supported by the Federal Ministry of Education and Research (BMBF)[1].

In drawing on the results of this cooperation, the Java Simulation Handbook presents a broad palette of important and timely topics within the context of discrete event simulation. All contributions were written by competent specialists in the respective areas from different universities, coordinated by the two main authors, who have provided the content framework for the book. The book's 16 chapters build on each other and present information in a common style. Although chapters often refer to each other's examples and discussions, each still remains reasonably self-contained. The fact that each chapter can stand on its own means that they may be read both sequentially, in the specified order, or more selectively, whenever particular information is needed. This makes the Java Simulation Handbook suitable to serve as either a textbook (e.g. for a university course) or as a reference.

The book presents its contents in four parts. The first, *Foundations*, discusses fundamental concepts of discrete event simulation. The next part, *Software*, concerns the implementation of discrete event simulation in general as well as in DESMO-J, while *Advanced Methodology* addresses current research, such as agent-based and distributed simulation. The *Applications* part then summarizes simulation's contributions to e-learning, logistics, and industrial practice. For the practitioner, it finally offers some guidelines for the successful completion of real-world simulation projects.

Systematic design and development of discrete event models is the dominant focus of the book. To support our presentation, we use the well established UML 2 (Unified Modelling Language, Version 2.0) notation for model design and the above mentioned DESMO-J Java-based simulation framework for model implementation. To give better support to the simulation domain, we also enrich UML 2 with some simulation-specific extensions.

Although this focus on model design and implementation distinguishes this handbook from many others, which favour a statistical perspective on simulation technology, the handbook also includes chapters dealing with relevant statistical aspects and other foundational issues.

It is one of this book's aspirations to teach by example. To further this purpose, it contains many examples of models, programs, and exercises, the completion of which will hopefully inspire the reader to explore further. This aspect and its associated web-based resources make the book particularly well-suited for teaching a course which emphasises the programming aspects of discrete event simulation.

[1]BMBF Programme for Scientific-Technological Collaboration with New Zealand, project number NZL 00/02

The Java Simulation Handbook is supported by a number of web-based resources. Readers can access a web server at `http://www.desmoj.de` with the DESMO-J software, a web-based DESMO-J tutorial, and a link to the cooperation platform CommSy (with a CommSy project room *JavaSimulationHandbook*). It contains a simulation laboratory with many example programs and some applet-based animations. These materials were developed as part of an e-learning project, which was financed by the E-Learning Consortium Hamburg from 2003 until 2005. The handbook itself is one of the results of this project and would not exist without this support. The web-based resources are a useful add-on to the book. In addition to its role as a repository for teaching materials, the project's online cooperation platform also offers a means for discussing the book's contents. User access to the CommSy-platform is open to readers of this book using the guest account *JavaBeans* and the password *snoopy*. The same account and password allow for access to the DESMO-J laboratory on the Internet.

The intended readership of this handbook and the DESMO-J framework are students and teachers of computer science, at both academic and other, possibly more applied, tertiary educational institutions. Readers should feel confident in using object-oriented programming styles and Java. Moreover, they should be interested in acquiring core competence in system analysis, model design, and the implementation of discrete event simulation programs. Some elementary knowledge of probability theory and statistics is assumed. Since the book only uses a small range of techniques, this can be obtained rather quickly. Suitable recommendations for further reading are given under *Further Reading* at the end of those chapters where additional background information seemed to be helpful for certain readers. Some knowledge of UML may prove helpful. To provide some background, the handbook presents those parts of UML that are particularly relevant to simulation.

The authors would like to thank all contributors, without whose participation a handbook of this depth and size would not have been possible. The DESMO-J software has been developed and extended by many colleagues and students, among who the main developers of the DESMO-J core Java foundation, Tim Lechler and Sönke Claasen, deserve particular thanks. Editorial work on the manuscript and its markup in LaTeXwas done by Johannes Göbel, with assistance by Ruth Meyer. Ruth Meyer also wrote the current version of the DESMO-J tutorial. Other e-learning materials for this book were created by Alexander Bentz and Gunnar Kiesel. At the end we must thank Kirsten-Verena Bull, Frank Heitmann, and Alexander Bentz for their final corrections, which have prevented many errors and inconsistencies to slip through.

We are very grateful to the BMBF and its New Zealand counterpart, for their financial support over many years. Among other things this has enabled us to travel in order to cooperate and communicate effectively over the large physical distance between Germany and New Zealand. Ultimately, this has provided a sound scientific basis on which the Java Simulation Handbook could be built. Further thanks go to the E-Learning Consortium Hamburg, for its financial support in writing the handbook and, in particular, for assistance in developing the accompanying electronic materials.

Finally, we thank the Shaker Verlag for the professional and timely publication of this handbook in its Informatik series. We are particularly pleased with the book's electronic version, which reflects our vision of a *blended* learning environment for the teaching of simulation rather well.

Bernd Page,
University of Hamburg

Wolfgang Kreutzer,
University of Canterbury

Hamburg and Christchurch, September 2005

Contents