
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>

Bernard P. Zeigler · Hessam S. Sarjoughian

Guide to Modeling and Simulation of Systems of Systems

Second Edition

With contributions by Raphaël Duboz and Jean-Christophe Soulié

 Springer

Authors

Bernard P. Zeigler
University of Arizona
Tucson, AZ
USA

With contributions by

Raphaël Duboz
CIRAD
Montpellier
France

Hessam S. Sarjoughian
Faculty of Computer Science
and Computer Systems Engineering
Arizona State University, School of
Computing, Informatics, and Decision
Systems Engineering
Tempe, AZ
USA

Jean-Christophe Soulié
CIRAD
Montpellier
France

ISSN 2195-2817 ISSN 2195-2825 (electronic)
Simulation Foundations, Methods and Applications
ISBN 978-3-319-64133-1 ISBN 978-3-319-64134-8 (eBook)
DOI 10.1007/978-3-319-64134-8

Library of Congress Control Number: 2017947847

1st edition: © Springer-Verlag London 2013

2nd edition: © Springer International Publishing AG 2017

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. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Systems of systems are at the root of this century's global challenges of economy, climate, and energy. We are accustomed to building such systems directly in the real world, yet it is becoming increasingly too dangerous, costly, unethical, or risky to do so. In such cases, the only workable alternative is to build and test within virtual reality. System of systems virtual build and test can be supported by a discrete-event systems specification (DEVS) simulation modeling formalism and a system entity structure (SES) simulation model ontology.

This book guides the reader in the use of software tools based on DEVS and SES to tackle a wide variety of systems of systems problems ranging from artificial systems based on cloud information technology to living systems such as agricultural food crops. Commercial and open-source DEVS Modeling and Simulation Environments are covered in depth.

This book is the first to provide an approach to integrating both energy and information processing requirements into system design. This approach, based on activity concepts that are intrinsic to DEVS-based system design, allows us to virtually build and test systems that are capable of emulating biological systems in their ability to balance their information processing functionalities against the energy and resource expenditure incurred in their use.

This edition adds a new chapter covering DEVS Support for Markov Modeling and Simulation. The new chapter provides some concepts and applications for the related facility that has been developed in MS4 Me since the first edition. The edition also augments Chapters 8 and 16 with material intended to enhance guide to modeling and simulation of systems of systems. We want to take the opportunity to recognize Raphaël Duboz and Jean-Christophe Soulié as co-authors for their authorship of Chapter 17.

Acknowledgement The authors would like to acknowledge the numerous developers and users whose efforts contributed to this project. Specifically, we thank Doohwan Kim, Chungman Seo, Robert Coop, Ranjit Singh, Robert Flasher, Ting-Shen Fu, Weilong Hu, Vignesh Elamvazhuthi, Soroosh Gholami,

Abdurrahman AlShareef, Mostafa Fard, Phillip Hammonds, Miguel Soto, Raphaël Duboz, and Jean-Christophe Soulié for their dedication, hard work, and expertise that brought the DEVS concept to its realization in environments including MS4 ME™, DEVS-Suite, CosMoS, and VLE.

Potomac, USA
Tempe, USA

Bernard P. Zeigler
Hessam S. Sarjoughian

Contents

Part I Basic Concepts

1	Modeling and Simulation of Systems of Systems	3
1.1	Virtual Build and Test	5
1.2	Modeling and Simulation Intrinsic to Virtual Build and Test . . .	6
1.3	Multi-disciplinary Collaboration Using Multi-formalism Modeling	7
1.4	Background in the Literature	9
1.5	Guide to Modeling and Simulation of Systems of Systems . . .	10
2	DEVS Integrated Development Environments	13
2.1	The MS4 Me Is a Modeling and Simulation (M&S) Environment	13
2.1.1	Introduction for the M&S User	14
2.1.2	Introduction for the M&S Developer	16
2.2	Introduction for the M&S Professional	20
2.2.1	System Structure and Behavior	20
2.2.2	Finite Deterministic DEVS (FDDEVS)	21
2.2.3	System Entity Structure (SES).	21
2.3	Jazz Band Continued	23
2.4	Summary	26
	References.	29
3	System Entity Structure Basics	31
3.1	Modeling and Simulation as a Simple Workflow	32
3.2	Decomposition and Coupling	33
3.3	Hierarchical Construction	36
3.4	Summary	42
4	DEVS Natural Language Models and Elaborations	43
4.1	FDDEVS Model for Generating Jobs in a Time Sequence	43
4.2	FDDEVS Model for Processing Jobs	46
4.3	A Simple Workflow Coupled Model	48
4.4	Elaborating FDDEVS into Fully Capable Models in Java. . . .	49

4.5	Elaborating ProcessorOfJobs into a Java Model	55
4.6	Transducer: Model to Measure Job Completion Time and Throughput	56
4.7	Using Elaboration to Handle Non-deterministic State Transitions	60
4.8	Using Elaboration to Handle Multiple Simultaneous Inputs . . .	62
4.9	Using Elaboration to Generate Multiple Simultaneous Outputs	63
4.10	Model Development Accelerated by the Sequence Diagram . . .	64
4.11	Summary	67
	Appendix: Transducer FDDEVS File (Transducer.dnl)	68
5	Specialization and Pruning	71
5.1	Specializations	71
5.2	Pruning of Specializations	72
5.3	Multiple Occurrences of Specializations	73
5.4	Rules for Adding Specializations: There Are None	76
	5.4.1 Specialization Under Root Entity	76
	5.4.2 Specialization Under Entity Under Aspect	76
	5.4.3 Specialization Under Entity Under Specialization . . .	76
	5.4.4 Specialization Under Entity Besides Another Specialization	77
5.5	Variables and Specializations	78
5.6	Summary	79
6	Aspects and Multi-aspects	81
6.1	Multiple Aspects (Decompositions)	81
	6.1.1 Expressing Different Aspects for Same Entity	81
	6.1.2 Pruning of Aspects	82
	6.1.3 Aspects: Perspectives and Abstractions	83
6.2	Multi-aspects—Multiple Related Decompositions of an Entity	84
	6.2.1 Limitations of Aspects	84
	6.2.2 Multi-aspect Restructuring	85
	6.2.3 Pruning Multi-aspects	86
	6.2.4 Multi-aspect Uniform Coupling	87
	6.2.5 One-to-All and All-to-One Coupling	89
	6.2.6 Hierarchical Construction with Multi-aspects	90
	6.2.7 Uniform Pairwise Coupling	93
	6.2.8 Predefined Coupling Specifications	95
6.3	Summary	97
	References	97

7	Managing Inheritance in Pruning	99
7.1	Creating Instances with Underscore	99
7.2	Specifying the Base Class for Inheritance.	100
7.3	Configuring the Base Class.	100
7.4	Inheritance in Pruning.	102
7.5	Specifying Inheritance from a Child	103
7.6	Summary	105
8	Automated and Rule-Based Pruning and Experimental Execution	107
8.1	Automated Pruning	107
8.1.1	Enumerative Pruning	108
8.1.2	Random Pruning	109
8.2	Context-Free and Context-Sensitive Pruning.	109
8.2.1	Pruning Algorithm for Context-Sensitive Selection	111
8.2.2	Conditional Rule-Based Pruning	113
8.2.3	The Unless or if-not Conditional Rule.	114
8.2.4	Example: Time-Critical Modeling and Simulation	115
8.2.5	Random Selection from Choices Remaining	116
8.3	Executive Control of Experimentation	116
8.3.1	First Level Control of Simulation	117
8.3.2	Second Level of Control with the SES	119
8.3.3	Third Level of Control	120
8.4	Summary	121
	References.	121

Part II Advanced Concepts

9	DEVS Simulation Protocol	125
9.1	DEVS Simulation Protocol	126
9.2	MS4 Me Exposition of the DEVS Simulation Protocol	128
9.2.1	Interface Objects	130
9.2.2	Input and Output Ports	130
9.2.3	FDDEVS Specifications.	131
9.3	Distributed Simulation Implementations of the DEVS Protocol	133
9.3.1	Standard DEVS Protocol	134
9.3.2	Peer Message Exchanging Implementation	135
9.3.3	Real-Time Message Exchanging Implementation.	137
9.4	DEVS Protocol as a Standard for Simulation Interoperability	139
9.4.1	DEVS Protocol with Event-Scheduling Simulator.	139

9.4.2	Lessons for Simulation Interoperability	141
9.5	Summary	141
	References.	146
10	Dynamic Structure: Agent Modeling and Publish/Subscribe	147
10.1	Dynamic Structure and Agent Modeling.	147
10.2	Publish/Subscribe Data Distribution	150
10.2.1	Publisher	151
10.2.2	Subscriber	152
10.2.3	PublishSubscribeRouter	153
10.2.4	Publish Subscribe Operation	154
10.3	Data Distribution Service	155
10.3.1	DEVS Simulation Protocols in DDS	157
10.3.2	DEVS Messages	158
10.3.3	Relating Ports and Topics	159
10.3.4	Summary	162
	References.	167
11	Interest-Based Information Exchange: Mappings and Models.	169
11.1	Background.	169
11.1.1	Example: The Information Framework Applied to Car Purchases	171
11.2	Application to Network Data Collection	173
11.2.1	Network Traffic Data Representations	174
11.3	Mapping Approach	177
11.3.1	Mapping Multi-aspects	179
11.4	DEVS Models that Exchange XML	181
11.4.1	Model for Generating XML Documents	181
11.4.2	DEVS Model for SES-Based XML Mapping	184
11.4.3	Models to Distribute Mappings of Master SES to Interest-Based SESs.	185
11.4.4	Models that Exchange the Same XML	187
11.5	Summary	190
	Appendix: System Entity Structures for Examples	190
	References.	191
12	Languages for Constructing DEVS Models	193
12.1	Constrained Natural Language Specification of Atomic Models	194
12.1.1	Limitations of FDDEVS Models	196
12.1.2	FDDEVS Enhancement Facility.	196
12.1.3	Development Advantages of the Enhancement Facility	198
12.2	Constrained Natural Language Specifications of Hierarchical CoupledModels.	199

12.3	DEVS, UML, and EMF	201
12.4	Summary	202
	Appendix: Formal Definition of FDDEVS	202
	References.	205

Part III Applications

13	Flexible Modeling Support Environments	209
13.1	Supporting Multiple Paths Through Development Process	209
13.2	M&S Tools as Services in a Service-Oriented Architecture	214
13.3	Case Study: Fractionated Satellite Systems.	215
13.3.1	How the MSE Adapts to Types of Stakeholders	216
13.3.2	System Entity Structure (SES): Key Support for MSE Flexibility	219
13.3.3	MSE Implementation: Service-Oriented Architecture.	219
13.3.4	MSE Simulation Service	221
13.3.5	Simulation Using Web Services.	223
13.4	MSE in Operation: An Example Thread.	225
13.5	Summary	229
	Appendix 1: GeneralClusterArchSeS.txt	231
	Appendix 2: Outline of GeneralCusterSeS	233
	Appendix 3: GeneralClusterArchBasicPrune.pes	233
	Appendix 4: GeneralClusterArchMonolithicPrune.pes.	234
	References.	235
14	Service-Based Software Systems	237
14.1	Introduction	237
14.2	Service-Based Software Systems.	238
14.3	Service-Oriented Architecture	240
14.4	SOA-DEVS Simulation Modeling.	241
14.4.1	Primitive Models	242
14.4.2	Composite Models.	243
14.5	SOA-DEVS Model Components.	244
14.5.1	Generic Messages	245
14.5.2	Primitive Service Models.	246
14.5.3	Composite Service Model	249
14.6	Exemplar Simulation Model	250
14.7	Dynamic Structure SOAD.	255
14.7.1	Broker-Executive Model Design	257
14.7.2	Flat and Hierarchical Model Compositions	259
14.8	Summary	262
14.9	Exercises.	263
	References.	264

15	Cloud System Simulation Modeling	265
15.1	Introduction	265
15.2	Software/Hardware Co-Design	266
15.3	SOC-DEVS SW/HW Modeling	268
15.3.1	Software Service System Model	268
15.3.2	Hardware System Model	278
15.3.3	Service System Mapping	282
15.4	Service-Oriented Voice Communication System	285
15.4.1	Basic Measurements	287
15.4.2	Simulation Parameter Estimation	288
15.4.3	Experimentation Setup and Execution	289
15.4.4	Example Simulation Model Results	289
15.5	Summary	291
	References	292
16	Model Development and Execution Process with Repositories, Validation, and Verification	295
16.1	Introduction	295
16.2	Unified Logical, Visual, and Persistent Modeling	297
16.2.1	Simple Network Virus Model	298
16.2.2	Template, Instance Template, and Instance Model Types	298
16.2.3	Simulatable and Non-simulatable Model Types	301
16.2.4	Logical Models	302
16.2.5	Visual Models	303
16.2.6	Complexity Metrics	305
16.2.7	Persistence Models	305
16.2.8	Model Namespaces	306
16.3	CoSMoS Process Life Cycle	308
16.4	Hybrid Software and Hardware Modeling in CoSMoS	311
16.4.1	Hardware Models	313
16.4.2	Software Models	314
16.4.3	Composite Software/Hardware Mapping Models	314
16.4.4	Model Constraints	315
16.5	Guided Model Validation and Constrained Model Verification	317
16.5.1	Model Validation	317
16.5.2	Model Verification	318
16.5.3	Constrained DEVS Model	319
16.5.4	Verification Algorithm	320
16.5.5	Dynamic Model Property Modeling	321
16.5.6	Model Development and Verification in CoSMoS	322
16.5.7	Summary	322
	References	323

17	Modeling and Simulation of Living Systems as Systems of Systems	325
17.1	Challenges for Living System Modeling and Simulation	326
17.2	Why DEVS and VLE for Living System Modeling and Simulation?	327
17.2.1	A Systemic Approach: Emergence and Scale Transfer	327
17.2.2	Heterogeneous Formalisms and Living Systems Complexity	328
17.2.3	VLE and the Experimental Plans	329
17.3	Surveillance and Control in Animal Epidemiology	331
17.3.1	Motivations and Objectives	331
17.3.2	Model Description	332
17.3.3	Simulation Results	333
17.4	Plant Growth Modeling	337
17.4.1	Motivations and Objectives	337
17.4.2	The Ecomeristem Model	338
17.4.3	Overall Functioning	340
17.4.4	Topology	341
17.4.5	Implementation in DEVS	342
17.4.6	Validation	344
17.4.7	Conclusion	346
17.5	Model Continuity for Living Systems	346
17.6	Summary	347
	References	349
18	Activity-Based Implementations of Systems of Systems	351
18.1	Energy and Activity	351
18.2	Prototype System of Systems	353
18.3	Experimental Frame and Timing Requirements	355
18.4	Including Energy and Activity in SoS Models	357
18.5	Review of Activity Concepts	358
18.6	Timing Requirements, Energy, and Activity	360
18.7	SoS Example: Forest Fire Fighting	362
18.8	Relating Activity to Hardware-Implemented SoS	365
18.9	Experimental Test	367
18.10	Summary	369
	References	370
19	DEVS Support for Markov Modeling and Simulation	371
19.1	Introduction	371
19.2	Markov Matrix Model	372
19.2.1	Steady-State Probabilities	373
19.3	CTM and DTM Models	375

19.3.1	Implementation of CTM and DTM in MS4 Me	376
19.4	Transient Behavior in Markov Models	377
19.5	DEVS Features for Markov Modeling	380
19.5.1	Individualizing Markov Models	380
19.5.2	Dynamic Structure Markov Models	380
19.5.3	Multi-aspects and Statistical Considerations	381
19.6	Case Studies	382
19.6.1	Markov Modeling of Populations with Multiple Distinguishing Characteristics	382
19.6.2	Speedup in Multiprocessor Computation	383
19.7	Summary	390
	Appendix 1 Exponential Distribution and Markov Model Basics	390
	References	394
Index	395