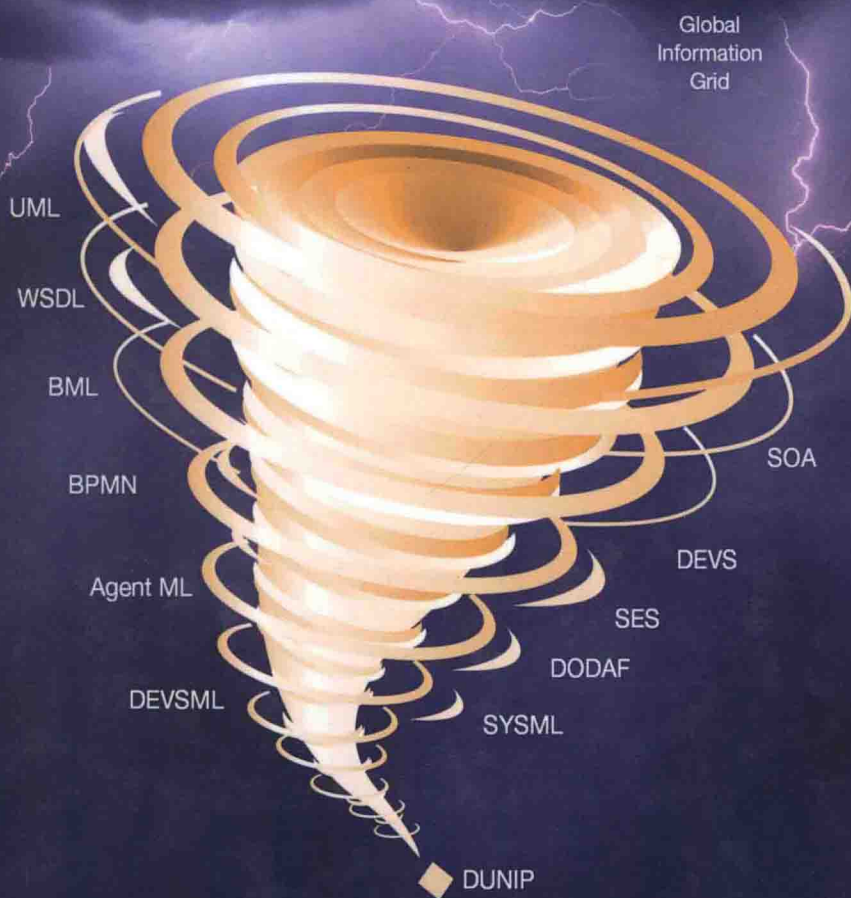


Netcentric System of Systems Engineering with DEVS Unified Process



Saurabh Mittal
José Luis Risco Martín

Netcentric System of Systems Engineering with DEVS Unified Process

Saurabh Mittal
José Luis Escobar Martín

With contributions by:

Deniz Cetinkaya • Alexander Verbraeck • Mamadou D. Seck



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **Informa** business

MATLAB® is a trademark of The MathWorks, Inc. and is used with permission. The MathWorks does not warrant the accuracy of the text or exercises in this book. This book's use or discussion of MATLAB® software or related products does not constitute endorsement or sponsorship by The MathWorks of a particular pedagogical approach or particular use of the MATLAB® software.

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2013 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper
Version Date: 20121214

International Standard Book Number: 978-1-4398-2706-2 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Netcentric System of Systems Engineering with DEVS Unified Process

Saurabh Mittal

José Luis Risco Martín

With contributions by

David Cortezaya • Alexander Gaborauer • Marianne El Gork



CRC Press

Taylor & Francis Group

London • New York • Washington, DC

1000 Lakeside Drive, Suite 1000, Boca Raton, FL 33431

9781439800000 (Hbk) 9781439800017 (Pbk)

Visit us online at www.crcpress.com

SYSTEM OF SYSTEMS ENGINEERING SERIES

Series Editor: Mo Jamshidi

Discrete-Time Inverse Optimal Control for Nonlinear Systems,
Edgar N. Sanchez and Fernando Ornelas-Tellez

Netcentric System of Systems Engineering with DEVS Unified Process,
Saurabh Mittal and José L. Risco Martín

Contents

Chapter 1: Introduction to System Modeling and Simulation	1
Chapter 2: Fundamentals of Modeling and Simulation	15
Chapter 3: System Modeling and Simulation with MATLAB	35

SECTION 1: Introduction

Dedication

This book is dedicated to our families, who have never doubted our dreams, no matter how crazy they might be. Specially dedicated to our kids, who are at the age of asking the question “Why?” We love our kids as they are, and we love that question as scientists. The answer is usually reached in simulation,

... in our minds and ... beyond.

Chapter 4: Modeling and Simulation with MATLAB	55
Chapter 5: Modeling and Simulation with Simulink	75
Chapter 6: Modeling and Simulation with State-Space Models	95
Chapter 7: Modeling and Simulation with Control Systems	115
Chapter 8: Modeling and Simulation with Optimization	135
Chapter 9: Modeling and Simulation with Data Analysis	155
Chapter 10: Modeling and Simulation with System Identification	175
Chapter 11: Modeling and Simulation with System Analysis	195
Chapter 12: Modeling and Simulation with System Design	215
Chapter 13: Modeling and Simulation with System Verification	235
Chapter 14: Modeling and Simulation with System Validation	255
Chapter 15: Modeling and Simulation with System Testing	275
Chapter 16: Modeling and Simulation with System Deployment	295
Chapter 17: Modeling and Simulation with System Maintenance	315
Chapter 18: Modeling and Simulation with System Upgrade	335
Chapter 19: Modeling and Simulation with System Retirement	355
Chapter 20: Modeling and Simulation with System Decommissioning	375
Chapter 21: Modeling and Simulation with System Disposal	395
Chapter 22: Modeling and Simulation with System Recycling	415
Chapter 23: Modeling and Simulation with System Reuse	435
Chapter 24: Modeling and Simulation with System Reproduction	455
Chapter 25: Modeling and Simulation with System Evolution	475
Chapter 26: Modeling and Simulation with System Transformation	495
Chapter 27: Modeling and Simulation with System Migration	515
Chapter 28: Modeling and Simulation with System Integration	535
Chapter 29: Modeling and Simulation with System Interoperability	555
Chapter 30: Modeling and Simulation with System Compatibility	575
Chapter 31: Modeling and Simulation with System Conformance	595
Chapter 32: Modeling and Simulation with System Compliance	615
Chapter 33: Modeling and Simulation with System Certification	635
Chapter 34: Modeling and Simulation with System Accreditation	655
Chapter 35: Modeling and Simulation with System Approval	675
Chapter 36: Modeling and Simulation with System Authorization	695
Chapter 37: Modeling and Simulation with System Acceptance	715
Chapter 38: Modeling and Simulation with System Release	735
Chapter 39: Modeling and Simulation with System Distribution	755
Chapter 40: Modeling and Simulation with System Delivery	775
Chapter 41: Modeling and Simulation with System Installation	795
Chapter 42: Modeling and Simulation with System Configuration	815
Chapter 43: Modeling and Simulation with System Management	835
Chapter 44: Modeling and Simulation with System Operation	855
Chapter 45: Modeling and Simulation with System Performance	875
Chapter 46: Modeling and Simulation with System Reliability	895
Chapter 47: Modeling and Simulation with System Availability	915
Chapter 48: Modeling and Simulation with System Maintainability	935
Chapter 49: Modeling and Simulation with System Supportability	955
Chapter 50: Modeling and Simulation with System Testability	975
Chapter 51: Modeling and Simulation with System Verifiability	995
Chapter 52: Modeling and Simulation with System Validity	1015
Chapter 53: Modeling and Simulation with System Usability	1035
Chapter 54: Modeling and Simulation with System Interactability	1055
Chapter 55: Modeling and Simulation with System Learnability	1075
Chapter 56: Modeling and Simulation with System Adaptability	1095
Chapter 57: Modeling and Simulation with System Resiliency	1115
Chapter 58: Modeling and Simulation with System Robustness	1135
Chapter 59: Modeling and Simulation with System Fault Tolerance	1155
Chapter 60: Modeling and Simulation with System Security	1175
Chapter 61: Modeling and Simulation with System Privacy	1195
Chapter 62: Modeling and Simulation with System Confidentiality	1215
Chapter 63: Modeling and Simulation with System Integrity	1235
Chapter 64: Modeling and Simulation with System Authenticity	1255
Chapter 65: Modeling and Simulation with System Accountability	1275
Chapter 66: Modeling and Simulation with System Transparency	1295
Chapter 67: Modeling and Simulation with System Openness	1315
Chapter 68: Modeling and Simulation with System Accessibility	1335
Chapter 69: Modeling and Simulation with System Inclusivity	1355
Chapter 70: Modeling and Simulation with System Diversity	1375
Chapter 71: Modeling and Simulation with System Complexity	1395
Chapter 72: Modeling and Simulation with System Uncertainty	1415
Chapter 73: Modeling and Simulation with System Risk	1435
Chapter 74: Modeling and Simulation with System Impact	1455
Chapter 75: Modeling and Simulation with System Consequence	1475
Chapter 76: Modeling and Simulation with System Burden	1495
Chapter 77: Modeling and Simulation with System Load	1515
Chapter 78: Modeling and Simulation with System Stress	1535
Chapter 79: Modeling and Simulation with System Strain	1555
Chapter 80: Modeling and Simulation with System Tension	1575
Chapter 81: Modeling and Simulation with System Pressure	1595
Chapter 82: Modeling and Simulation with System Force	1615
Chapter 83: Modeling and Simulation with System Power	1635
Chapter 84: Modeling and Simulation with System Energy	1655
Chapter 85: Modeling and Simulation with System Work	1675
Chapter 86: Modeling and Simulation with System Activity	1695
Chapter 87: Modeling and Simulation with System Motion	1715
Chapter 88: Modeling and Simulation with System Change	1735
Chapter 89: Modeling and Simulation with System Development	1755
Chapter 90: Modeling and Simulation with System Growth	1775
Chapter 91: Modeling and Simulation with System Expansion	1795
Chapter 92: Modeling and Simulation with System Extension	1815
Chapter 93: Modeling and Simulation with System Enhancement	1835
Chapter 94: Modeling and Simulation with System Improvement	1855
Chapter 95: Modeling and Simulation with System Advancement	1875
Chapter 96: Modeling and Simulation with System Progress	1895
Chapter 97: Modeling and Simulation with System Success	1915
Chapter 98: Modeling and Simulation with System Achievement	1935
Chapter 99: Modeling and Simulation with System Attainment	1955
Chapter 100: Modeling and Simulation with System Realization	1975

Chapter 101: Modeling and Simulation with System Fulfillment	1995
Chapter 102: Modeling and Simulation with System Completion	2015
Chapter 103: Modeling and Simulation with System Accomplishment	2035
Chapter 104: Modeling and Simulation with System Attainment	2055
Chapter 105: Modeling and Simulation with System Achievement	2075
Chapter 106: Modeling and Simulation with System Success	2095
Chapter 107: Modeling and Simulation with System Progress	2115
Chapter 108: Modeling and Simulation with System Advancement	2135
Chapter 109: Modeling and Simulation with System Improvement	2155
Chapter 110: Modeling and Simulation with System Enhancement	2175
Chapter 111: Modeling and Simulation with System Extension	2195
Chapter 112: Modeling and Simulation with System Expansion	2215
Chapter 113: Modeling and Simulation with System Growth	2235
Chapter 114: Modeling and Simulation with System Development	2255
Chapter 115: Modeling and Simulation with System Change	2275
Chapter 116: Modeling and Simulation with System Motion	2295
Chapter 117: Modeling and Simulation with System Activity	2315
Chapter 118: Modeling and Simulation with System Work	2335
Chapter 119: Modeling and Simulation with System Energy	2355
Chapter 120: Modeling and Simulation with System Power	2375
Chapter 121: Modeling and Simulation with System Force	2395
Chapter 122: Modeling and Simulation with System Pressure	2415
Chapter 123: Modeling and Simulation with System Strain	2435
Chapter 124: Modeling and Simulation with System Stress	2455
Chapter 125: Modeling and Simulation with System Load	2475
Chapter 126: Modeling and Simulation with System Burden	2495
Chapter 127: Modeling and Simulation with System Consequence	2515
Chapter 128: Modeling and Simulation with System Impact	2535
Chapter 129: Modeling and Simulation with System Risk	2555
Chapter 130: Modeling and Simulation with System Uncertainty	2575
Chapter 131: Modeling and Simulation with System Complexity	2595
Chapter 132: Modeling and Simulation with System Diversity	2615
Chapter 133: Modeling and Simulation with System Inclusivity	2635
Chapter 134: Modeling and Simulation with System Accessibility	2655
Chapter 135: Modeling and Simulation with System Interactability	2675
Chapter 136: Modeling and Simulation with System Learnability	2695
Chapter 137: Modeling and Simulation with System Adaptability	2715
Chapter 138: Modeling and Simulation with System Resiliency	2735
Chapter 139: Modeling and Simulation with System Robustness	2755
Chapter 140: Modeling and Simulation with System Fault Tolerance	2775
Chapter 141: Modeling and Simulation with System Security	2795
Chapter 142: Modeling and Simulation with System Privacy	2815
Chapter 143: Modeling and Simulation with System Confidentiality	2835
Chapter 144: Modeling and Simulation with System Integrity	2855
Chapter 145: Modeling and Simulation with System Authenticity	2875
Chapter 146: Modeling and Simulation with System Accountability	2895
Chapter 147: Modeling and Simulation with System Transparency	2915
Chapter 148: Modeling and Simulation with System Openness	2935
Chapter 149: Modeling and Simulation with System Availability	2955
Chapter 150: Modeling and Simulation with System Maintainability	2975
Chapter 151: Modeling and Simulation with System Supportability	2995
Chapter 152: Modeling and Simulation with System Testability	3015
Chapter 153: Modeling and Simulation with System Verifiability	3035
Chapter 154: Modeling and Simulation with System Validity	3055
Chapter 155: Modeling and Simulation with System Usability	3075
Chapter 156: Modeling and Simulation with System Interactability	3095
Chapter 157: Modeling and Simulation with System Learnability	3115
Chapter 158: Modeling and Simulation with System Adaptability	3135
Chapter 159: Modeling and Simulation with System Resiliency	3155
Chapter 160: Modeling and Simulation with System Robustness	3175
Chapter 161: Modeling and Simulation with System Fault Tolerance	3195
Chapter 162: Modeling and Simulation with System Security	3215
Chapter 163: Modeling and Simulation with System Privacy	3235
Chapter 164: Modeling and Simulation with System Confidentiality	3255
Chapter 165: Modeling and Simulation with System Integrity	3275
Chapter 166: Modeling and Simulation with System Authenticity	3295
Chapter 167: Modeling and Simulation with System Accountability	3315
Chapter 168: Modeling and Simulation with System Transparency	3335
Chapter 169: Modeling and Simulation with System Openness	3355
Chapter 170: Modeling and Simulation with System Availability	3375
Chapter 171: Modeling and Simulation with System Maintainability	3395
Chapter 172: Modeling and Simulation with System Supportability	3415
Chapter 173: Modeling and Simulation with System Testability	3435
Chapter 174: Modeling and Simulation with System Verifiability	3455
Chapter 175: Modeling and Simulation with System Validity	3475
Chapter 176: Modeling and Simulation with System Usability	3495
Chapter 177: Modeling and Simulation with System Interactability	3515
Chapter 178: Modeling and Simulation with System Learnability	3535
Chapter 179: Modeling and Simulation with System Adaptability	3555
Chapter 180: Modeling and Simulation with System Resiliency	3575
Chapter 181: Modeling and Simulation with System Robustness	3595
Chapter 182: Modeling and Simulation with System Fault Tolerance	3615
Chapter 183: Modeling and Simulation with System Security	3635
Chapter 184: Modeling and Simulation with System Privacy	3655
Chapter 185: Modeling and Simulation with System Confidentiality	3675
Chapter 186: Modeling and Simulation with System Integrity	3695
Chapter 187: Modeling and Simulation with System Authenticity	3715
Chapter 188: Modeling and Simulation with System Accountability	3735
Chapter 189: Modeling and Simulation with System Transparency	3755
Chapter 190: Modeling and Simulation with System Openness	3775
Chapter 191: Modeling and Simulation with System Availability	3795
Chapter 192: Modeling and Simulation with System Maintainability	3815
Chapter 193: Modeling and Simulation with System Supportability	3835
Chapter 194: Modeling and Simulation with System Testability	3855
Chapter 195: Modeling and Simulation with System Verifiability	3875
Chapter 196: Modeling and Simulation with System Validity	3895
Chapter 197: Modeling and Simulation with System Usability	3915
Chapter 198: Modeling and Simulation with System Interactability	3935
Chapter 199: Modeling and Simulation with System Learnability	3955
Chapter 200: Modeling and Simulation with System Adaptability	3975

Preface

The foundation for this book was laid during Dr. Saurabh Mittal's doctoral study (2005 through 2007) at the Arizona Center for Integrative Modeling and Simulation (ACIMS), earlier known as the Artificial Intelligence Simulation Laboratory, in the Department of Electrical and Computer Engineering of the University of Arizona, Tucson, under the advisement of Professor Bernard P. Zeigler, the pioneer of Discrete Event Systems Specification (DEVS) formalism. Dr. Mittal was involved in multiple projects such as the modernization of the High Frequency Global Communications System (HFGCS) simulation model for the U.S. Air Force, the Automated Test-Case Generator (ATC-Gen) at the Joint Interoperability Test Command (JITC), Ft. Huachuca, and various other efforts. These projects were accomplished at ACIMS, which acted as a subcontractor to Northrop Grumman Information Technology (NGIT), JITC, Ft. Huachuca. The methodology adopted for all the projects had recurring themes of model specification, transformation, coding, testing, and execution. Most of the time, specifications directly led to executable code in DEVJSJAVA (©). Sometime in 2006, Dr. Mittal coauthored an eXtensible Markup Language (XML) representation based on Finite Deterministic DEVS (XFD-DEVS) that allowed automated transformation of models into XML and executable code. Integration of XML with DEVS was a synergistic development as the ATC-Gen project also pushed the XML-based System Entity Structure (SES) ontology development. The research on platform-independent DEVS modeling was just beginning.

During the summer of 2006, Dr. José Luis Risco Martín arrived at the ACIMS Laboratory as a visiting research scholar. A strong collaboration was forged during his two-month stay in Tucson. Prior to coming at ACIMS lab, Dr. Martin had already developed the XML version of DEVS coupled models that acted as the common ground for the ensuing collaboration. Drs. Mittal and Risco Martín exchanged a lot of ideas, and the first version of the DEVS Modeling Language (DEVSMML) stack was conceived at that point. They developed the execution of DEVS using Web Services technology and also developed the DEVS/Service Oriented Architecture (SOA) framework. They now had an architecture that takes in platform-independent XML-based DEVS model and executes them on a netcentric platform. The seeds of a platform-neutral DEVS Virtual Machine (DEVSVML) were laid.

During his doctoral study, Dr. Mittal developed many transformations for sources like XML-based state machines, structured natural language message-based systems, the Department of Defense Architecture Framework (DoDAF) (Version 1.0), Business Process Modeling Notation (BPMN), Unified Modeling Language (UML), XFD-DEVS, and so on, that all led to executable DEVS. Based on the bifurcated model-continuity-based life cycle methodology developed during the ATC-Gen project and various efforts in developing the DEVS back-end, the DEVS Unified Process (DUNIP) was conceptualized in 2007.

As a research scientist at L-3 Communications and a contractor to the U.S. Air Force Research Laboratory (AFRL), at Wright-Patterson Air Force Base, Ohio,

Dr. Mittal is now implementing the DEVSML stack at various AFRL initiatives. A new version, DEVSML 2.0, was proposed as a part of the Large-scale Cognitive Modeling (LSCM) initiative that integrates domain-specific languages (DSLs) with the prior version of DEVSML stack. A DEVS DSL called DEVSML, a platform-independent language in the DEVSML 2.0 stack, was also proposed that redefines the way DEVS models are written in a structured ecosystem such as Eclipse.

XML was not the only means to DEVS transformation, and Drs. Mittal and Risco Martín ventured into the domain of model transformations and model-to-DEVs transformation through a platform-independent DEVS DSL. Various tools were explored, most notably the Eclipse Modeling Framework (EMF), Eclipse Graphical Modeling Framework (GMF), Xtext, Xpand, Generic Modeling Environment, Groovy, Scala, Ruby, and so on that allow the development of DSLs. As DEVSML is an integral part of DUNIP, DUNIP was updated to include DSLs as a means to incorporate other domains that want to interface with a netcentric DEVSVM.

While Dr. Mittal was making advances at the modeling level, Dr. Risco Martín was making advances in the DEVS to non-DEVs interoperability using XML message communication and using the DEVS netcentric platform to run large-scale performance studies. Based on the advances and the capabilities of netcentric DEVS, their work was a strong force in the development of a DEVS Modeling and Simulation (M&S) Standard.

In 2011, Drs. Mittal and Risco Martín updated their server-side capabilities to the latest Web Service specification standards such as JAX-WS and developed advanced concepts for session replication and platform scalability. They also developed various domain-authoring tools in Eclipse and NetBeans ecosystems. This book is an outcome of the application of a lot of foundational ideas by the DEVS community and the latest in netcentric DEVS systems engineering by the authors. This book presents the latest on DUNIP and how it makes model-driven engineering (MDE) concepts applicable to the DEVS and at-large communities. DUNIP creates a possibility for wider adoption of DEVS systems engineering methodologies by integrating various DSLs, existing or non-existing, to a formal computational complex dynamical systems framework. This book is largely based on the authors' experience with the DEVS formalism and application of MDE in the context of netcentric system of systems.

HOW TO USE THIS BOOK

The authors expect the reader to be aware of the basic principles of object-oriented programming. The book is divided into five sections with each section covering a particular community of interest. A summary of each chapter is provided below.

SECTION I: BASICS

Section I is designed for the undergraduate students and novices who want to venture into the DEVS world. The students can refer to some of the case studies in Section IV.

Chapter 1: Introduction to Systems Modeling and Simulation

Chapter 1 provides the basics of computational M&S. This chapter also provides an overview of UML and how various concepts of Object-oriented modeling are applied to M&S.

Chapter 2: System of Systems Modeling and Simulation with DEVS

Chapter 2 provides definitions of system of systems (SoS), complex systems, complex adaptive systems (CAS), and netcentric SoS. This chapter also provides the DEVS framework for M&S.

Chapter 3: DEVS Formalism and Variants

Chapter 3 describes DEVS formalism and its computational representation in Java. Various examples are provided that aid understanding of DEVS systems specification concepts.

Chapter 4: DEVS Software: Model and Simulator

Chapter 4 provides more examples and the computational representation of DEVS simulators. This chapter gives in-depth descriptions on how to write your own DEVS simulators. While this chapter and Chapter 3 give computational representations, Chapter 5 provides details on DEVS modeling without any programming language constructs.

Chapter 5: DEVS Modeling Language

Chapter 5 describes a DEVS DSL. This chapter is designed for early DEVS adopters who want to delve into the DEVS world at a much coarser level. This chapter provides an Eclipse editor to allow creation of platform-independent DEVS models. Various design issues of DEVS grammar are explained and its XML representation is also presented.

Chapter 6: DEVS Unified Process

Chapter 6 provides an overview of DUNIP and how it can be realized in a systems M&S effort. This chapter also describes the DEVSMML 2.0 stack, which is the linchpin of netcentric DEVS systems engineering. This chapter binds the foundational concepts and presents a vision of an integrated spiral methodology using DUNIP.

SECTION II: MODELING AND SIMULATION-BASED SYSTEMS ENGINEERING

Section II is for graduate students and advanced practitioners of DEVS. It is also for industry professionals who would like to learn the advanced capabilities of DEVS-based systems engineering methodology and to use MDE in their efforts. This section provides methodologies to apply M&S principles to SoS design and also provides an overview on the development of executable architectures based on architecture framework like DoDAF.

Chapter 7: Reconfigurable DEVS

Chapter 7 describes the variable structure capability in the DEVS component-based M&S framework. This chapter also presents the dynamic structure capability through the use of experimental frames and enhanced Model View Simulator Controller (MSVC) paradigm.

Chapter 8: Real-Time and Virtual-Time DEVS

Chapter 8 describes real-time and virtual-time DEVS and provides more details on the development of real-time systems.

Chapter 9: Model-Driven Engineering and Its Application to Modeling and Simulation

Chapter 9 is a collaborative effort with Deniz Cetinkaya, Dr. Alexander Verbraeck, and Dr. Mamadou D. Seck, all at the Delft University of Technology, The Netherlands. Deniz is a PhD student, Dr. Seck is an assistant professor, and Dr. Verbraeck is a professor, and they have all contributed their expertise in sharing the authors' vision. This chapter formally presents a framework for model-driven development (MDD) as applicable to M&S domain.

Chapter 10: System Entity Structures and Contingency-Based Systems

Chapter 10 presents SES formalism, its computational representation, a constraint-based language, and a larger Knowledge-Based Contingency Driven Generative Systems (KCGS) framework for designing contingency-based artificial systems with DUNIP. The foundation of KCGS was laid at Cognitive Models and Agents Branch, AFRL, where a Cognitive Systems Specification Framework (CS2F) was developed, a realization of KCGS.

Chapter 11: Department of Defense Architecture Framework (Version 1.0)

Chapter 11 dives into the area of architecture frameworks and describes the U.S. DoDAF Version 1.0. This chapter analyzes the DoDAF specification document, finds gaps in the specification, and suggests augmenting the DoDAF 1.0 specification with two new documents that allow the M&S community to build executable architectures.

Chapter 12: Modeling and Simulation-Based Testing and DoDAF Compliance

Chapter 12 gives an overview on various model-based testing methodologies at various levels of system specifications and on how architecture frameworks such as DoDAF can be subjected to test and evaluation studies.

SECTION III: NETCENTRIC SYSTEM OF SYSTEMS

Section III is for graduate students, advanced users of DEVS, and industry professionals who are interested in building DEVS virtual machines and netcentric SoS. This section can be read directly after Section I.

Chapter 13: DEVS Standard

Chapter 13 gives an overview of the complexities in standardization efforts at the modeling level as well as at the simulation level. This chapter also describes the integration of DEVS and non-DEVS (MATLAB®) systems. Examples are presented in sufficient detail.

Chapter 14: Architecture for DEVS/SOA

Chapter 14 describes architecture for DEVS/SOA and the DEVS Virtual Machine. This chapter provides details about the server and client designs and how the DEVSML 2.0 stack is implemented in a netcentric infrastructure.

Chapter 15: Model and Simulator Deployment in a Netcentric Environment

After an understanding of the netcentric DEVS is given in Chapter 14, Chapter 15 provides practical examples on how to deploy the models at the client side and the simulators at the server side. Various netcentric architectural constructs are described in sufficient detail.

Chapter 16: Netcentric System of Systems with DEVS-Based Event-Driven Architectures

Chapter 16 integrates the emerging paradigm of Event Driven Architecture (EDA) with the netcentric DEVSML 2.0, the DEVSVSM, and MDE toward the development of agile netcentric DEVS-based EDAs. This chapter also provides details about integrating the Web Service Description Language (WSDL), a DSL, into the DEVSML 2.0 stack.

Chapter 17: Metamodeling in Department of Defense Architecture Framework (Version 2.0) Metamodel

Chapter 17 revisits the DoDAF with its latest version, which is oriented toward data engineering, sharing, and interoperability. This chapter represents DoDAF 2.0 viewpoints with the SES ontology and presents an argument that DoDAF 2.0 is now in a state where it can effectively act as a DSL that can be anchored to a formal M&S framework. This chapter addresses issues of interoperability and integration of DoDAF DSL based on SES ontology to the netcentric DEVSVSM.

SECTION IV: CASE STUDIES

Section IV presents the case studies that realize many of the concepts defined in Sections I–III.

Chapter 18: Joint Close Air Support: Building from Informal Scenarios

Chapter 18 provides a complete example using DEVSML. Joint Close Air Support (JCAS) is a U.S. Joint Measure Thread (JMT) that describes the JCAS scenario. The user is led from informal natural language specification through a requirements refinement process into a DEVSML specification that is executable. This chapter can be read in conjunction with Chapter 5.

Chapter 19: DEVS Simulation Framework for Multiple Unmanned Aerial Vehicles in Realistic Scenarios

Chapter 19 brings a realistic context in the specification of scenarios. This chapter presents a simulation framework executable on DEVS/SOA in the context of the modeling of multiple unmanned aerial vehicles. Instead of DEVSML, the model is

directly built into the implementation programming language. Various performance studies are conducted in a netcentric DEVS/SOA infrastructure. This chapter can be read in conjunction with Chapters 4 and 15.

Chapter 20: Generic Network Systems Capable of Planned Expansion: From Monolithic to Netcentric Systems

Chapter 20 presents an overview of the state-of-the-art HFGCS simulation model, also known as the Generic Network System Capable of Planned Expansion (GENETSCOPE). A methodology is presented that made a monolithic system, a modular system that is executable in a DEVS component-based framework. Further ideas are presented to make it netcentric. This chapter can be read in conjunction with Chapters 7 and 16.

Chapter 21: Executable UML

Chapter 21 takes a hard look at the UML framework and develops various transformations that take UML elements to the DEVS world and vice versa. This chapter can be read in conjunction with Chapters 5, 6, and 10.

Chapter 22: BPMN to DEVS: Application of MDD4MS Framework in Discrete Event Simulation

Chapter 22 takes BPMN to the DEVS executable using the MDD4MS framework developed in collaboration with Centikaya, Dr. Verbraeck, and Dr. Seck. This chapter describes a BPMN metamodel and the transformations to a DEVS metamodel. The metamodel is also integrated with the DEVSML metamodel. This chapter can be read in conjunction with Chapters 5 and 9.

SECTION V: NEXT STEPS

Section V presents the application of DEVS extensions in a netcentric environment to the discipline of CAS.

Chapter 23: Netcentric Complex Adaptive Systems

Chapter 23 provides an overview of CAS and how modeling of CAS using DEVS concepts can be attempted. This chapter also conceptualizes a netcentric CAS and argues that a netcentric SoS becomes a CAS. Finally, this chapter provides various features and requirements to achieve integration and interoperability in a netcentric CAS.

ONLINE

Various other supporting materials are available online at:
<http://www.duniptechnologies.com/book/sos>.

Exercises

Some of the chapters come with exercises that advance the idea and concept described earlier. A more serious student is encouraged to attempt these. Partial solutions to the exercises are available online at:

<http://www.duniptechnologies.com/book/sos/exercises>.

Software

Instructions to use and install the software used in this book are available online at:

<http://www.duniptechnologies.com/book/sos/software>.

WHAT THIS BOOK IS NOT

This book is not about learning a programming language or a specific tool. It is about the application of MDE concepts and how to build models of complex dynamical that are executable on a DEVS platform. Specifically, this book will not help if you are looking to learn the following:

- Advanced programming skills in Java or associated implementation languages
- DSL development tools like Xtext, Xpand, EMF, Eclipse GMF, or eCore
- Server administration (Glassfish 3.x or Tomcat 6.x)
- Graphical user interface development

QUESTIONS AND CONTACT INFORMATION

For any questions, concerns, or errors in this book, please send an e-mail to: sosbook@duniptechnologies.com.

DEVJSJAVA is developed by University of Arizona, Tucson, AZ, USA Copyright © 1996–2000. All rights reserved. The software in part or in whole is not transferable to any other party, individual or entity without explicit permission from the University of Arizona's Office of Technology Transfer or Arizona Board of Regents.

MATLAB® is a registered trademark of The MathWorks, Inc. For product information, please contact:

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098, USA
Tel: 508 647 7000
Fax: 508-647-7001
E-mail: info@mathworks.com
Web: www.mathworks.com

Acknowledgments

In this arduous journey of two years, we have to express many thanks to our families and our young kids. Both of us became parents during the manuscript preparation phase. We especially thank our wives for those moments in which they took care of the home and especially the kids, who are now toddlers, while we stayed away working on the book. We wholeheartedly thank our families for their unconditional love, patience, kindness, understanding, and encouragement.

We are grateful toward our series editor, Professor Mo Jamshidi, and the publisher, Taylor & Francis Group/CRC Press, for finding merit in our work and giving us an opportunity to develop it further.

We sincerely thank Professor Bernard Zeigler for his support in conceptualizing the ideas in the early stages of our work at the ACIMS Laboratory. We came to know many people at NGIT who helped create the foundation of DUNIP. We thank all our friends and colleagues at NGIT for their support.

We would like to thank Deniz Cetinkaya, Alexander Verbraeck, and Mamadou Seck at Delft University of Technology for collaborating on two chapters for this book. Their important contributions are very much appreciated.

We especially thank our friends and colleagues at L-3 Communications, AFRL, and the Department of Computer Architecture and Automation (Universidad Complutense de Madrid [UCM], Spain).

Finally, we thank our friends who collaborated on the development of certain ideas and experiments given in this book. We thank Jesús Manuel de la Cruz and Eva Besada (UCM), and Joaquín Aranda and Alejandro Moreno (Universidad Nacional de Educación a Distancia, Spain).

Saurabh Mittal
José Luis Risco Martín

Authors

Saurabh Mittal is the founder and principal scientist at Dunip Technologies, which he manages in his free time. He is currently a full-time research scientist at L-3 Communications and is a contractor to the U.S. Air Force Research Laboratory, Wright-Patterson Air Force Base, Ohio. Here in this capacity, he is working on large-scale cognitive M&S using DEVS systems engineering, cognitive domain ontologies extending SES theory and various other cross-directorate M&S integration and interoperability efforts using architecture frameworks such as Department of Defense Architecture Framework. He is a recipient of the highest civilian contractor recognition “Golden Eagle” award by Joint Interoperability Test Command, Defense Information Systems Agency, U.S. Department of Defense. He serves on various conference program committees and is a reviewer for many prestigious international journals. His areas of interest include web-based M&S using SOA, executable architectures, netcentricity, DUNIP, domain ontologies, platform independent modeling, domain specific languages, SES theory and applications, parallel-distributed simulation, complex systems, complex adaptive systems, and system of systems engineering using DoDAF.

José L. Risco Martín is an associate professor at the Computer Architecture and Automation Department of UCM, Spain. His research interests focus on the design methodologies for integrated systems and high-performance embedded systems, including new modeling frameworks to explore thermal management techniques for multiprocessor system-on-chip, novel architectures for logic and memories in forthcoming nanoscale electronics, dynamic memory management and memory hierarchy optimizations for embedded systems, networks-on-chip interconnection design, and low-power design of embedded systems. He is also interested in theory of M&S, with an emphasis on DEVS, and the application of bioinspired optimization techniques in computer-aided design problems.

Contents

Preface.....	xvii
Acknowledgments.....	xxv
Authors.....	xxvii

SECTION I *The Basics*

Chapter 1	Introduction to Systems Modeling and Simulation.....	3
1.1	The Nature of Simulation	3
1.2	Systems, Models, and Modeling.....	4
1.2.1	Systems.....	4
1.2.2	Model of a System	5
1.2.3	Modeling	6
1.3	Phases in Model Development	7
1.4	Types of Simulation Models	7
1.4.1	Stochastic or Deterministic	7
1.4.2	Steady State or Dynamic	8
1.4.3	Continuous or Discrete.....	8
1.4.4	Local or Distributed	8
1.5	Examples of Models	8
1.5.1	Population Growth	8
1.5.2	Particle in a Potential Field	9
1.5.3	Amortization Process Model.....	9
1.6	Software Systems Engineering with Objects, Classes, and UML	9
1.6.1	UML for Object-Oriented Modeling.....	10
1.6.2	Introduction to Java	17
	References	29
Chapter 2	System of Systems Modeling and Simulation with DEVS	31
2.1	Definitions	31
2.2	DEVS Hierarchy of Systems Specification	34
2.2.1	Hierarchy of Systems Specification	35
2.2.2	DEVS System Components.....	36
2.3	Framework for M&S	37
2.3.1	Computational Representation	39
2.4	Summary	39
	References	41

Chapter 3	DEVS Formalism and Variants	43
3.1	DEVS Formalism	45
3.1.1	Classic DEVS Formalism	45
3.1.2	P-DEVS Formalism	51
3.2	Well-Defined Systems and Legitimacy	54
3.3	DEVS Model Example	55
3.4	DEVS Representation of Quantized Systems	58
3.5	DEVS Representation of Systems	62
3.5.1	DTSS Models	62
3.5.2	DESS Models	67
	References	67
Chapter 4	DEVS Software: Model and Simulator	69
4.1	Introduction	69
4.2	DEVS Modeling Metamodel	71
4.2.1	Port Class	71
4.2.2	Component Class	76
4.2.3	Atomic Class	78
4.2.4	Coupling Class	86
4.2.5	Coupled Model	87
4.3	DEVS Simulation Metamodel	88
4.3.1	DevsSimulator Class	91
4.3.2	Simulator Class	92
4.3.3	Coordinator Class	93
4.4	Simulation of Coupled Models	96
4.4.1	Pulse	99
4.4.2	Ramp	100
4.4.3	Experimental Frame and Processor Model	104
	References	113
Chapter 5	DEVS Modeling Language	115
5.1	Language	115
5.1.1	Entity	116
5.1.2	Atomic	117
5.1.3	Coupled	121
5.2	Dynamic Code Generation	123
5.2.1	Entity	125
5.2.2	Atomic	127
5.2.3	Coupled	138
5.2.4	Execution of DEVS Models	141
5.3	Illustration	142
5.3.1	Installation and Setup	142
5.3.2	EFP Model	142