

CONTENTS—SHORT

PART I INTRODUCTION 1

- 1 OBJECT-ORIENTED ANALYSIS AND DESIGN 3
- 2 INTRODUCTION TO A DEVELOPMENT PROCESS 17
- 3 DEFINING MODELS AND ARTIFACTS 29

PART II PLAN AND ELABORATE PHASE 33

- 4 CASE STUDY: POINT-OF-SALE 35
- 5 UNDERSTANDING REQUIREMENTS 39
- 6 USE CASES: DESCRIBING PROCESSES 47
- 7 RANKING AND SCHEDULING USE CASES 73
- 8 STARTING A DEVELOPMENT CYCLE 81

PART III ANALYZE PHASE (1) 83

- 9 BUILDING A CONCEPTUAL MODEL 85
- 10 CONCEPTUAL MODEL—ADDING ASSOCIATIONS 105
- 11 CONCEPTUAL MODEL—ADDING ATTRIBUTES 119
- 12 RECORDING TERMS IN THE GLOSSARY 131
- 13 SYSTEM BEHAVIOR—SYSTEM SEQUENCE DIAGRAMS 135
- 14 SYSTEM BEHAVIOR—CONTRACTS 145

PART IV DESIGN PHASE (1) 159

- 15 FROM ANALYSIS TO DESIGN 161
- 16 DESCRIBING REAL USE CASES 163
- 17 COLLABORATION DIAGRAMS 167
- 18 GRASP: PATTERNS FOR ASSIGNING RESPONSIBILITIES 185
- 19 DESIGNING A SOLUTION WITH OBJECTS AND PATTERNS 217
- 20 DETERMINING VISIBILITY 247
- 21 DESIGN CLASS DIAGRAMS 255

22	ISSUES IN SYSTEM DESIGN	271
PART V CONSTRUCT PHASE (1) 293		
23	MAPPING DESIGNS TO CODE	295
24	PROGRAM SOLUTION IN JAVA	309
PART VI ANALYZE PHASE (2) 315		
25	CHOOSING DEVELOPMENT CYCLE 2 REQUIREMENTS	317
26	RELATING MULTIPLE USE CASES	321
27	EXTENDING THE CONCEPTUAL MODEL	329
28	GENERALIZATION	335
29	PACKAGES: ORGANIZING ELEMENTS	349
30	POLISHING THE CONCEPTUAL MODEL	355
31	CONCEPTUAL MODEL—SUMMARY	367
32	SYSTEM BEHAVIOR	373
33	MODELING BEHAVIOR IN STATE DIAGRAMS	379
PART VII DESIGN PHASE (2) 391		
34	GRASP: MORE PATTERNS FOR ASSIGNING RESPONSIBILITIES	393
35	DESIGNING WITH MORE PATTERNS	405
PART VIII SPECIAL TOPICS 425		
36	OTHER UML NOTATION	427
37	DEVELOPMENT PROCESS ISSUES	433
38	FRAMEWORKS, PATTERNS, AND PERSISTENCE	455
APPENDIX A. RECOMMENDED READINGS 487		
APPENDIX B. SAMPLE DEVELOPMENT ACTIVITIES AND MODELS 489		
BIBLIOGRAPHY 495		
GLOSSARY 497		
INDEX 503		

CONTENTS—EXPANDED

PART I INTRODUCTION 1

1	OBJECT-ORIENTED ANALYSIS AND DESIGN	3
	1.1 Applying UML, Patterns, and Object-Oriented Analysis and Design	3
	1.2 Assigning Responsibilities	5
	1.3 What is Analysis and Design?	6
	1.4 What is Object-Oriented Analysis and Design?	6
	1.5 An Analogy—Organizing the MicroChaos Business	7
	1.6 An Example of Object-Oriented Analysis and Design	10
	1.7 Object Versus Function-Oriented Analysis and Design	14
	1.8 Warning: “Analysis” and “Design” Can Lead to Terminology Wars	14
	1.9 The Unified Modeling Language	15
2	INTRODUCTION TO A DEVELOPMENT PROCESS	17
	2.1 Introduction	17
	2.2 The UML and Development Processes	19
	2.3 Macro-level Steps	20
	2.4 Iterative Development	20
	2.5 The Plan and Elaborate Phase	23
	2.6 Build Phase—Development Cycles	25
	2.7 Choosing When to Create Artifacts	26
3	DEFINING MODELS AND ARTIFACTS	29
	3.1 Introduction	29
	3.2 Modeling Systems	29
	3.3 Sample Models	30
	3.4 Relationship Between Artifacts	31

PART II PLAN AND ELABORATE PHASE 33

4	CASE STUDY: POINT-OF-SALE	35
	4.1 The Point-of-Sale System	35
	4.2 Architectural Layers and Case Study Emphasis	36
	4.3 Our Strategy: Iterative Learning and Development	36
5	UNDERSTANDING REQUIREMENTS	39
	5.1 Introduction	39
	5.2 Requirements	41
	5.3 Overview Statement	41
	5.4 Customers	41
	5.5 Goals	42
	5.6 System Functions	42
	5.7 System Attributes	44
	5.8 Other Requirements Phase Artifacts	46
6	USE CASES: DESCRIBING PROCESSES	47
	6.1 Introduction	47
	6.2 Activities and Dependencies	49
	6.3 Use Cases	49

	6.4	Actors	52
	6.5	A Common Mistake with Use Cases	53
	6.6	Identifying Use Cases	53
	6.7	Use Case and Domain Processes	54
	6.8	Use Cases, System Functions, and Traceability	55
	6.9	Use Case Diagrams	55
	6.10	Use Case Formats	55
	6.11	Systems and their Boundaries	56
	6.12	Primary, Secondary, and Optional Use Cases	58
	6.13	Essential versus Real Use Cases	58
	6.14	Notational Points	61
	6.15	Use Cases Within a Development Process	63
	6.16	Process Steps for the Point-of-Sale System	64
	6.17	Sample Models	71
7		RANKING AND SCHEDULING USE CASES	73
	7.1	Introduction	73
	7.2	Scheduling Use Cases to Development Cycles	75
	7.3	Ranking the Point-of-Sale Application Use Cases	76
	7.4	The “Start Up” Use Case	76
	7.5	Scheduling the Point-of-Sale Application Use Cases	77
	7.6	Use Case Versions of “Buy Items”	78
	7.7	Summary	80
8		STARTING A DEVELOPMENT CYCLE	81
	8.1	Starting a Development Cycle	81
 PART III ANALYZE PHASE (1) 83			
9		BUILDING A CONCEPTUAL MODEL	85
	9.1	Introduction	85
	9.2	Activities and Dependencies	87
	9.3	Conceptual Models	87
	9.4	Strategies to Identify Concepts	91
	9.5	Candidate Concepts for the Point-of-Sale Domain	94
	9.6	Conceptual Modeling Guidelines	96
	9.7	Resolving Similar Concepts—POST versus Register	97
	9.8	Modeling the <i>Unreal</i> World	98
	9.9	Specification or Description Concepts	99
	9.10	Defining Terms in the UML	101
	9.11	Sample Models	103
10		CONCEPTUAL MODEL—ADDING ASSOCIATIONS	105
	10.1	Introduction	105
	10.2	Associations	105
	10.3	The UML Association Notation	106
	10.4	Finding Associations—Common Associations List	107
	10.5	How Detailed Should Associations Be?	109
	10.6	Association Guidelines	110
	10.7	Roles	110
	10.8	Naming Associations	111
	10.9	Multiple Associations Between Two Types	112
	10.10	Associations and Implementation	113
	10.11	Point-of-Sale Domain Associations	113
	10.12	Point-of-Sale Conceptual Model	115

11	CONCEPTUAL MODEL—ADDING ATTRIBUTES 119 <ul style="list-style-type: none"> 11.1 Introduction 119 11.2 Attributes 120 11.3 UML Attribute Notation 120 11.4 Valid Attribute Types 120 11.5 Non-primitive Attribute Types 124 11.6 Modeling Attribute Quantities and Units 125 11.7 Attributes for the Point-of-Sale System 126 11.8 Attributes in the Point-of-Sale Model 127 11.9 Multiplicity From SalesLineItem to Item 128 11.10 Point-of-Sale Conceptual Model 129 11.11 Conclusion 129
12	RECORDING TERMS IN THE GLOSSARY 131 <ul style="list-style-type: none"> 12.1 Introduction 131 12.2 Glossary 131 12.3 Activities and Dependencies 132 12.4 Sample Point-of-Sale System Glossary 132
13	SYSTEM BEHAVIOR—SYSTEM SEQUENCE DIAGRAMS 135 <ul style="list-style-type: none"> 13.1 Introduction 135 13.2 Activities and Dependencies 135 13.3 System Behavior 137 13.4 System Sequence Diagrams 137 13.5 Example of a System Sequence Diagram 137 13.6 System Events and System Operations 138 13.7 How to Make a System Sequence Diagram 140 13.8 System Sequence Diagrams and other Artifacts 140 13.9 System Events and the System Boundary 141 13.10 Naming System Events and Operations 142 13.11 Showing Use Case Text 143 13.12 Sample Models 144
14	SYSTEM BEHAVIOR—CONTRACTS 145 <ul style="list-style-type: none"> 14.1 Introduction 145 14.2 Activities and Dependencies 145 14.3 System Behavior 147 14.4 Contracts 147 14.5 Example contract—enterItem 147 14.6 Contract Sections 148 14.7 How to Make A Contract 149 14.8 Post-conditions 150 14.9 The Spirit of Post-conditions: the Stage and Curtain 151 14.10 Discussion—enterItem Post-conditions 152 14.11 How Complete Should Post-conditions Be? 153 14.12 Describing Design Details and Algorithms—Notes 153 14.13 Pre-conditions 153 14.14 Advice on Writing Contracts 154 14.15 Contracts for the Buy Items Use Case 155 14.16 Contracts for the StartUp Use Case 157 14.17 Changes to the Conceptual Model 158 14.18 Sample Models 158

PART IV DESIGN PHASE (1) 159

- 15 FROM ANALYSIS TO DESIGN 161**
 - 15.1 Analysis Phase Conclusion 161
 - 15.2 Design Phase Initiation 162

- 16 DESCRIBING REAL USE CASES 163**
 - 16.1 Introduction 163
 - 16.2 Activities and Dependencies 163
 - 16.3 Real Use Cases 163
 - 16.4 Example—Buy Items-Version 1 165
 - 16.5 Sample Models 166

- 17 COLLABORATION DIAGRAMS 167**
 - 17.1 Introduction 167
 - 17.2 Activities and Dependencies 167
 - 17.3 Interaction Diagrams 169
 - 17.4 Example Collaboration Diagram: makePayment 170
 - 17.5 Interaction Diagrams are a Valuable Artifact 170
 - 17.6 This is a Notation-Only Chapter 171
 - 17.7 Read the Following Chapters for Design Guidelines 171
 - 17.8 How to Make Collaboration Diagrams 172
 - 17.9 Basic Collaboration Diagram Notation 173
 - 17.10 Sample Models 183

- 18 GRASP: PATTERNS FOR ASSIGNING RESPONSIBILITIES 185**
 - 18.1 Introduction 185
 - 18.2 Activities and Dependencies 187
 - 18.3 Well-Designed Interaction Diagrams are Valuable 187
 - 18.4 Responsibilities and Methods 187
 - 18.5 Responsibilities and Interaction Diagrams 188
 - 18.6 Patterns 189
 - 18.7 GRASP: Patterns of General Principles in Assigning Responsibilities 191
 - 18.8 The UML Class Diagram Notation 192
 - 18.9 Expert 193
 - 18.10 Creator 197
 - 18.11 Low Coupling 200
 - 18.12 High Cohesion 203
 - 18.13 Controller 206
 - 18.14 Responsibilities, Role Playing and CRC Cards 215

- 19 DESIGNING A SOLUTION WITH OBJECTS AND PATTERNS 217**
 - 19.1 Introduction 217
 - 19.2 Interaction Diagrams and Other Artifacts 218
 - 19.3 Point-of-Sale Conceptual Model 222
 - 19.4 Collaboration Diagrams for the POST Application 222
 - 19.5 Collaboration Diagram: enterItem 223
 - 19.6 Collaboration Diagram: endSale 229
 - 19.7 Collaboration Diagram: makePayment 233
 - 19.8 Collaboration Diagram: startUp 238
 - 19.9 Connecting the Presentation Layer to the Domain Layer 243
 - 19.10 Summary 245

- 20 DETERMINING VISIBILITY 247**
 - 20.1 Introduction 247
 - 20.2 Visibility Between Objects 247
 - 20.3 Visibility 248

	20.4	Illustrating Visibility in the UML	253
21		DESIGN CLASS DIAGRAMS	255
	21.1	Introduction	255
	21.2	Activities and Dependencies	255
	21.3	When to Create Design Class Diagrams	257
	21.4	Example Design Class Diagram	257
	21.5	Design Class Diagrams	257
	21.6	How to Make a Design Class Diagram	258
	21.7	Conceptual Model versus Design Class Diagrams	259
	21.8	Creating the Point-of-Sale Design Class Diagrams	259
	21.9	Notation for Member Details	268
	21.10	Sample Models	270
	21.11	Summary	270
22		ISSUES IN SYSTEM DESIGN	271
	22.1	Introduction	271
	22.2	Classic Three-Tier Architecture	273
	22.3	Multi-Tiered Object-Oriented Architectures	274
	22.4	Showing Architecture with UML Packages	275
	22.5	Identifying Packages	278
	22.6	Layers and Partitions	278
	22.7	Visibility Between Package Classes	279
	22.8	Service Packages Interface—The Facade Pattern	280
	22.9	No Direct Visibility to Windows—The Model-View Separation Pattern	281
	22.10	Indirect Communication in a System	284
	22.11	Application Coordinators	287
	22.12	Storage and Persistence	290
	22.13	Sample Models	291

PART V CONSTRUCT PHASE (1) 293

23		MAPPING DESIGNS TO CODE	295
	23.1	Introduction	295
	23.2	Programming and the Development Process	295
	23.3	Mapping Designs to Code	298
	23.4	Creating Class Definitions from Design Class Diagrams	299
	23.5	Creating Methods from Collaboration Diagrams	302
	23.6	Updating Class Definitions	305
	23.7	Container/Collection Classes in Code	306
	23.8	Exceptions and Error Handling	306
	23.9	Defining the Sale--makeLineItem Method	307
	23.10	Order of Implementation	307
	23.11	Summary of Mapping Designs to Code	308
24		PROGRAM SOLUTION IN JAVA	309
	24.1	Introduction to the Program Solution	309

PART VI ANALYZE PHASE (2) 315

25		CHOOSING DEVELOPMENT CYCLE 2 REQUIREMENTS	317
	25.1	Development Cycle Two Requirements	317
	25.2	Assumptions and Simplifications	317

26	RELATING MULTIPLE USE CASES 321
	26.1 Introduction 321
	26.2 When to Create Separate Use Cases 321
	26.3 Use Case Diagrams with uses Relationships 322
	26.4 Use Case Documents with uses Relationships 323
27	EXTENDING THE CONCEPTUAL MODEL 329
	27.1 New Concepts in the Point-of-Sale System 329
28	GENERALIZATION 335
	28.1 Generalization 335
	28.2 Defining Super and Subtypes 336
	28.3 When to Define a Subtype 339
	28.4 When to Define a Supertype 342
	28.5 Point-of-Sale Type Hierarchies 342
	28.6 Abstract Types 345
	28.7 Modeling Changing States 347
	28.8 Class Hierarchies and Inheritance 348
29	PACKAGES: ORGANIZING ELEMENTS 349
	29.1 Introduction 349
	29.2 UML Package Notation 350
	29.3 How to Partition the Conceptual Model 351
	29.4 Point-of-Sale Conceptual Model Packages 352
30	POLISHING THE CONCEPTUAL MODEL 355
	30.1 Introduction 355
	30.2 Associative Types 355
	30.3 Aggregation and Composition 359
	30.4 Association Role Names 362
	30.5 Roles As Concepts versus Roles in Associations 363
	30.6 Derived Elements 364
	30.7 Qualified Associations 365
	30.8 Recursive or Reflexive Associations 366
31	CONCEPTUAL MODEL—SUMMARY 367
	31.1 Introduction 367
	31.2 Domain Concepts Package 368
	31.3 Core/Misc Package 368
	31.4 Payments 369
	31.5 Products 369
	31.6 Sales 370
	31.7 Authorization Transactions 371
32	SYSTEM BEHAVIOR 373
	32.1 System Sequence Diagrams 373
	32.2 New System Events 374
	32.3 Contracts 375
33	MODELING BEHAVIOR IN STATE DIAGRAMS 379
	33.1 Introduction 379
	33.2 Events, States, and Transitions 379
	33.3 State Diagrams 381
	33.4 Use Case State Diagrams 382
	33.5 System State Diagrams 383
	33.6 Use Case State Diagrams for the Point-of-Sale Application 384
	33.7 Types That Need State Diagrams 384

- 33.8 Other State Diagrams for the Point-of-Sale Application 386
- 33.9 Illustrating External and Interval Events 387
- 33.10 Additional State Diagram Notation 388

PART VII DESIGN PHASE (2) 391

- 34 **GRASP: MORE PATTERNS FOR ASSIGNING RESPONSIBILITIES 393**
 - 34.1 GRASP: General Responsibility Assignment Software Patterns 393
 - 34.2 Polymorphism 394
 - 34.3 Pure Fabrication 396
 - 34.4 Indirection 398
 - 34.5 Don't Talk to Strangers 400
- 35 **DESIGNING WITH MORE PATTERNS 405**
 - 35.1 Introduction 405
 - 35.2 State (GoF) 406
 - 35.3 Polymorphism (GRASP) 411
 - 35.4 Singleton (GoF) 413
 - 35.5 Remote Proxy and Proxy (GoF) 416
 - 35.6 Facade and Device Proxy (GoF) 418
 - 35.7 Command (GoF) 421
 - 35.8 Conclusion 424

PART VIII SPECIAL TOPICS 425

- 36 **OTHER UML NOTATION 427**
 - 36.1 Introduction 427
 - 36.2 General Notation 427
 - 36.3 Interfaces 429
 - 36.4 Implementation Diagrams 429
 - 36.5 Asynchronous Messages in Collaboration Diagrams 430
 - 36.6 Package Interfaces 432
- 37 **DEVELOPMENT PROCESS ISSUES 433**
 - 37.1 Introduction 433
 - 37.2 Why Bother? 434
 - 37.3 Guiding Principles of a Successful Process 434
 - 37.4 Iterative and Incremental Development 435
 - 37.5 Use Case Driven Development 437
 - 37.6 Early Emphasis on Architecture 437
 - 37.7 Phases of Development 438
 - 37.8 Length of Development Cycles 447
 - 37.9 Development Cycle Issues 448
 - 37.10 Scheduling Development of Architectural Layers 452
- 38 **FRAMEWORKS, PATTERNS, AND PERSISTENCE 455**
 - 38.1 Introduction 455
 - 38.2 The Problem: Persistent Objects 456
 - 38.3 The Solution: A Persistence Framework 456
 - 38.4 What is a Framework? 457
 - 38.5 Requirements for the Persistence Framework 458
 - 38.6 PersistentObject Superclass? 459
 - 38.7 Key Ideas 459
 - 38.8 Mapping—Pattern *Representing Objects as Tables* 460
 - 38.9 Object Identify—*Object Identifier* Pattern 461

38.10	Brokers— <i>Database Broker</i> Pattern	462
38.11	Framework Design— <i>Template Method</i> Pattern	463
38.12	Materialization— <i>Template Method</i> Pattern	464
38.13	Cached Objects— <i>Cache Management</i> Pattern	467
38.14	Smart References— <i>Virtual Proxy, Bridge</i> Patterns	468
38.15	Virtual Proxies and Database Brokers	473
38.16	How to Represent Relationships in Tables	475
38.17	The <i>Complex Object Instantiation</i> Pattern	476
38.18	Transaction Operations	479
38.19	Searching for Objects in the Persistent Store	483
38.20	Alternate Designs	484
38.21	Unresolved Issues	486

APPENDIX A. RECOMMENDED READINGS 487

APPENDIX B. SAMPLE DEVELOPMENT ACTIVITIES AND MODELS 489

BIBLIOGRAPHY 495

GLOSSARY 497

INDEX 503