



Designing Concurrent, Distributed, and Real-Time Applications with UML

Hassan Gomaa

George Mason University



ADDISON-WESLEY

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City



Contents

Foreword by Peter Freeman xv
Foreword by Bran Selic xvii
Preface xix

**PART I UML NOTATION, DESIGN CONCEPTS,
TECHNOLOGY, LIFE CYCLES, AND METHODS 1**

1 Introduction 3

 1.1 Object-Oriented Methods
 and the Unified Modeling Language 4

 1.2 Method and Notation 5

 1.3 Concurrent Applications. 6

 1.4 Real-Time Systems and Applications 8

 1.5 Distributed Systems and Applications 10

 1.6 Summary. 11

2 Overview of UML Notation. 13

 2.1 UML Diagrams. 13

 2.2 Use Case Diagrams 14

 2.3 UML Notation for Classes and Objects 14

 2.4 Class Diagrams. 15

 2.5 Interaction Diagrams. 17

 2.6 Statechart Diagrams 19

 2.7 Packages 20

 2.8 Concurrent Collaboration Diagrams. 21

2.9	Deployment Diagrams	23
2.10	UML Extension Mechanisms	24
2.11	The UML as a Standard	25
2.12	Summary	26
3	Software Design and Architecture Concepts	27
3.1	Object-Oriented Concepts	27
3.2	Information Hiding	30
3.3	Inheritance	36
3.4	Active and Passive Objects	37
3.5	Concurrent Processing	38
3.6	Cooperation between Concurrent Tasks	40
3.7	Information Hiding Applied to Access Synchronization	49
3.8	Monitors	51
3.9	Design Patterns	53
3.10	Software Architecture and Component-Based Systems	55
3.11	Summary	56
4	Concurrent and Distributed System Technology	57
4.1	Environments for Concurrent Processing	57
4.2	Runtime Support for Multiprogramming and Multiprocessing Environments	60
4.3	Task Scheduling	63
4.4	Operating System Input/Output Considerations	65
4.5	Client/Server and Distributed System Technology	68
4.6	World Wide Web Technology	73
4.7	Distributed Operating System Services	75
4.8	Middleware	78
4.9	Common Object Request Broker Architecture (CORBA)	81
4.10	Other Component Technologies	85
4.11	Transaction Processing Systems	86
4.12	Summary	88
5	Software Life Cycles and Methods	91
5.1	Software Life Cycle Approaches	91
5.2	Design Verification and Validation	98
5.3	Software Testing	99
5.4	Evolution of Software Design Methods	101

5.5	Evolution of Object-Oriented Analysis and Design Methods	103
5.6	Survey of Concurrent and Real-Time Design Methods ...	105
5.7	Summary	106

PART II	COMET: CONCURRENT OBJECT MODELING AND ARCHITECTURAL DESIGN WITH UML	107
6	Overview of COMET	109
6.1	COMET Object-Oriented Software Life Cycle	109
6.2	Comparison of the COMET Life Cycle with Other Software Processes	112
6.3	Requirements, Analysis, and Design Models	113
6.4	The COMET in a Nutshell	115
6.5	Summary	118
7	Use Case Modeling	119
7.1	Use Cases	119
7.2	Actors	120
7.3	Actors, Roles, and Users	123
7.4	Identifying Use Cases	123
7.5	Documenting Use Cases in the Use Case Model	124
7.6	Examples of Use Cases	125
7.7	Use Case Relationships	130
7.8	Use Case Packages	134
7.9	Summary	135
8	Static Modeling	137
8.1	Associations between Classes	137
8.2	Composition and Aggregation Hierarchies	145
8.3	Generalization/Specialization Hierarchy	147
8.4	Constraints	149
8.5	Static Modeling and the UML	149
8.6	Static Modeling of the System Context	152
8.7	Static Modeling of Entity Classes	155
8.8	Summary	157
9	Object and Class Structuring	159
9.1	Object Structuring Criteria	160
9.2	Categorization of Application Classes	160

9.3	Object Structuring Categories	162
9.4	External Classes and Interface Classes	163
9.5	Interface Objects	164
9.6	Entity Objects	170
9.7	Control Objects	172
9.8	Application Logic Objects	175
9.9	Subsystems	177
9.10	Summary	179
10	Finite State Machines and Statecharts	181
10.1	Finite State Machines	182
10.2	Events and States	182
10.3	Finite State Machines and Objects	183
10.4	Examples of Statecharts	183
10.5	Events and Conditions	187
10.6	Actions	190
10.7	Modeling Different Aspects of the System	196
10.8	Hierarchical Statecharts	198
10.9	Concurrent Statecharts	201
10.10	Guidelines for Developing Statecharts	203
10.11	Developing Statecharts from Use Cases	204
10.12	Example of Developing a Statechart from a Use Case. . .	205
10.13	Summary	212
11	Dynamic Modeling	215
11.1	Object Interaction Modeling	216
11.2	Message Labels on Interaction Diagrams	220
11.3	Dynamic Analysis	224
11.4	Non-State-Dependent Dynamic Analysis	224
11.5	Example of Non-State-Dependent Dynamic Analysis . .	226
11.6	State-Dependent Dynamic Analysis	227
11.7	Example of State-Dependent Dynamic Analysis: Banking System	230
11.8	Example of State-Dependent Dynamic Analysis: Cruise Control System	240
11.9	Summary	250

12	Software Architecture Design	253
12.1	Software Architectural Styles	253
12.2	System Decomposition Issues	257
12.3	Guidelines for Determining Subsystems	259
12.4	Consolidated Collaboration Diagrams	260
12.5	Subsystem Software Architecture	261
12.6	Separation of Concerns in Subsystem Design	261
12.7	Subsystem Structuring Criteria	265
12.8	Examples of Subsystem Decomposition	269
12.9	Static Modeling at the Design Level	270
12.10	Summary	274
13	Architectural Design of Distributed Applications	275
13.1	Configurable Architectures and Software Components	276
13.2	Steps in Designing Distributed Applications	276
13.3	System Decomposition	277
13.4	Designing Subsystem Interfaces	283
13.5	Transaction Management	292
13.6	Design of Server Subsystems	295
13.7	Distribution of Data	300
13.8	System Configuration	301
13.9	Summary	303
14	Task Structuring	305
14.1	Concurrent Task Structuring Issues	306
14.2	Task Structuring Categories	307
14.3	I/O Task Structuring Criteria	308
14.4	Internal Task Structuring Criteria	317
14.5	Task Priority Criteria	324
14.6	Task Clustering Criteria	325
14.7	Design Restructuring by Using Task Inversion	337
14.8	Developing the Task Architecture	341
14.9	Task Communication and Synchronization	345
14.10	Task Behavior Specifications	354
14.11	Summary	359

15	Class Design	361
15.1	Designing Information Hiding Classes.....	361
15.2	Designing Class Operations.....	362
15.3	Data Abstraction Classes.....	367
15.4	Device Interface Classes.....	369
15.5	State-Dependent Classes.....	375
15.6	Algorithm Hiding Classes.....	378
15.7	User Interface Classes.....	378
15.8	Business Logic Classes.....	381
15.9	Database Wrapper Classes.....	383
15.10	Software Decision Classes.....	384
15.11	Inheritance in Design.....	386
15.12	Examples of Inheritance.....	387
15.13	Class Interface Specifications.....	393
15.14	Summary.....	396
16	Detailed Software Design	397
16.1	Design of Composite Tasks.....	397
16.2	Synchronization of Access to Classes.....	405
16.3	Designing Connectors for Inter-Task Communication.....	414
16.4	Task Event Sequencing Logic.....	420
16.5	Summary.....	421
17	Performance Analysis of Concurrent	
	Real-Time Software Designs	423
17.1	Real-Time Scheduling Theory.....	423
17.2	Advanced Real-Time Scheduling Theory.....	431
17.3	Performance Analysis Using Event Sequence Analysis.....	436
17.4	Performance Analysis Using Real-Time Scheduling Theory and Event Sequence Analysis.....	437
17.5	Example of Performance Analysis Using Event Sequence Analysis.....	438
17.6	Example of Performance Analysis Using Real-Time Scheduling Theory.....	442
17.7	Example of Performance Analysis Using Real-Time Scheduling Theory and Event Sequence Analysis.....	444
17.8	Design Restructuring.....	452
17.9	Estimation and Measurement of Performance Parameters.....	453
17.10	Summary.....	454

PART III CASE STUDIES IN CONCURRENT, DISTRIBUTED, AND REAL-TIME APPLICATION DESIGN	457
18 Elevator Control System Case Study	459
18.1 Problem Description	459
18.2 Use Case Model	460
18.3 Static Model of the Problem Domain	465
18.4 Object Structuring	466
18.5 Dynamic Model	467
18.6 Statechart Model	476
18.7 Consolidation of Collaboration Diagrams	480
18.8 Subsystem Structuring	482
18.9 Structuring System into Tasks	487
18.10 Design of Distributed Elevator Control System	495
18.11 Design of Information Hiding Classes	503
18.12 Developing Detailed Software Design	508
18.13 Target System Configuration	511
18.14 Performance Analysis of Non-Distributed Elevator Control System	512
18.15 Performance Analysis of Distributed Elevator Control System	520
19 Banking System Case Study	531
19.1 Problem Description	531
19.2 Use Case Model	532
19.3 Static Modeling	536
19.4 Object Structuring	541
19.5 Dynamic Modeling	547
19.6 ATM Statechart	560
19.7 Design of Banking System	565
19.8 Consolidating the Collaboration Model	565
19.9 Structuring the System into Subsystems	568
19.10 Design of ATM Client Subsystem	571
19.11 Design of Bank Server Subsystem	580
19.12 Configuration of Banking System	587
19.13 Alternative Design Considerations	587
19.14 Task Behavior Specifications	587
20 Cruise Control and Monitoring System Case Study	595
20.1 Problem Description	595

20.2	Use Case Model	597
20.3	Use Case Descriptions	600
20.4	Problem Domain Static Modeling	605
20.5	Dynamic Modeling	608
20.6	Subsystem Structuring	623
20.7	Refined Static Modeling	632
20.8	Structuring the System into Tasks	635
20.9	Information Hiding Class Design	657
20.10	Developing Detailed Software Design	665
20.11	Software Architecture of Distributed Automobile System	671
21	Distributed Factory Automation System Case Study	673
21.1	Problem Description	673
21.2	Use Case Model	675
21.3	Conceptual Static Model of the Problem Domain	678
21.4	Object Structuring	680
21.5	Dynamic Model	682
21.6	Subsystem Structuring	698
21.7	Distributed Software Architecture	702
21.8	System Configuration	710
22	Electronic Commerce System Case Study	713
22.1	Electronic Commerce Problem	713
22.2	Use Case Model	714
22.3	Agent Support for Electronic Commerce System	715
22.4	Object Broker Support for Electronic Commerce System	717
22.5	Static Modeling of the Problem Domain	718
22.6	Collaboration Model	719
22.7	Distributed Software Architecture	728
Appendix A	Conventions and Alternative Notations	737
A.1	Conventions Used in This Book	737
A.2	Alternative Notation for Stereotypes	739
A.3	Alternative Notation for Active Objects	740
	Glossary	743
	Bibliography	757
	Index	767