

Object-Oriented Analysis & Design

Understanding System Development with UML 2.0



Mike O'Docherty

Contents

1	Introduction	2
1.1	Background	4
1.2	A Brief History of Programming	4
1.3	Methodologies	5
1.4	About this Book	5
1.4.1	Content Overview	6
1.4.2	Case Studies	7
1.4.3	Navigation	7
	Part I Setting the Scene	9
2	Object Concepts	10
2.1	Introduction	12
2.2	What Is an Object?	13
2.3	Identical or Equal?	16
2.4	Depicting Objects	19
2.5	Encapsulation	20
2.6	Association and Aggregation	21
2.7	Graphs and Trees	23
2.8	Links and Navigability	25
2.9	Messages	27
2.10	Invoking an Operation	28
2.11	An Example Collaboration	30
2.12	How an Object-Oriented Program Works	32
2.13	Garbage Collection	32
2.14	Classes	34
2.15	What Does a Class Define?	37
2.16	Shared Data and Shared Operations	39
2.17	Types	40
2.18	Terminology	41
2.19	Reusing Code	43

2.20 Summary	48
Further Reading	48
Review Questions	48
Answers to Activity 1	50
Answers to Review Questions	50
3 Inheritance	52
3.1 Introduction	54
3.2 Designing a Class Hierarchy	55
3.3 Adding Implementations to a Class Hierarchy	58
3.4 Abstract Classes	59
3.5 Redefining Methods	63
3.6 Implementing a Stack Class	64
3.6.1 Implementing a Stack using Inheritance	65
3.6.2 Implementing a Stack using Composition	66
3.6.3 Inheritance versus Composition	68
3.7 Multiple Inheritance	69
3.8 Guidelines for Using Inheritance	73
3.9 Summary	73
Further Reading	74
Review Questions	74
Answers to Review Questions	75
4 Type Systems	78
4.1 Introduction	80
4.2 Dynamic and Static Type Systems	80
4.3 Polymorphism	82
4.3.1 Polymorphic Variables	82
4.3.2 Polymorphic Messages	83
4.4 Dynamic Binding	85
4.5 Polymorphism Guideline	87
4.6 Type Casting	88
4.7 Explicit Casts	89
4.8 Genericity with Templates	91
4.9 Summary	93
Further Reading	93
Review Questions	93
Answers to Activity 2	95
Answers to Activity 3	95
Answers to Review Questions	96

5 Software Development Methodologies	98
5.1 Introduction	100
5.2 Classical Phases in Software Production	102
5.2.1 Requirements	102
5.2.2 Analysis	102
5.2.3 Design	103
5.2.4 Specification	103
5.2.5 Implementation	104
5.2.6 Testing	104
5.2.7 Deployment	104
5.2.8 Maintenance	104
5.2.9 Key Questions	105
5.3 Software Engineering and the Waterfall Methodology	106
5.4 New Methodologies	110
5.4.1 Spiral Methodology	110
5.4.2 Iterative Methodology	111
5.4.3 Incremental Methodology	112
5.4.4 Combining the Methodologies	113
5.5 Object-Oriented Methodologies	114
5.5.1 UML, RUP and XP	115
5.5.2 The Need for Development Tools	116
5.6 Ripple Overview	117
5.6.1 Use Case Diagram	120
5.6.2 Class Diagram (Analysis Level)	121
5.6.3 Communication Diagram	122
5.6.4 Deployment Diagram	123
5.6.5 Class Diagram (Design Level)	124
5.6.6 Sequence Diagram	125
5.7 Summary	126
Further Reading	126
Review Questions	127
Answers to Review Questions	128
 Part II Understanding the Problem	 129
6 Gathering Requirements	130
6.1 Introduction	132
6.2 The Birth of a System	133

6.3	Use Cases	135
6.4	Business Perspective	136
6.4.1	Identifying Business Actors	137
6.4.2	Writing the Project Glossary	138
6.4.3	Identifying Business Use Cases	139
6.4.4	Illustrating Use Cases on a Communication Diagram	141
6.4.5	Illustrating Use Cases on an Activity Diagram	143
6.5	Developer Perspective	145
6.5.1	Specializing Actors	149
6.5.2	Use Case Relationships	150
6.5.3	System Use Case Details	155
6.5.4	Preconditions, Postconditions and Inheritance	156
6.5.5	Supplementary Requirements	158
6.5.6	User Interface Sketches	158
6.5.7	Prioritizing System Use Cases	159
6.6	Summary	162
	Further Reading	162
	Review Questions	163
	Answers to Review Questions	165
7	Analyzing the Problem	166
7.1	Introduction	168
7.2	Why Do Analysis?	168
7.3	Overview of the Analysis Process	169
7.4	Static Analysis	170
7.4.1	Finding Classes	170
7.4.2	Identifying Class Relationships	171
7.4.3	Drawing Class and Object Diagrams	172
7.4.4	Drawing Relationships	173
7.4.5	Attributes	178
7.4.6	Association Classes	182
7.4.7	Tangible versus Intangible Objects	183
7.4.8	Good Objects	188
7.5	Dynamic Analysis	188
7.5.1	Drawing Use Case Realizations	189
7.5.2	Boundaries, Controllers and Entities	191
7.5.3	Communication Diagram Elements	192
7.5.4	Adding Operations to Classes	194
7.5.5	Responsibilities	194

7.5.6 State Modeling	195
7.6 Summary	197
Further Reading	197
Review Questions	197
Answers to Activity 4	201
Answers to Review Questions	201

Part III Designing the Solution 203

8 Designing the System Architecture 204

8.1 Introduction	206
8.2 Design Priorities	207
8.3 Steps in System Design	207
8.4 Choosing a Networked System Topology	208
8.4.1 The History of Network Architectures	208
8.4.2 Three-Tier Architecture	210
8.4.3 Personal Computers	212
8.4.4 Network Computers	213
8.4.5 The Internet and the World Wide Web	214
8.4.6 Intranets	215
8.4.7 Extranets and Virtual Private Networks	215
8.4.8 Client–Server versus Distributed Architectures	216
8.4.9 Depicting Network Topology in UML	218
8.5 Designing for Concurrency	220
8.6 Designing for Security	222
8.6.1 Digital Encryption and Decryption	223
8.6.2 General Security Rules	225
8.7 Partitioning Software	225
8.7.1 Systems and Subsystems	226
8.7.2 Layers	227
8.7.3 Java Layers: Applet plus RMI	230
8.7.4 Message Flow in Layers	233
8.8 Summary	237
Further Reading	238
Review Questions	238
Answers to Review Questions	239

9 Choosing Technologies 240

9.1 Introduction	242
------------------	-----

9.2	Client Tier Technologies	242
9.3	Client Tier to Middle Tier Protocols	244
9.4	Middle Tier Technologies	246
9.5	Middle Tier to Data Tier Technologies	247
9.6	Other Technologies	248
9.7	Typical Front-End Configurations	250
9.7.1	HTML/CGI-with-Scripts	250
9.7.2	HTML/CGI-with-Servlets	252
9.7.3	RMI	254
9.7.4	CORBA	255
9.7.5	EJB	256
9.8	Back-End Configurations	258
9.9	Java E-Commerce Configuration	258
9.10	UML Packages	262
9.11	Summary	267
	Further Reading	267
	Review Questions	267
	Answers to Review Questions	268

10 Designing the Subsystems 270

10.1	Introduction	272
10.2	Mapping the Analysis Class Model into the Design Class Model	273
10.2.1	Mapping Operations	273
10.2.2	Variable Types	274
10.2.3	Visibility of Fields	274
10.2.4	Accessors	275
10.2.5	Mapping Classes, Attributes and Compositions	276
10.2.6	Mapping Other Types of Association	277
10.2.7	Universal Identifier	282
10.3	Handling Persistence with a Relational Database	284
10.3.1	Database Management Systems	285
10.3.2	The Relational Model	286
10.3.3	Mapping Entity Classes	288
10.3.4	Mapping Associations	289
10.3.5	Mapping Object State	292
10.4	Finalizing the User Interfaces	298
10.5	Designing the Business Services	304
10.5.1	Using Proxies and Copies	305
10.5.2	Classifying Business Services	307

10.5.3	Session Identifiers	309
10.5.4	Business Service Realization	310
10.6	Using Patterns, Frameworks and Libraries	312
10.7	Transactions	312
10.7.1	Pessimistic and Optimistic Concurrency	313
10.7.2	General Guidelines for Using Transactions with Objects	314
10.7.3	Transactions in Upper Layers	315
10.8	Handling Multiple Activities	315
10.8.1	Controlling Multiple Tasks	315
10.8.2	Controlling Multiple Threads	316
10.8.3	Thread Safety	318
10.9	Summary	321
	Further Reading	321
	Review Questions	322
	Answers to Review Questions	323
11	Reusable Design Patterns	326
11.1	Introduction	328
11.1.1	A Brief History of Patterns	328
11.1.2	Software Patterns Today	329
11.2	A Pattern Template	330
11.3	Common Design Patterns	331
11.3.1	Observer	331
11.3.2	Singleton	337
11.3.3	Multiton	341
11.3.4	Iterator	342
11.3.5	Factory Method and Abstract Factory	346
11.3.6	State	347
11.3.7	Facade	352
11.3.8	Adapter	354
11.3.9	Strategy and Template Method	356
11.3.10	Flyweight	358
11.3.11	Composite	361
11.3.12	Proxy	364
11.4	Using Patterns	367
11.5	Discovering, Combining and Adapting Patterns	367
11.6	Summary	370
	Further Reading	371

12 Specifying the Interfaces of Classes	372
12.1 Introduction	374
12.2 What Is a Specification?	375
12.3 Formal Specification	376
12.4 Informal Specification	378
12.5 Dynamic Checking	380
12.6 Object-Oriented Specification	382
12.6.1 Formal Specification in OCL	383
12.6.2 Informal Specification in Eiffel	384
12.7 Design by Contract	385
12.7.1 Contracts and Inheritance	389
12.7.2 Reducing Error-Checking Code	391
12.7.3 Enforcing the Contract	394
12.7.4 Application Firewalls	395
12.8 Informal Specification in Java	396
12.8.1 Documenting a Contract using Comments	396
12.8.2 Checking Conditions Dynamically	396
12.8.3 Signaling Contract Violations using RuntimeExceptions	397
12.8.4 External Systems	398
12.8.5 Enabling and Disabling Dynamic Checks	400
12.9 Summary	402
Further Reading	403
Review Questions	403
Answers to Review Questions	404
13 Continuous Testing	406
13.1 Introduction	408
13.2 Testing Terminology	408
13.2.1 Black-Box Testing	409
13.2.2 White-Box Testing	410
13.3 Types of Test	411
13.3.1 Unit Testing	412
13.3.2 Integration Testing	412
13.3.3 Alpha Testing	413
13.3.4 Beta Testing	413
13.3.5 Use Case Testing	414
13.3.6 Component Testing	414
13.3.7 Build Testing	415
13.3.8 Load Testing	417
13.3.9 Installation Testing	418

13.3.10	Acceptance Testing	418
13.3.11	Regression Tests	419
13.3.12	Documentation Tests	419
13.3.13	Testing for Security	419
13.3.14	Metrics	420
13.4	Automating Tests	421
13.5	Preparing for Testing	422
13.6	Testing Strategies	424
13.6.1	Testing During Development	424
13.6.2	Testing During the Testing Phase	425
13.6.3	Testing After Release	426
13.7	What to Test For	426
13.8	Test-Driven Development	430
13.9	An Example of Test-Driven Development using JUnit	431
13.9.1	Testing the Car Class	433
13.9.2	Implementing the Car Class	434
13.9.3	Refactoring Tests	435
13.9.4	Creating a Test Suite for Regression Testing	439
13.9.5	Testing Across Methods	441
13.9.6	Completing the Store Class	442
13.10	Summary	445
	Further Reading	445
A	Ripple Summary	446
B	iCoot Case Study	450
B.1	Business Requirements	450
B.1.1	Customer's Mission Statement	450
B.1.2	Actor List	450
B.1.3	Use Case List	451
B.1.4	Use Case Communication Diagrams	452
B.1.5	Use Case Activity Diagrams	452
B.1.6	Use Case Details	452
B.2	System Requirements	456
B.2.1	User Interface Sketches	456
B.2.2	Actor List	460
B.2.3	Use Case List	460
B.2.4	Use Case Diagram	460
B.2.5	Use Case Survey	461

B.2.6	Use Case Details	462
B.2.7	Supplementary Requirements	465
B.2.8	Use Case Priorities	465
B.3	Analysis	466
B.3.1	Class Diagram	466
B.3.2	Attributes	467
B.3.3	Operation List	467
B.3.4	State Machine for a Reservation	469
B.3.5	Use Case Realization	470
B.4	System Design	476
B.4.1	Technology Choices	476
B.4.2	Layer Diagram	477
B.4.3	Layer Interaction Policy	479
B.4.4	Packages	479
B.4.5	Deployment Diagram	479
B.4.6	Security Policy	482
B.4.7	Concurrency Policy	482
B.5	Subsystem Design	483
B.5.1	Business Services	483
B.5.2	ServletsLayer Class Diagram	483
B.5.3	ServletsLayer Field List	484
B.5.4	ServletsLayer Message List	484
B.5.5	ServerLayer Class Diagram	485
B.5.6	ServerLayer Field List	486
B.5.7	ServerLayer Message List	486
B.5.8	BusinessLayer Class Diagram	487
B.5.9	BusinessLayer Field List	488
B.5.10	Protocol Objects Class Diagram	492
B.5.11	Database Schema	494
B.5.12	User Interface Design	494
B.5.13	Business Service Realization	494
B.6	Class Specification	508
B.6.1	Server Class Specification	508
B.6.2	Business Logic Class Specification	509
B.7	Outline Test Plan	512
B.7.1	Introduction	512
B.7.2	The Impact of Spirals and Increments	512
B.7.3	Testing of Non-Code Artifacts	513
B.7.4	Code Reviews	513

B.7.5	Test-Driven Development	513
B.7.6	Assertions	514
B.7.7	Testing Phase	514
B.7.8	Documentation Testing	515
B.7.9	Build Testing	515
B.7.10	Test Documentation and Logging	515
B.7.11	Testing Activities by Phase	516
B.8	Glossary	517
C	Summary of UML Notation Used	526
	Bibliography	538
	Index	541