Simulation Foundations, Methods and Applications

Okan Topçu
Umut Durak
Halit Oğuztüzün
Levent Yilmaz

Distributed Simulation

A Model Driven Engineering Approach

Springer

Simulation Foundations, Methods and Applications

Series Editor:
Louis G. Birta, University of Ottawa, Canada

Advisory Board:
Roy E. Crosbie, California State University, Chico, USA
Tony Jakeman, Australian National University, Australia
Axel Lehmann, Universität der Bundeswehr München, Germany
Stewart Robinson, Loughborough University, UK
Andreas Tolk, Old Dominion University, USA
Bernard P. Zeigler, University of Arizona, USA

More information about this series at http://www.springer.com/series/10128

Okan Topçu • Umut Durak • Halit Oğuztüzün
Levent Yilmaz

Distributed Simulation

A Model Driven Engineering Approach

Springer

Okan Topçu
Department of Computer Engineering
Turkish Naval Academy
Istanbul, Turkey

Halit Oğuztüzün
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey

Umut Durak
Department of Flight Dynamics
and Simulation
DLR Institute of Flight Systems
Braunschweig, Germany

Levent Yilmaz
Department of Industrial
and Systems Engineering
Auburn University
Auburn, AL, USA

ISSN 2195-2817 ISSN 2195-2825 (electronic)
Simulation Foundations, Methods and Applications
ISBN 978-3-319-03049-4 ISBN 978-3-319-03050-0 (eBook)
DOI 10.1007/978-3-319-03050-0

Library of Congress Control Number: 2015956543

Springer Cham Heidelberg New York Dordrecht London
© Springer International Publishing Switzerland 2016
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

To Oğuz and Tuğçe with love.
You make it all worthwhile.
Okan Topçu

To my wife, my sailmate, Rabia.
Without you, I could not navigate the high seas.
Umut Durak

To Serpil, Çerağ and Ozan.
My circle of love.
Halit Oğuztüzün

To my mentors and teachers who inspired and motivated me to keep expanding my horizons.
Levent Yilmaz

Foreword

The impressive and imposing stature of contemporary "simulation" is in contrast with its humble beginnings at the fourteenth century. At the beginning, the word "simulation" used to mean "an imitating, feigning, false show, or hypocrisy." Currently, hundreds of types of simulation can be grouped under two categories: performing *experiments* and gaining *experience* by using models of dynamic systems.

Performing experiments – as the essence of the scientific method – was promoted by Francis Bacon in his "Novum Organum" early seventeenth century (1620) in contrast to Aristotelian rhetoric. The well-known advantages of performing experiments on models (i.e., simulation) over performing experiments on real systems are numerous and are augmented by the fact that in several cases, experiments on real systems cannot or should not be done. This last aspect lets simulation to be the only possibility to perform experiments. Performing simulation experiments while real system is running – in tandem or in sequence – has additional advantages that no other technique can provide.

Gaining experience by simulation can enhance three types of skills (i.e., motor, decision making, and operational skills), or can be entertaining. Use of simulation to enhance motor, decision making, and operational skills correspond to virtual, constructive, and live simulations. Gaming simulation to provide experience for entertainment purposes is technically so advanced that some of its techniques are used for serious games.

Advancements of all types of simulation lead to *simulation-based methodologies* and provide infrastructure for a multitude of application areas and disciplines, including *simulation-based engineering* as well as *simulation-based science*. An important type of simulation is distributed simulation to provide several types of training and assessment possibilities for complex systems. In fact, *distributed simulation* has been the de facto approach to tackle several types of complex problems.

At an abstract level, any type of simulation is a model-based knowledge processing activity, and due to vital role of models, the terms simulation and modeling are often used together as *modeling and simulation*. Accordingly, several concepts around models have been developed; they include, in addition to hundreds of types of models, modeling – including static and dynamic model composition – building, maintaining, and searching model bases and relevant activities and concerns, such as model integrity, and many types of model processing – including model transformations and checking. Accordingly, *model-driven approaches* have gained well-deserved and very important momentum in many disciplines, including science and engineering. *Model-driven engineering* is already an important and fundamental concept and practice.

The volume "Distributed Simulation: A Model Driven Engineering Approach" by Topçu, Durak, Oğuztüzün, and Yilmaz is an important contribution for the advancement of modeling and simulation. It is the first book to cover all aspects of distributed simulation from the model-driven engineering perspective. Some of the topics covered in detail include: model-driven engineering, high level architecture, distributed simulation engineering, conceptual modeling, simulation environment design, federate architecture, scenario management, implementation and execution, simulation evolution, and modernization. The advantages of the synergies of model-driven engineering and agents, which provide a very powerful computational paradigm, are also covered for the future advancements. Three case studies are given to provide ample clarification of the concepts presented. The volume provides practitioners a powerful way to tackle complex problems.

University of Ottawa Tuncer Ören
Ottawa, ON, Canada
September 10, 2015

Preface

Purpose of the Book

Both distributed simulation (DS), an area of modeling and simulation (M&S), and model-driven engineering (MDE), an area of software engineering, are fields with their distinct body of knowledge. This book builds on the premise that developments in one field can provide new avenues for advancing the theory, methodology, and the practice in the other field. Therefore, it provides a comprehensive view on DS from the MDE perspective by illustrating how MDE affects the overall lifecycle of the simulation development process.

Rationale

Software engineering aims to improve the quality and efficacy of software intensive systems engineering processes. In software engineering, one of the major problems is the gap between the high-level design and the application code deployed on a platform. MDE, as an up-and-coming approach, allows increasing the abstraction level to models to address both the platform and the application complexity. By automating transformations across models at different levels of resolution, an MDE-based process model supports the technology, engineering, and management of software development via tool-based automation and code generation and hence increases both productivity and quality.

DS applications can be regarded as software-intensive systems. The high level architecture (HLA), which is a standard for DS, has been around for over 20 years, and it has already proven its worth in many distributed simulation applications. The synergy between DS, particularly the HLA, and the MDE methodology has been promoted as a viable solution to improve productivity, reuse, and longevity of simulation models. However, both DS and MDE require substantial domain expertise due to their sophisticated and technical body of knowledge. Moreover, the principles and best practices in both fields are dispersed across many publications. In the light of these observations, this book presents a comprehensive reference bringing together the principles, concepts, and development processes of DS from the point of view of the MDE perspective by covering a wide spectrum of topics from conceptual modeling to simulation modernization.

The book supplements the theoretical overview with practical case studies to demonstrate the utility and applicability of the methodology. Multiple case studies are presented to demonstrate the technological aspects of engineering and management of models. Most of the case studies and models presented in this book are developed with public-domain tools that can be downloaded from their project websites.

Book Overview

The book is structured as follows. Chapter 1 presents an introduction to the essential concepts of modeling and simulation, while highlighting DS as the focal area of interest. Chapter 2 provides an overview of the fundamental concepts, principles, and processes of MDE. Chapter 3 gives an introduction to the HLA. These three chapters together lay the technical background for linking two distinct disciplines, DS and MDE. Chapter 4 presents a road map for building and deploying a distributed simulation system in accordance with the MDE perspective, and it introduces a process model based on the principles and best practices of MDE. Chapters 5, 6, 7, 8, 9 and 10 elaborate on the process model presented in this chapter. Chapter 5 treats conceptual modeling from the MDE perspective, and it presents a methodology and a technical framework to develop conceptual models. Chapter 6 introduces the concept of federation (simulation environment) architecture. By demonstrating the formalization of a federation architecture using the metamodeling concepts, we promote automated code generation and semi-automated model validation over the machine processable specifications of a federation architecture. Chapter 7 focuses on federate architectures and presents a practical approach to the design of federations (i.e. simulation member design) by applying a well-known architectural style, the layered architecture. After introducing the model-driven scenario development process, Chapter 8 explains the major concepts of simulation scenarios and discusses the main activities related to scenario management in a distributed simulation. Chapter 9 delineates the nature of MDE-based implementation activities, including model development, model checking, code generation and integration, and testing. Chapter 10 introduces simulation evolution and modernization. It presents and adopts the software modernization approaches, particularly architecture-driven modernization (ADM) for simulation modernization. Finally, Chapter 11 brings the agent paradigm to the fore and examines potential synergies among the agent, DS, and MDE methodologies, pointing to avenues of future research and development at the intersection of these three fields.

Final

We believe that the most prominent contribution of this book is its unique frame of reference in presenting the principles, concepts, and development processes of distributed simulation (DS) from the model-driven engineering (MDE) perspective. As the only book so far that builds on the synergies of DS and MDE, it explains the theoretical underpinnings of DS and MDE and demonstrates the utility and effectiveness of MDE principles in developing DS applications. In this respect, the book covers *de facto* DS standard, namely, High Level Architecture (HLA) (also a *de jure* standard IEEE 1516-2010) in order to illustrate theoretical issues and serve as a testbed to substantiate the role of MDE for DS. We hope that this book gives a direction for the readers, who are interested in adopting MDE principles and practices for developing complex DS systems.

Istanbul, Turkey — Okan Topçu
Braunschweig, Germany — Umut Durak
Ankara, Turkey — Halit Oğuztüzün
Auburn, AL, USA — Levent Yilmaz

Supplementary Material

Software Tools	URL (Web Address)
SimGe	https://sites.google.com/site/okantopcu/simge
RACoN	https://sites.google.com/site/okantopcu/racon
FAMM	https://sites.google.com/site/okantopcu/famm
NSTMSS	https://sites.google.com/site/okantopcu/navysim
DeCoAgent	https://sites.google.com/site/okantopcu/decoagent

Acknowledgments

We would like to thank our colleagues Gürkan Özhan, Mehmet Adak, Deniz Çetinkaya, Vijdan Kızılay, Thorsten Pawletta, Artur Schmidt, Torsten Gerlach, Jürgen Gotschlich, and Robert Siegfried for their contribution in creating some material that found its way into this book. Also we would like to acknowledge Dr. Tuncer Ören for providing an insightful foreword and our associate editor Simon Rees for smooth editorial assistance.

Okan would like to thank his parents, Selime and Bekir, for their boundless support and his graduate students for their worthy feedback.

Umut is grateful for Dr. Holger Duda for his invaluable support. His special thanks go to Simulation Technologies Group in DLR Institute of Flight Systems for the comfort and understanding.

Halit would like to thank all his graduate students; he learned a lot from them.

For various chapters of this book, we have adapted parts of the following articles:

Topçu, O., Adak, M., & Oğuztüzün, H. (2008, July). A metamodel for federation architectures. *Transactions on Modeling and Computer Simulation (TOMACS)*, *18*(3), 10:1–10:29. Parts adapted and reprinted with permission from ACM. Appears in Chap. 6.

Topçu, O., Adak, M., & Oğuztüzün, H. (2009). Metamodeling live sequence charts for code generation. *Software and Systems Modeling (SoSym), 8*(4), 567–583. Parts adapted and reprinted with permission from Springer. Appears in Chap. 6.

Topçu, O., & Oğuztüzün, H. (2010). Scenario management practices in HLA-based distributed simulation. *Journal of Naval Science and Engineering, 6*(2), 1–33. Parts adapted and reprinted with permission from DEBIM. Appears in Chap. 8.

Topçu, O., & Oğuztüzün, H. (2013, March). Layered simulation architecture: A practical approach. *Simulation Modelling Practice and Theory*, *32*, 1–14. Parts adapted and reprinted with permission from Elsevier. Appears in Chap. 7.

Topçu, O. (2004). *Development, representation, and validation of conceptual models in distributed simulation*. Halifax, NS, Canada: Defence R&D Canada – Atlantic (DRDC Atlantic). Parts adapted and reprinted with permission from DRDC-Atlantic. Appears in Chap. 5.

Durak, U., Schmidt, A., & Pawletta, T. (2015). Model-based testing for objective fidelity evaluation of engineering and research flight simulators. In *AIAA modeling and simulation technologies conference*. Dallas, TX. Parts adapted and reprinted with permission from DLR and Hochschule Wismar. Appears in Chap. 9.

Durak, U., Topcu, O., Siegfried, R., & Oguztuzun, H. (2014). Scenario development: A model-driven engineering perspective. In M. S. Obaidat, J. Kacprzyk, & T. Ören (Eds.). *Proceedings of the 4th international conference on simulation and modeling methodologies, technologies and applications*. Vienna, Austria: SCITEPRESS. Parts adapted and reprinted with permission from SCITEPRESS. Appears in Chap. 8.

Gerlach, T., Durak, U., & Gotschlich, J. (2014). Model integration workflow for keeping models up to date in a research simulator. In M. S. Obaidat, J. Kacprzyk, T. Ören (Eds.). *Proceedings of the 4th international conference on simulation and modeling methodologies, technologies and applications*. Vienna, Austria: SCITEPRESS. Parts adapted and reprinted with permission from SCITEPRESS. Appears in Chap. 9.

Schmidt, A., Durak, U., Rasch, C., & Pawletta, T. (2015). Model-based testing approach for MATLAB/Simulink using system entity structure and experimental frames. In *Proceedings of 2015 Spring Simulation Multi-Conference*. Alexandria, DC. Some figures are reprinted with permission from SCS.

Özhan,G., & Oğuztüzün, H. (2015). Transformation of conceptual models to executable high level architecture federation models. In L. Yılmaz (Eds.). *Concepts and methodologies for modeling and simulation – A tribute to Tuncer Ören* (pp. 135–173). Basel, Switzerland: Springer. Some figures and text are reproduced with permission from Springer. Appears in Chap. 5.

Contents

Part III Modeling and Design

List of Symbols/Abbreviations/Acronyms/Initialisms

A

ACM	(Field) Artillery Conceptual Model
ACMM	ACM Metamodel
ADM	Architecture Driven Modernization
ALSP	Aggregate Level Simulation Protocol
AMG	Architecture Management Group
AMMA	ATLAS Model Management Architecture
AOP	Aspect Oriented Programming
API	Application Programming Interface
ARPA	Advanced Research Projects Agency
ATL	ATLAS Transformation Language
ASTM	Abstract Syntax Tree Metamodel

B

BM	Base Model
BMM	Behavioral Metamodel
BOM	Base Object Model
BPMN	Business Process Model and Notation

C

CASE	Computer Assisted Software Engineering
CBD	Causal Block Diagrams
CGF	Computer Generated Force
ChatFdApp	Chat Federate Application
CFF	Call For Fire
CIM	Computation-Independent Model

CM	Conceptual Model
COTS	Commercial off-the-shelf
CSSL	Continuous System Simulation Language

D

DARPA	US Defense Advanced Research Projects Agency
DAVE-ML	Dynamic Aerospace Vehicle Exchange Markup Language
DDM	Data Distribution Management
	Detailed Design Model
DeCoAgent	Deliberative Coherence Driven Agent
DeCoAgentLib	DeCoAgent Library
DeMo	Discrete Event Modeling and Simulation Ontology
DIF	Data Interchange Format
DIN	Deutsches Institut für Normung
DIS	Distributed Interactive Simulation
DLC	Dynamic Link Compatibility
DLL	Dynamic Link Library
DLR	German Aerospace Center
DMAO	DSEEP Multi-Architecture Overlay
DMSO	U.S. Defense Modeling and Simulation Office
DoD	US Department of Defense
DSEEP	Distributed Simulation Engineering and Execution Process
DSL	Domain-Specific Language
DSM	Domain-Specific (Meta)Modeling
DSML	Domain-Specific Modeling Language

E

EF	Experimental Frame
EM	Executable Model
EMF	Eclipse Modeling Framework
EnviFd	Environment Controller Federate
ESA	European Space Agency
ExMFd	Exercise Manager Federate

F

FA	Field Artillery
FAM	Federation Architecture Model
FAME	Federation Architecture Modeling Environment

FAMM	Federation Architecture Metamodel
FCO	First Class Object
FDC	Fire Direction Center
FDD	FOM Document Data
FEAT	Federation Engineering Agreements Template
FED	Federation Execution Details
FEDEP	Federation Development and Execution Process
FedMonFd	Federation Monitor Federate
FEPW	Federation Execution Planner's Workbook
FFE	Fire For Effect
FFL	Federation Foundation Library
FRG	Federation Rapid Generation
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
FMUFd	FMU Federate
FOM	Federation Object Model
FSMM	Federation Structure Metamodel

G

GASTM	Generic Abstract Syntax Meta-Model
GME	Generic Modeling Environment
GPL	General-Purpose Language
GpML	General-purpose Modeling Languages
GReAT	Graph Rewriting and Transformation
GRT	Generic Real-time Target
GUI	Graphical User Interface

H

HDefLib	IEEE1516.2 HLA Defaults Library
HFMM	HLA Federation Metamodel
HIL	Hardware-in-the-Loop
HLA	High Level Architecture
HMOMLib	IEEE 1516.1 Management Object Model Library
HMSC	High Level MSC
HOMM	HLA Object Metamodel
HSMM	HLA Services Metamodel
HTML	HyperText Markup Language

I

IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IF	Interface Specification
IMLib	Methods Library for IEEE 1516.1
ITU	International Telecommunications Union

J

JC3IEDM	(NATO's) Joint C3 Information Exchange Data Model

K

KDM	Knowledge Discovery Metamodel

L

LHS	Left Hand Side
LMM	LSC Metamodel
LSC	Live Sequence Chart

M

M0	Physical Level
M1	Model
M2	Metamodel
M2M	Model-to-Model
M2T	Model-to-Text
M3	Meta-Metamodel
MAAB	MathWorks Automotive Advisory Board
MATE	Model Advisor Transformation Extension
MB	Model Base
MBT	Model-Based Testing
MDA	Model-Driven Architecture
MDE	Model-Driven Engineering
MDRE	Model-Driven Reverse Engineering
MDS	Model-Driven Science
MDScD	Model-Driven Scenario Development
MEE	Model Experiencing Environment
MetaGME	GME Metamodel

MIC	Model Integrated Computing
MIL	Model-in-the-Loop
MiLEST	Model-in-the-Loop for Embedded System Test
MIM	MOM and Initialization Module
MMM	MSC Metamodel
MMT	Model-to-Model Transformation
MOF	Meta Object Facility
MOM	Management Object Model
MS	Microsoft
M&S	Modeling and Simulation
MSC	Message Sequence Chart
MSDL	Military Scenario Definition Language
MUT	Model Under Test

N

NFL	NSTMSS Foundation Library
NSTMSS	Naval Surface Tactical Maneuvering Simulation System (pronounced as ‘NiSTMiSS’)

O

OM	Object Model
OCL	Object Constraint Language
OLMECO	Open Library for Models of Mechatronic Components
OME	Object Model Editor
OMG	Object Modeling Group
OMT	Object Model Template, Object Modeling Technique
OOA&D	Object Oriented Analysis and Design
OOW	Officer of the Watch
OWL	Web Ontology Language
OSE	Officer Scheduling the Exercise
OTC	Officer in Tactical Command

P

PDU	Protocol Data Unit
PES	Pruned Entity Structure
PIM	Platform-Independent Model
POC	Point of Contact
P-Process	PROMELA Process

PROMELA	Process Metalanguage
P/S	Publish/Subscribe
PSM	Platform-Specific Model
PSML	Platform-Specific Modeling Language
PSMM	P/S Metamodel

Q

QVT	Query/View/Transformation

R

RACoN	RTI Abstraction Component for .Net
RHS	Right Hand Side
RPR	Real-time Platform-level Reference
RID	RTI Initialization Data
RTI	Runtime Infrastructure

S

SASTM	Specialized Abstract Syntax Meta-Models
SBVR	Semantics of Business Vocabulary and Rules
SCM	Simulation Conceptual Model
ScOM	Scenario Object Model
SDEM	Simulation Data Exchange Model
SDL	Scenario Definition Language
SES	System Entity Structure
SES/MB	System Entity Structure and Model Base
ShipFd	Ship Federate
SimGe	SIMulation Generator
SIMNET	Simulation Networking
SISO	Simulation Interoperability Standards Organization
SIL	Software-in-the-Loop
SKDM	Simulation KDM
SM	Scenario Management
SMDL	Simulation Model Definition Language
SME	Subject Matter Expert
SMP2	Simulation Model Portability 2
SOAP	Service-Oriented Access Protocol
SOM	Simulation Object Model
StationFd	Station Federate
STMS	Strait Traffic Monitoring Simulation
SysML	Systems Modeling Language

T

TCP/IP	Transmission Control Protocol/Internet Protocol
TENA	Test and Training Enabling Architecture
TLC	Target Language Compiler
TSONT	Trajectory Simulation Ontology

U

UI	User Interface
UML	Unified Modeling Language
US DoD	United States Department of Defense

V

V&V	Verification and Validation
VV&A	Verification, Validation, and Accreditation

W

W3C	World Wide Web Consortium

X

XML	Extensible Markup Language
XMI	XML Metadata Interchange