

Contents

<i>Preface</i>	<i>page xv</i>
<i>Annotated Table of Contents</i>	xix
<i>Acknowledgments</i>	xxv
PART I Overview	
1 Introduction	3
1.1 Software Modeling	3
1.2 Object-Oriented Methods and the Unified Modeling Language	3
1.3 Software Architectural Design	5
1.4 Method and Notation	5
1.5 COMET: A UML-Based Software Modeling and Design Method for Software Applications	6
1.6 UML as a Standard	6
1.7 Multiple Views of Software Architecture	7
1.8 Evolution of Software Modeling and Design Methods	8
1.9 Evolution of Object-Oriented Analysis and Design Methods	9
1.10 Survey of Concurrent, Distributed, and Real-Time Design Methods	11
1.11 Summary	12
Exercises	12
2 Overview of the UML Notation	14
2.1 UML Diagrams	14
2.2 Use Case Diagrams	15
2.3 Classes and Objects	15
2.4 Class Diagrams	16
2.5 Interaction Diagrams	18
2.6 State Machine Diagrams	19
2.7 Packages	21

viii **Contents**

2.8	Concurrent Communication Diagrams	21
2.9	Deployment Diagrams	23
2.10	UML Extension Mechanisms	23
2.11	Conventions Used in This Book	25
2.12	Summary	27
	Exercises	28
3	Software Life Cycle Models and Processes	29
3.1	Software Life Cycle Models	29
3.2	Design Verification and Validation	40
3.3	Software Life Cycle Activities	41
3.4	Software Testing	42
3.5	Summary	43
	Exercises	43
4	Software Design and Architecture Concepts	45
4.1	Object-Oriented Concepts	45
4.2	Information Hiding	48
4.3	Inheritance and Generalization/Specialization	51
4.4	Concurrent Processing	53
4.5	Design Patterns	57
4.6	Software Architecture and Components	58
4.7	Software Quality Attributes	59
4.8	Summary	59
	Exercises	60
5	Overview of Software Modeling and Design Method	61
5.1	COMET Use Case-Based Software Life Cycle	61
5.2	Comparison of the COMET Life Cycle with Other Software Processes	64
5.3	Requirements, Analysis, and Design Modeling	65
5.4	Designing Software Architectures	67
5.5	Summary	68
	Exercises	68
PART II Software Modeling		
6	Use Case Modeling	71
6.1	Requirements Modeling	72
6.2	Use Cases	74
6.3	Actors	76
6.4	Identifying Use Cases	78
6.5	Documenting Use Cases in the Use Case Model	80
6.6	Example of Use Case Description	80
6.7	Use Case Relationships	82
6.8	The Include Relationship	82
6.9	The Extend Relationship	85
6.10	Use Case Structuring Guidelines	88

	Contents	ix
6.11	Specifying Nonfunctional Requirements	89
6.12	Use Case Packages	89
6.13	Activity Diagrams	89
6.14	Summary	92
	Exercises	92
7	Static Modeling	94
7.1	Associations between Classes	95
7.2	Composition and Aggregation Hierarchies	100
7.3	Generalization/Specialization Hierarchy	102
7.4	Constraints	103
7.5	Static Modeling and the UML	103
7.6	Static Modeling of the System Context	104
7.7	Categorization of Classes Using UML Stereotypes	106
7.8	Modeling External Classes	107
7.9	Static Modeling of Entity Classes	111
7.10	Summary	113
	Exercises	114
8	Object and Class Structuring	115
8.1	Object and Class Structuring Criteria	116
8.2	Modeling Application Classes and Objects	116
8.3	Object and Class Structuring Categories	117
8.4	External Classes and Software Boundary Classes	118
8.5	Boundary Classes and Objects	119
8.6	Entity Classes and Objects	123
8.7	Control Classes and Objects	124
8.8	Application Logic Classes and Objects	127
8.9	Summary	130
	Exercises	130
9	Dynamic Interaction Modeling	132
9.1	Object Interaction Modeling	133
9.2	Message Sequence Numbering on Interaction Diagrams	136
9.3	Dynamic Interaction Modeling	139
9.4	Stateless Dynamic Interaction Modeling	139
9.5	Examples of Stateless Dynamic Interaction Modeling	140
9.6	Summary	148
	Exercises	148
10	Finite State Machines	151
10.1	Finite State Machines and State Transitions	151
10.2	Examples of Statecharts	153
10.3	Events and Guard Conditions	157
10.4	Actions	158
10.5	Hierarchical Statecharts	163
10.6	Guidelines for Developing Statecharts	167

x	Contents	
	10.7	Developing Statecharts from Use Cases 168
	10.8	Example of Developing a Statechart from a Use Case 169
	10.9	Summary 175
		Exercises 175
	11	State-Dependent Dynamic Interaction Modeling 177
	11.1	Steps in State-Dependent Dynamic Interaction Modeling 177
	11.2	Modeling Interaction Scenarios Using Interaction Diagrams and Statecharts 178
	11.3	Example of State-Dependent Dynamic Interaction Modeling: Banking System 179
	11.4	Summary 187
		Exercises 188
	PART III	Architectural Design
	12	Overview of Software Architecture 193
	12.1	Software Architecture and Component-Based Software Architecture 193
	12.2	Multiple Views of a Software Architecture 194
	12.3	Software Architectural Patterns 198
	12.4	Documenting Software Architectural Patterns 205
	12.5	Interface Design 206
	12.6	Designing Software Architectures 207
	12.7	Summary 209
		Exercises 210
	13	Software Subsystem Architectural Design 212
	13.1	Issues in Software Architectural Design 212
	13.2	Integrated Communication Diagrams 213
	13.3	Separation of Concerns in Subsystem Design 216
	13.4	Subsystem Structuring Criteria 220
	13.5	Decisions about Message Communication between Subsystems 226
	13.6	Summary 228
		Exercises 228
	14	Designing Object-Oriented Software Architectures 230
	14.1	Concepts, Architectures, and Patterns 231
	14.2	Designing Information Hiding Classes 231
	14.3	Designing Class Interface and Operations 232
	14.4	Data Abstraction Classes 234
	14.5	State-Machine Classes 236
	14.6	Graphical User Interaction Classes 237
	14.7	Business Logic Classes 239
	14.8	Inheritance in Design 239
	14.9	Class Interface Specifications 245
	14.10	Detailed Design of Information Hiding Classes 246

	Contents	xi
14.11 Polymorphism and Dynamic Binding	248	
14.12 Implementation of Classes in Java	249	
14.13 Summary	250	
Exercises	251	
15 Designing Client/Server Software Architectures	253	
15.1 Concepts, Architectures, and Patterns for Client/Server Architectures	254	
15.2 Client/Service Software Architectural Structure Patterns	254	
15.3 Architectural Communication Patterns for Client/Server Architectures	258	
15.4 Middleware in Client/Server Systems	260	
15.5 Design of Service Subsystems	261	
15.6 Design of Wrapper Classes	266	
15.7 From Static Models to Relational Database Design	268	
15.8 Summary	275	
Exercises	276	
16 Designing Service-Oriented Architectures	278	
16.1 Concepts, Architectures, and Patterns for Service-Oriented Architecture	279	
16.2 Software Architectural Broker Patterns	280	
16.3 Technology Support for Service-Oriented Architecture	283	
16.4 Software Architectural Transaction Patterns	285	
16.5 Negotiation Pattern	289	
16.6 Service Interface Design in Service-Oriented Architecture	292	
16.7 Service Coordination in Service-Oriented Architecture	294	
16.8 Designing Service-Oriented Architectures	295	
16.9 Service Reuse	297	
16.10 Summary	298	
Exercises	298	
17 Designing Component-Based Software Architectures	300	
17.1 Concepts, Architectures, and Patterns for Component-Based Software Architectures	300	
17.2 Designing Distributed Component-Based Software Architectures	301	
17.3 Composite Subsystems and Components	302	
17.4 Modeling Components with UML	303	
17.5 Component Structuring Criteria	307	
17.6 Group Message Communication Patterns	310	
17.7 Application Deployment	314	
17.8 Summary	316	
Exercises	316	
18 Designing Concurrent and Real-Time Software Architectures	318	
18.1 Concepts, Architectures, and Patterns for Concurrent and Real-Time Software Architectures	318	

xii **Contents**

18.2	Characteristics of Real-Time Systems	319
18.3	Control Patterns for Real-Time Software Architectures	320
18.4	Concurrent Task Structuring	322
18.5	I/O Task Structuring Criteria	323
18.6	Internal Task Structuring Criteria	327
18.7	Developing the Concurrent Task Architecture	331
18.8	Task Communication and Synchronization	332
18.9	Task Interface and Task Behavior Specifications	338
18.10	Implementation of Concurrent Tasks in Java	342
18.11	Summary	342
	Exercises	343
19	Designing Software Product Line Architectures	344
19.1	Evolutionary Software Product Line Engineering	344
19.2	Requirements Modeling for Software Product Lines	345
19.3	Analysis Modeling for Software Product Lines	349
19.4	Dynamic State Machine Modeling for Software Product Lines	352
19.5	Design Modeling for Software Product Lines	353
19.6	Summary	355
	Exercises	355
20	Software Quality Attributes	357
20.1	Maintainability	357
20.2	Modifiability	358
20.3	Testability	360
20.4	Traceability	360
20.5	Scalability	361
20.6	Reusability	363
20.7	Performance	364
20.8	Security	365
20.9	Availability	366
20.10	Summary	367
	Exercises	367
PART IV Case Studies		
21	Client/Server Software Architecture Case Study	371
21.1	Problem Description	371
21.2	Use Case Model	372
21.3	Static Modeling	376
21.4	Object Structuring	381
21.5	Dynamic Modeling	384
21.6	ATM Statechart	396
21.7	Design of Banking System	401
21.8	Integrating the Communication Model	401
21.9	Structuring the System into Subsystems	403
21.10	Design of ATM Client Subsystem	404

	Contents	xiii
21.11 Design of Banking Service Subsystem	410	
21.12 Relational Database Design	415	
21.13 Deployment of Banking System	417	
21.14 Alternative Design Considerations	419	
21.15 Detailed Design	419	
22 Service-Oriented Architecture Case Study	424	
22.1 Problem Description	424	
22.2 Use Case Modeling	425	
22.3 Static Modeling	430	
22.4 Object and Class Structuring	433	
22.5 Dynamic Modeling	434	
22.6 Broker and Wrapper Technology Support for Service-Oriented Architecture	440	
22.7 Design Modeling	440	
22.8 Service Reuse	451	
23 Component-Based Software Architecture Case Study	453	
23.1 Problem Description	453	
23.2 Use Case Modeling	453	
23.3 Static Modeling	456	
23.4 Dynamic Modeling	457	
23.5 Design Modeling	462	
23.6 Software Component Deployment	471	
24 Real-Time Software Architecture Case Study	472	
24.1 Problem Description	472	
24.2 Use Case Modeling	473	
24.3 Static Modeling	474	
24.4 Object and Class Structuring	476	
24.5 Dynamic State Machine Modeling	476	
24.6 Dynamic Interaction Modeling	478	
24.7 Design Modeling	482	
Appendix A: Catalog of Software Architectural Patterns	495	
Appendix B: Teaching Considerations	521	
<i>Glossary</i>	523	
<i>Answers to Exercises</i>	537	
<i>Bibliography</i>	539	
<i>Index</i>	547	