

---

# Theory of Modeling and Simulation

Integrating Discrete Event  
and Continuous Complex  
Dynamic Systems

*Second Edition*

**BERNARD P. ZEIGLER**

Electrical and Computer Engineering Department  
University of Arizona  
Tucson, Arizona

**HERBERT PRAEHOFER**

Institute of Systems Science  
Johannes Kepler University  
Linz, Austria

**TAG GON KIM**

Department of Electrical Engineering  
Korea Advanced Institute of Science and Technology  
Taejon, Korea



**ACADEMIC PRESS**

*An Imprint of Elsevier*

Amsterdam Boston Heidelberg London New York Oxford  
Paris San Diego San Francisco Singapore Sydney Tokyo

# Preface to the First Edition

## A Theory of Modeling and Simulation—Why?

The field of modeling and simulation is as diverse as the concerns of man. Every discipline has developed, or is developing, its own models and its own approach and tools for studying these models. Why then is an overall theory of modeling and simulation necessary?

Why should a college senior or first-year graduate student majoring in biology, aeronautical engineering or business management be introduced to the general theory of modeling and simulation—yet another discipline not specific to his own? The answer relies on the same reasoning that determined that the student should have acquired at least some grounding in mathematics. Nobody questions the role of arithmetic in the sciences, engineering, and management. Arithmetic is all pervasive, yet it is a mathematical discipline having its own axioms and logical structure. Its content is not specific to any other disciplines but is directly applicable to them all. Thus students of biology and engineering are not taught how to add differently—the different training comes in what to add, when to do it, and why.

The practice of modeling and simulation too is all pervasive. However, it has its own concepts of model description, simplification, validation, simulation, and exploration, which are not specific to any particular discipline. These statements would be agreed to by all. Not everyone, however, would say that the concepts named could be isolated and abstracted in a generally useful form.

In response, this book attempts to provide a framework in which the concepts can be sketched in their clear abstract forms; embodying the concepts in concrete modeling situations further serves to illuminate them. Of what use is such a framework for the specializing student, or the one who later will specialize? It can fulfill the following purposes:

1. Introduce the student to the full range of concepts and tools available in the field, rather than just those currently employed in his discipline.
2. Provide a perspective on the modeling activity of his own discipline, permitting the creative employment of insights and approaches stemming from other disciplines.
3. Provide an understanding of why we model and simulate, what can be achieved, and what cannot be achieved.
4. Provide a universal language for communicating the structure and behavior of models to others, regardless of whether they are conversant with the interpretive frameworks of their respective disciplines.

We have been talking about the theory of modeling and simulation relative to the noncomputer sciences. However, the book is also addressed to

those who are interested in general and mathematical systems theory or in the more specialized computer and information sciences. Indeed, the framework developed here reflects several related influences:

- General systems theory, with its belief in an underlying and exploitable unity in all systems, diverse though they may seem.
- Mathematical systems theory, with its attempts to formalize the concepts of system structure and behavior.
- Automata theory, with its formal, logical, and algebraic analysis of the behavior and interrelations of computer-related models.

In this context this book marshals the available approaches and formalisms in order to deal with the particular issues raised by modeling and simulation. Thus the theory presented should introduce the student of systems and computer science to the many open problems in the area needing conceptual and/or mathematical development, as well as furnish a basis for attacking them.

For the veteran practitioner of modeling and simulation, the book will fill an important gap in the literature between the material that is very abstract and that which is very applied. The reader who has had little formal mathematics may find the going rough. Still, my hope is that he will be led to more clearly perceive the issues faced daily in his modeling efforts.

B.P.Z.

Ann Arbor, 1976



# Preface to the Second Edition

This is the second edition of *Theory of Modeling and Simulation* originally published by Wiley Interscience in 1976 and reissued by Krieger Publishers in 1984. The first edition made the case that a theory was necessary to help bring some coherence and unity to the ubiquitous field of modeling and simulation. Although nearly a quarter of a century later there have been many advances in the field, we believe that a widely accepted framework and theoretical foundation are even more necessary today. Modeling and simulation lore is still fragmented across the disciplines making it difficult to share in the advances, reuse other discipline's ideas, and work collaboratively in multidisciplinary teams. As a consequence of the growing specialization of knowledge, there is even more fragmentation in the field now than ever. The need for "knowledge workers" who can synthesize disciplinary fragments into cohesive wholes is increasingly recognized. Modeling and simulation—as a generic, non-discipline-specific, set of activities—can provide a framework of concepts and tools for such knowledge work.

In the years since the first edition, there has been much significant progress in modeling and simulation, but the progress has not been uniform across the board. Generally, model building and simulation execution have been made easier and faster by riding piggyback on the technology advances in software (e.g., object-oriented programming) and hardware (e.g., faster processors). However, hard, fundamental issues such as model credibility (e.g., validation, verification, and model family consistency) and interoperability (e.g., repositories, reuse of components, and resolution matching) have received a lot less attention. But these issues are now moving to the front and center under the impetus of the High Level Architecture (HLA) standard mandated by the United States Department of Defense for all its contractors and agencies.

In this edition, two major contributors to the theory of modeling and simulation join with the original author to completely revise the original text. As suggested by its subtitle, the current book concentrates on the integration of the continuous and discrete paradigms for modeling and simulation. A second major theme is that of distributed simulation and its potential to support the coexistence of multiple formalisms in multiple model components.

Although the material is mostly new, the presentation format remains the same. There are three major sections. Part I introduces a framework for modeling and simulation and the primary continuous and discrete approaches to making models and simulating them on computers. This part offers a unified view of the field that most books lack and, written in an informal manner, it can be used as instructional material for undergraduate and graduate courses.

Part II revisits the introductory material but with a rigorous, multi-layered systems theoretic basis. It then goes on to provide an in-depth



account of models as systems specifications, the major systems specification formalisms and their integration, and simulators for such formalisms, in sequential, parallel, and distributed forms.

The fundamental role of systems morphisms is taken up in Part III—any claim relating systems, models and simulators to each other ultimately must be phrased with an equivalence or morphism of such kinds. Both perfect and approximate morphisms are discussed and applied to model abstraction and system representation. Especially, in the latter vein, we focus on the ability of the DEVS (Discrete Event System Specification) formalism to represent arbitrary systems, including those specified in other discrete event and continuous formalisms. The importance of this discussion derives from two sources—the burgeoning use of discrete event approaches in high technology design (e.g., manufacturing control systems, communication, computers) and the HLA-stimulated growth of distributed simulation, for which discrete events match the discreteness of message exchange.

Part IV continues with the theme of DEVS-based modeling and simulation as a foundation for a high-technology systems design methodology. We include integration with other formalisms for analysis and the system entity structure/model base concepts for investigating design alternatives and reusing good designs. Thoughts on future support of collaborative modeling and simulation close the book.

Although the book is primarily intended as a reference, its structure makes it appropriate for use as a textbook in graduate courses on modeling and simulation. As a textbook, the book affords the advantage of providing an open systems view that mitigates the closed-box trust-on-faith approach of many commercial domain-specific simulation packages. If nothing else, the student will have a more sophisticated skepticism about the model reliability and simulator correctness inside such boxes. For hands-on experience, the book needs to be supplemented with an instructional modeling and simulation environment such as DEVSJAVA (available from the Web site: [arizona.edu](http://arizona.edu)). Other books on statistical aspects of simulation and application to particular domains should be part of the background as well.

We suggest that Part IV might be a good place to start reading, or teaching, since most of the concepts developed earlier in the book are put into use in the last chapters. In this strategy, the learner soon realizes that new concepts are needed to achieve successful designs and is motivated to fill in the gaps by turning to the chapters that supply the requisite knowledge. More likely, a good teacher will guide the student back and forth between the later and earlier material.

Space limitations have prevented us from including all the material in the first edition. The decision on what to leave out was based on relevance to the current theme, whether significant progress had been made in an area, and whether this could be reduced to the requirements of a book. Thus, for example, a major omission is the discussion of structural inference in Chapters 14 and 15 of the original. We hope that a next revision would be able to include much more on developments in these important directions.

B.P.Z.  
H.P.  
T.G.K.



# Acknowledgments

The authors have, when combined, way too many mentors, colleagues, students, and sponsors to acknowledge individually. Our mentors have provided the right word at the right time, and their colleagues have collaborated to develop concepts and software that are beyond the individual capacities to achieve. Our graduate students have struggled with the theories and the recalcitrant software tools to implement prototypes that demonstrated key principles. And of course our sponsors have made it all possible, often with only faith as collateral. The relationship of the authors to each other illustrates the student-mediated intergenerational transmission process. One of the book's authors, Tag Gon Kim, now a full professor at KAIST (Korea Advanced Institute of Science and Technology), obtained his Ph.D. degree under supervision of the senior author (BPZ) at the University of Arizona. Herbert Praehofer, the third author, is now a member of the Systems Science Institute at the University of Linz, Austria, where he also obtained his Ph.D. degree under Franz Pichler, a colleague of Zeigler's. HP especially thanks Prof. Pichler for initiating his interest in systems theory and for the continuing promotion of his work.

Starting near the beginning, Tunçer Ören of Turkey and Canada, and Maurice Elzas of the Netherlands, initiated a series of meetings that were critical to the development of modeling and simulation methodology. Along the way Art Burks, John Holland, George Klir, Ghi Vansteenkiste, Richard Nance, Francois Cellier, Larry Ho, Andrew Sage, Alex Sisti, Paul Davis, and Judith Dahmann provided encouragement at critical turning points. Colleagues Roger Weinberg, Kyetu Ho Park, Paul Fishwick, George Ball, and Steve Hall played important roles in theory and software developments. Of all the students that have contributed important advances, Jerzy Rozenblit, Hessam Sarjoughian, and Adelinde Uhrmacher, now noted researchers in their own right, deserve special mention. Among the many other graduates that dragged us, sometimes hesitantly, in new directions, space allows us to mention David Belogus, Art Concepcion, Doo Kwan Baik, Joon Sung Hong, Ki Hyung Kim, Hyu Chan Park, Kyung Pyo Hong, Yound Jae Kim, Wan Bok Lee, Yi Won Kwon, Alex Chow, Maria Vasconcelos, Doo Hwan Kim, Yoonkeon Moon, Gernot Reisinger, Sung-Do Chi, Jerry Couretas, and Hyup Jae Cho. From his varied roles in the United States Department of Defense, Dell Lunceford was the mover and shaker behind the Advanced Simulation Technology Thrust, a DARPA initiative that funded advanced work on DEVS, some of which is described in Chapter 16.

Finally, we want to thank our parents, wives, and children for their love, patience, and understanding.

B.P.Z.  
H.P.  
T.G.K.

# Contents

Preface to First Edition .....	xvii
Preface to Second Edition .....	xix
Acknowledgments .....	xxi

## **Part I: Basics .....** 1

### **Chapter 1 Introduction to Systems Modeling Concepts .....** 3

1.1 Systems Specification Formalisms .....	3
1.1.1 Relation to Object Orientation .....	5
1.1.2 Evolution of Systems Formalisms .....	6
1.1.3 Combining Continuous and Discrete Formalisms .....	7
1.1.4 Quantized Systems .....	8
1.1.5 Extensions of DEVS .....	9
1.2 Levels of System Knowledge .....	11
1.3 Introduction to the Hierarchy of Systems Specifications .....	13
1.4 The Specification Levels Informally Presented .....	14
1.4.1 Observation Frame .....	14
1.4.2 I/O Behavior and I/O Function .....	16
1.4.3 State Transition System Specification .....	17
1.4.4 Coupled Component System Specification .....	18
1.5 System Specification Morphisms: Basic Concepts .....	18
1.6 Summary .....	21
1.7 Sources .....	22
Definitions, Acronyms, Abbreviations .....	23

### **Chapter 2 Framework for Modeling and Simulation .....** 25

2.1 The Entities of the Framework .....	25
2.1.1 Source System .....	25
2.1.2 Experimental Frame .....	27
2.1.3 Model .....	29
2.1.4 Simulator .....	30
2.2 Primary Relations Among Entities .....	30
2.2.1 Modeling Relation: Validity .....	30
2.2.2 Simulation Relation: Simulator Correctness .....	32
2.3 Other Important Relationships .....	32
2.3.1 Modeling As Valid Simplification .....	32
2.3.2 Experimental Frame: Model Relationships .....	33
2.4 Time .....	34
2.5 Summary .....	35
2.6 Sources .....	36



<b>Chapter 3</b>	<b>Modeling Formalisms and Their Simulators</b>	<b>37</b>
3.1	Introduction	37
3.2	Discrete Time Models and Their Simulators	37
3.2.1	Discrete Time Simulation	39
3.2.2	Cellular Automata	40
3.2.3	Cellular Automaton Simulation Algorithms	43
3.2.4	Discrete Event Approach to Cellular Automaton Simulation	44
3.2.5	Switching Automata/Sequential Machines	45
3.2.6	Linear Discrete Time Networks and Their State Behavior	47
3.3	Differential Equation Models and Their Simulators	49
3.3.1	Continuous System Simulation	50
3.3.2	Feedback in Continuous Systems	55
3.3.3	Elementary Linear Systems	56
3.3.4	Nonlinear Oscillators: Limit Cycles and Chaotic Behaviors	61
3.3.5	Continuous System Simulation Languages and Systems	64
3.4	Discrete Event Models and Their Simulators	66
3.4.1	Introduction	66
3.4.2	Discrete Event Cellular Automata	67
3.4.3	Discrete Event World Views	70
3.5	Summary	72
3.6	Sources	73
<b>Chapter 4</b>	<b>Introduction to Discrete Event System Specifications (DEVS)</b>	<b>75</b>
4.1	Introduction	75
4.2	Classic DEVS System Specification	75
4.2.1	DEVS Examples	77
4.2.2	Classic DEVS With Ports	84
4.2.3	Classic DEVS Coupled Models	85
4.3	Parallel DEVS System Specification	89
4.3.1	Processor With Buffer	90
4.3.2	Parallel DEVS Coupled Models	91
4.4	Hierarchical Models	93
4.5	Object-Oriented Implementations of DEVS: An Introduction	93
4.5.1	Structural Inheritance	94
4.6	Summary	96
4.7	Sources	96
<b>Chapter 5</b>	<b>Hierarchy of System Specifications</b>	<b>99</b>
5.1	Time Base	99
5.2	Segments and Trajectories	100
5.2.1	Piecewise Continuous Segments	102
5.2.2	Piecewise Constant Segments	103
5.2.3	Event Segments	103
5.2.4	Sequences	104



5.3	I/O Observation Frame .....	104
5.4	I/O Relation Observation .....	105
5.5	I/O Function Observation .....	107
5.6	I/O System .....	108
5.6.1	Going from System Structure to Behavior .....	110
5.6.2	Time-Invariant Systems .....	112
5.6.3	Special Cases: Input-Free and Memoryless Systems .....	115
5.7	Iterative Specification of Systems .....	116
5.7.1	Generator Segments .....	117
5.7.2	Generator State Transition Systems .....	120
5.8	Multivariable Sets and Structured Systems .....	123
5.9	Multicomponent System Specification .....	125
5.10	Network of System Specifications (Coupled Systems) .....	127
5.10.1	Coupled System Specification .....	128
5.10.2	Coupled System Specification at the Structured System Level .....	129
5.11	Summary .....	131
5.12	Sources .....	133

## **Part II Modeling Formalisms and Simulation Algorithms ..... 135**

### **Chapter 6 Basic Formalisms: DEVS, DTSS, DESS ..... 137**

6.1	Basic System Specification Formalisms .....	137
6.2	Discrete Event System Specifications (DEVS) .....	138
6.2.1	Classic DEVS .....	138
6.2.2	Structure Specified by DEVS .....	139
6.2.3	Legitimacy: When is the Structure Specified by a DEVS Really a System? .....	141
6.3	Parallel DEVS .....	142
6.3.1	System Specified by Parallel DEVS .....	144
6.4	Discrete Time System Specification (DTSS) .....	144
6.5	Differential Equation System Specification (DESS) .....	146
6.6	Summary .....	147

### **Chapter 7 Basic Formalisms: Coupled Multicomponent Systems ..... 149**

7.1	Discrete Event Specified Network Formalism .....	149
7.1.1	Classic DEVS Coupled Models .....	150
7.1.2	Parallel DEVS Coupled Models .....	152
7.2	Multicomponent Discrete Event System Formalism .....	155
7.2.1	Event Scheduling Models .....	159
7.2.2	Combined Event Scheduling, Activity Scanning Simulation Strategy .....	159
7.2.3	Process Interaction Models .....	161
7.2.4	Translating Nonmodular Multicomponent DEVS Models into Modular Form .....	161
7.2.5	State Updating in Distributed Simulation .....	162

7.3	Discrete Time Specified Network Formalism .....	163
7.4	Multicomponent Discrete Time System Formalism .....	166
7.5	Differential Equation Specified Network Formalism .....	168
7.6	Multicomponent Differential Equations Specified System Formalism .....	169
7.7	Summary .....	171
7.8	Sources .....	172
	Appendix .....	173
<b>Chapter 8</b>	<b>Simulators for Basic Formalisms .....</b>	<b>175</b>
8.1	Simulators for DEVS .....	176
8.1.1	Simulator for Basic DEVS .....	177
8.1.2	Simulators for Modular DEVS Networks .....	180
8.1.3	The Root-Coordinator .....	184
8.2	DEVS Bus .....	184
8.2.1	Simulator for Event Scheduling Multicomponent DEVS .....	185
8.2.2	Simulator for Activity Scanning and Process Interaction Multicomponent DEVS .....	187
8.3	Simulators for DTSS .....	189
8.3.1	Simulator for Atomic DTSS .....	190
8.3.2	Simulator for Instantaneous Functions .....	192
8.3.3	Simulator for Nonmodular Multicomponent DTSS .....	192
8.3.4	Simulators for Coupled DTSS .....	193
8.3.5	The Root-Coordinator .....	196
8.4	Simulators for DESS .....	197
8.4.1	Causal Simulator for DESS .....	197
8.4.2	Noncausal Simulator for DESS .....	198
8.5	Summary .....	201
8.6	Sources .....	201
<b>Chapter 9</b>	<b>Multiformalism Modeling and Simulation .....</b>	<b>203</b>
9.1	Brief Introduction to Specialized Formalisms .....	203
9.1.1	DESS Subformalisms: Systems Dynamics and Bond Graphs .....	203
9.1.2	DEVS Subformalisms: Petri Nets and Statecharts .....	204
9.2	Multiformalism Modeling .....	206
9.3	DEV&DESS: Combined Discrete Event and Differential Equation Specified Systems .....	208
9.3.1	A Simple Example: DEV&DESS Model of a Barrel Filler .....	210
9.3.2	System Specified by a DEV&DESS .....	213
9.4	Multimodeling with DEV&DESS .....	214
9.4.1	Example: Pot System with Command Inputs and Threshold Value Outputs .....	216
9.5	Coupled DEV&DESS: Network of Multiformalism Models .....	217
9.5.1	Basic Formalisms are Subsumed by DEV&DESS .....	217
9.5.2	Coupled DEV&DESS Formalism .....	219



9.6	Simulator for DEVS&DESS .....	222
9.6.1	The DEV&DESS-Simulator and-Coordinator .....	223
9.6.2	Integrating Different Modeling Formalisms .....	226
9.7	Sources .....	227
	Appendix A: The System Specified By a DEV&DESS .....	229
	Appendix B: The System Specified By a Multiformalism System–Closure Under Coupling of Networks of DEV&DESS .....	231
<b>Chapter 10</b>	<b>DEVS-Based Extended Formalisms .....</b>	<b>233</b>
10.1	Stochastic Systems: Conventional Approach to Uncertainty .....	233
10.2	DSDEVS (Dynamic Structure DEVS) .....	235
10.2.1	DSDEVS Closure Under Coupling .....	237
10.2.2	Example: Adaptive Processing Architectures .....	237
10.3	Symbolic DEVS .....	240
10.3.1	Symbolic DEVS Coupled Models .....	241
10.3.2	Example: Performance Evaluation of Basic Architectures .....	241
10.4	Fuzzy DEVS .....	244
10.4.1	Basic Fuzzy DEVS Formalism .....	247
10.4.2	Embedding Fuzzy DEVS in DEVS .....	250
10.4.3	Example: Fuzzy DEVS Model for Boiler System .....	251
10.5	Real-Time DEVS (RT-DEVS) .....	252
10.5.1	Formal Specification of Real-Time DEVS Models .....	252
10.5.2	Execution of RT-DEVS Models .....	253
10.5.3	Example of RT-DEVS: Elevator Control .....	254
10.5.4	Real-Time Simulation and RT-DEVS .....	257
10.6	Summary .....	259
10.7	Sources .....	259
<b>Chapter 11</b>	<b>Parallel and Distributed Discrete Event Simulation .....</b>	<b>261</b>
11.1	Problem Characterization of Parallel Discrete Event Simulation .....	262
11.2	Conservative Parallel Discrete Event Simulation .....	264
11.2.1	Conservative Parallel DEVS Simulator .....	268
11.3	Optimistic Parallel Discrete Event Simulation .....	273
11.3.1	Time-Warp DEVS Simulator .....	275
11.3.2	Riskfree Optimistic DEVS Simulator .....	281
11.4	Parallel DEVS Simulator .....	284
11.5	Summary .....	287
11.6	Sources .....	287
<b>Part III</b>	<b>System Morphisms: Abstraction, Representation, Approximation .....</b>	<b>293</b>
<b>Chapter 12</b>	<b>Hierarchy of System Morphisms .....</b>	<b>295</b>
12.1	The I/O Frame Morphism .....	297
12.2	The I/O Relation Observation Morphism .....	297

12.3	The I/O Function Morphism .....	298
12.3.1	IOFO System Morphism Implies IORO System Morphism .....	300
12.4	The I/O System Morphism .....	301
12.4.1	I/O System Morphism Implies IOFO and IORO Morphism .....	302
12.4.2	The Lattice of Partitions and the Reduced Version of a System .....	305
12.5	System Morphism for Iteratively Specified Systems .....	308
12.5.1	Iterative Specification Morphism Implies I/O System Morphism .....	309
12.5.2	Specialization of Morphisms for Iteratively Specified Systems .....	310
12.6	The Structured System Morphism .....	311
12.7	Multicomponent System Morphism .....	314
12.8	The Network of Systems Morphism .....	317
12.9	Homomorphism and Cascade Decompositions .....	320
12.10	Characterization of Realizable I/O Relations and Functions .....	324
12.10.1	Canonical Realization .....	326
12.11	Summary .....	327
12.12	Sources .....	327
<b>Chapter 13</b>	<b>Abstraction: Constructing Model Families .....</b>	<b>329</b>
13.1	Scope/Resolution/Interaction Product .....	329
13.1.1	Complexity .....	330
13.1.2	Size/Resolution Trade-Off: Simplification Methods .....	332
13.1.3	How Objectives and Experimental Frame Determine Abstraction Possibilities .....	334
13.2	Integrated Families of Models .....	335
13.2.1	Integrated Model Family Example: Space Travel .....	335
13.2.2	Space Travel Base Model .....	336
13.3	Aggregation: Homogeneity/Coupling Indifference Principles ...	339
13.3.1	Coupling Conditions Imposed by Anonymity .....	342
13.3.2	Constructing Lumped Models Based on Identity-Erasing Aggregation .....	344
13.3.3	All-To-One Coupling .....	348
13.3.4	Example of Aggregation Model Construction: Space Travel .....	349
13.3.5	Constructing Aggregations Through State and Block Refinement .....	351
13.3.6	Time Scale Relations .....	353
13.3.7	Universality/Applicability of Identity-Erasing Aggregations .....	353
13.4	Abstractions for Event-Based Control .....	354
13.4.1	Boundary-Based DEVS .....	354
13.4.2	DEVS Abstraction: Space Travel Example .....	357
13.5	Parameter Morphisms .....	358
13.5.1	Linear Systems Parameter Morphisms .....	359



13.5.2	Example Lumpable: Linear DTSS and Parameter Morphisms .....	360
13.5.3	Constraints on Parameter and State Spaces: Disaggregation .....	362
13.5.4	Using Parameter Morphisms in an Integrated Model Family .....	362
13.6	Summary .....	364
13.7	Sources .....	364
<b>Chapter 14</b>	<b>Verification, Validation, Approximate Morphisms: Living with Error . . .</b>	<b>367</b>
14.1	Verification .....	367
14.2	Validation at the Behavioral Level .....	368
14.2.1	Quantitative Comparison .....	371
14.2.2	Qualitative Comparison .....	372
14.3	Performance/Validity (e.g., Speed/Accuracy) Trade-off .....	373
14.4	Approximate Morphisms and Error Behavior .....	377
14.4.1	Approximate Morphisms and the Specification Hierarchy .....	377
14.4.2	Approximate Homomorphisms and Error Propagation .....	378
14.4.3	Example: Approximate Linear System Homomorphisms .....	382
14.5	Approximate Morphisms at the Coupled System Level .....	384
14.5.1	Error-Driven Aggregation Refinement .....	384
14.6	Validation at Structural Levels .....	387
14.6.1	Calibration, Parameter Identification, Sensitivity .....	388
14.6.2	Local/Global, Cross-Model Validation .....	388
14.7	Sources .....	389
<b>Chapter 15</b>	<b>DEVS and DEVS-like Systems: Universality and Uniqueness .....</b>	<b>391</b>
15.1	Relation Between Classical and Parallel DEVS: Is There One or Two? .....	391
15.2	Universality and Uniqueness of DEVS .....	393
15.2.1	Systems with DEVS Interfaces .....	393
15.2.2	Behavior of DEVS-like Systems .....	395
15.2.3	Universality of DEVS .....	396
15.2.4	Example: DEVS Realization of DEVS-like System .....	397
15.2.5	Uniqueness of DEVS .....	398
15.3	DEVS Representation of DTSS .....	399
15.3.1	Input-Free Moore .....	400
15.3.2	Multiported FNSS .....	400
15.3.3	Moore DTSS with Input .....	401
15.3.4	Mealy DTSS .....	401
15.3.5	DEVS Strong Simulation of DTSS Coupled Models .....	401
15.4	Efficient DEVS Simulation of DTSS Networks .....	403
15.5	Summary .....	405
15.6	Sources .....	405
	Appendix: Isomorphically Representing DEVS-like Systems by DEVS .....	406

<b>Chapter 16</b>	<b>DEVS Representation of Systems</b> .....	<b>411</b>
16.1	DEVS Bus Revisited .....	411
16.1.1	Approaches to DEVS Representation of Continuous Systems .....	412
16.2	DEVS Representation Using Conventional Approach .....	414
16.2.1	DTSS Simulation of a DESS Integrator .....	415
16.2.2	Simulation of Coupled Systems by DTSS .....	416
16.2.3	Discretized Simulation of Coupled DESS with Arbitrarily Small Error .....	418
16.2.4	DEVS Representation of DESS via DTSS Simulation .....	419
16.3	Quantization: An Alternative Approach for DEVS Representation .....	419
16.3.1	Quantized Systems .....	421
16.3.2	Exactly Quantizable Systems .....	422
16.3.3	Quantized Integrator: Approximation of DESS .....	424
16.3.4	Coupled Systems with Quantized Components .....	425
16.3.5	Quantized Simulation of Coupled Systems with Arbitrarily Small Error .....	427
16.3.6	Quantized Simulation of Coupled DESS with Arbitrarily Small Error .....	429
16.3.7	DEVS Representation of Quantized Systems .....	430
16.3.8	DEVS Representation of Quantized Integrator .....	431
16.3.9	DEVS Simulation of Coupled Quantized Systems .....	432
16.3.10	Quantization-Based DEVS Simulation of DESS .....	433
16.4	Simulation Study of Quantization .....	434
16.4.1	Some Indicative Simulation Results .....	434
16.4.2	Comparing Quantized DEVS with Pure DTSS Simulation of DESS .....	437
16.4.3	Insight from Second-Order Linear Oscillator .....	439
16.5	Conjectures for Further Research .....	441
16.6	Summary .....	443
16.7	Sources .....	444
16.8	Problems .....	445
	Appendix 1 Closed Loop DTSS Simulation .....	447
	Appendix 2 Uniformly Segmentable Input Sets .....	449
	Appendix 3 Exact Simulation by DEVS .....	451
	Appendix 4 Closed Loop Quantized Simulation .....	452
<b>Part IV</b>	<b>System Design and Modeling and Simulation Environments</b> .....	<b>455</b>
<b>Chapter 17</b>	<b>DEVS-Based Design Methodology</b> .....	<b>457</b>
17.1	Methodology Overview .....	457
17.2	DEVS Definition Language .....	459
17.2.1	Atomic DEVS Definition .....	460
17.2.2	The Coupled DEVS Definition .....	462
17.2.3	Example: Generator-Buffer-Processor Model .....	463



17.3	Execution of DEVS Definition .....	466
17.4	Logical Analysis: Model Verification .....	468
17.4.1	Assertional Specification of Temporal Logic .....	468
17.4.2	TL Specification Language .....	473
17.5	Model Verification by Language Acceptance Checking .....	475
17.6	Performance Evaluation .....	477
17.7	Implementation: DEVS Model Execution .....	477
17.8	Summary .....	479
17.9	Sources .....	479
<b>Chapter 18</b>	<b>System Entity Structure/Model Base Framework .....</b>	<b>481</b>
18.1	Model Base Management by System Entity Structure .....	481
18.2	System Entity Structure .....	482
18.3	System Entity Structure/Model Base (SES/MB) Framework .....	485
18.4	Example: Design of a Transaction Processing System .....	486
18.4.1	System Entity Structure .....	487
18.4.2	Model Base .....	487
18.4.3	Pruning and Model Synthesis .....	487
18.4.4	Performance Evaluation .....	488
18.5	Automatic Pruning of an SES .....	490
18.6	Summary .....	491
18.7	Sources .....	492
<b>Chapter 19</b>	<b>Collaboration and the Future .....</b>	<b>495</b>
19.1	An Architecture for M&S .....	495
19.2	How Does Collaboration Support M&S? .....	497
19.2.1	Model Construction .....	497
19.2.2	Model Composition .....	497
19.2.3	Supporting M&S with Collaboration Environments .....	497
19.3	Summary .....	499
19.4	Sources .....	499
<b>Index</b>	.....	<b>501</b>



# Basics



