

# APPLYING UML AND PATTERNS

An Introduction to Object-Oriented Analysis  
and Design and the Unified Process

SECOND EDITION

Free Book for  
Everyone

"People often ask me which is the best book to introduce them to the world of OO design. Ever since I came across it, *Applying UML and Patterns* has been my unreserved choice."

—Martin Fowler, author, *UML Distilled* and *Refactoring*

**CRAIG LARMAN**

*Foreword by Philippe Kruchten*

# TABLE OF CONTENTS

Foreword xv

Preface xvii

## PART I INTRODUCTION

1	Object-Oriented Analysis and Design	3
	Applying UML and Patterns in OOA/D	3
	Assigning Responsibilities	6
	What Is Analysis and Design?	6
	What Is Object-Oriented Analysis and Design?	7
	An Example	7
	The UML	10
	Further Readings	11
2	Iterative Development and the Unified Process	13
	The Most Important UP Idea: Iterative Development	14
	Additional UP Best Practices and Concepts	18
	The UP Phases and Schedule-Oriented Terms	19
	The UP Disciplines (was Workflows)	20
	Process Customization and the Development <i>Case</i>	23
	The Agile UP	24
	The Sequential "Waterfall" Lifecycle	25
	You Know You Didn't Understand the UP When...	26
	Further Readings	26
3	Case Study: The NextGen POS System	29
	The NextGen POS System	29
	Architectural Layers and Case Study Emphasis	30
	The Book's Strategy: Iterative Learning and Development	31

## PART II INCEPTION

4	Inception	35
	Inception: An Analogy	36
	Inception May Be Very Brief	36
	What Artifacts May Start in Inception?	37
	You Know You Didn't Understand Inception When...	38
5	Understanding Requirements	41
	Types of Requirements	42
	Further Readings	43
6	Use-Case Model: Writing Requirements in Context	45
	Goals and Stories	46
	Background	46
	Use Cases and Adding Value	47
	Use Cases and Functional Requirements	48
	Use Case Types and Formats	49
	Fully Dressed Example: Process Sale	50
	Explaining the Sections	54
	Goals and Scope of a Use Case	59
	Finding Primary Actors, Goals, and Use Cases	63
	Congratulations: Use Cases Have Been Written, and Are Imperfect	67
	Write Use Cases in an Essential UI-Free Style	68
	Actors	70
	Use <i>Case</i> Diagrams	71
	Requirements in Context and Low-Level Feature Lists	73
	Use Cases Are Not Object-Oriented	75

## TABLE OF CONTENTS

	Use Cases Within the UP	75
	Case Study: Use Cases in the NextGen Inception Phase	79
	Further Readings	79
	UP Artifacts and Process Context	81
7	Identifying Other Requirements	83
	NextGen POS Examples	84
	NextGen Example: (Partial) Supplementary Specification	84
	Commentary: Supplementary Specification	88
	NextGen Example: (Partial) Vision	91
	Commentary: Vision	93
	NextGen Example: A (Partial) Glossary	98
	Commentary: Glossary (Data Dictionary)	99
	Reliable Specifications: An Oxymoron?	100
	Online Artifacts at the Project Website	101
	Not Much UML During Inception?	101
	Other Requirement Artifacts Within the UP	101
	Further Readings	104
	UP Artifacts and Process Context	105
8	From Inception to Elaboration	107
	Checkpoint: What Happened in Inception?	108
	On to Elaboration	109
	Planning the Next Iteration	110
	Iteration 1 Requirements and Emphasis: Fundamental OOA/D Skills	112
	What Artifacts May Start in Elaboration?	118
	You Know You Didn't Understand Elaboration When...	114
<b>PART III ELABORATION ITERATION 1</b>		
9	Use-Case Model: Drawing System Sequence Diagrams	117
	System Behavior	118
	System Sequence Diagrams	118
	Example of an SSD	119
	Inter-System SSDs	120
	SSDs and Use Cases	120
	System Events and the System Boundary	120
	Naming System Events and Operations	121
	Showing Use Case Text	122
	SSDs and the Glossary	122
	SSDs Within the UP	123
	Further Readings	124
	UP Artifacts	125
10	Domain Model: Visualizing Concepts	127
	Domain Models	128
	Conceptual Class Identification	132
	Candidate Conceptual Classes for the Sales Domain	136
	Domain Modeling Guidelines	137
	Resolving Similar Conceptual Classes—Register vs. "POST"	139
	Modeling the <i>Unreal</i> World	140
	Specification or Description Conceptual Classes	140
	UML Notation, Models, and Methods: Multiple Perspectives	144
	Lowering the Representational Gap	146
	Example: The NextGen POS Domain Model	148
	Domain Models Within the UP	148
	Further Readings	150

TABLE OF CONTENTS

	UP Artifacts	151	
11	Domain Model: Adding Associations	153	
	Associations	153	
	The UML Association Notation	154	
	Finding Associations—Common Associations List	155	
	Association Guidelines	157	
	Roles	157	
	How Detailed Should Associations Be?	159	
	Naming Associations	160	
	Multiple Associations Between Two Types	161	
	Associations and Implementation	161	
	NextGen POS Domain Model Associations	162	
	NextGen POS Domain Model	163	
12	Domain Model: Adding Attributes	167	
	Attributes	167	
	UML Attribute Notation	168	
	Valid Attribute Types	168	
	Non-primitive Data Type Classes	170	
	Design Creep: No Attributes as Foreign Keys	172	
	Modeling Attribute Quantities and Units	173	
	Attributes <i>in</i> the NextGen Domain Model	174	
	Multiplicity From SalesLineItem to Item	175	
	Domain Model Conclusion	175	
13	Use-Case Model: Adding Detail with Operation Contracts	177	
	Contracts	177	
	Example Contract: enterItem	178	
	Contract Sections	179	
	Postconditions	179	
	Discussion—enterItem Postconditions	182	
	Writing Contracts Leads to Domain Model Updates	183	
	When Are Contracts Useful? Contracts vs. Use Cases?	183	
	Guidelines: Contracts	184	
	NextGen POS Example: Contracts	185	
	Changes to the Domain Model	186	
	Contracts, Operations, and the UML	186	
	Operation Contracts Within the UP	188	
	Further Readings	191	
14	From Requirements to Design in this Iteration	193	
	Iteratively Do the Right Thing, Do the Thing Right	193	
	Didn't That Take Weeks To Do? No, Not Exactly.	194	On to Object Design
		194	
15	Interaction Diagram Notation	197	
	Sequence and Collaboration Diagrams	198	Example
	Collaboration Diagram: makePayment	199	Example
	Sequence Diagram: makePayment	200	Interaction
	Diagrams Are Valuable	200	Common Interaction
	Diagram Notation	201	Basic Collaboration Diagram
	Notation	202	Basic Sequence Diagram Notation
		208	
16	GRASP: Designing Objects with Responsibilities	215	
	Responsibilities and Methods	216	
	Responsibilities and Interaction Diagrams	217	
	Patterns	218	

TABLE OF CONTENTS

	GRASP: Patterns of General Principles in Assigning Responsibilities	219
	The UML Class Diagram Notation	220
	Information Expert (or Expert)	221
	Creator	226
	Low Coupling	229
	High Cohesion	232
	Controller	237
	Object Design and CRC Cards	245
	Further Readings	246
17	Design Model: Use-Case Realizations with GRASP Patterns	247
	Use-Case Realizations	248
	Artifact Comments	249
	Use-Case Realizations for the NextGen Iteration	2.52
	Object Design: makeNewSale	253
	Object Design: enter-Item	255
	Object Design: endSale	260
	Object Design: makePayment	264
	Object Design: startUp	269
	Connecting the UI Layer to the Domain Layer	273
	Use-Case Realizations Within the UP	276
	Summary	278
18	Design Model: Determining Visibility	279
	Visibility Between Objects	279
	Visibility	280
	Illustrating Visibility in the UML	284
19	Design Model: Creating Design Class Diagrams	285
	When to Create DCDs	285
	Example DCD	286
	DCD and UP Terminology	286
	Domain Model vs. Design Model Classes	287
	Creating a NextGen POS BCD	287
	Notation for Member Details	296
	DCDs, Drawing, and CASE Tools	298
	DCDs Within the UP	298
	UP Artifacts	299
20	Implementation Model: Mapping Designs to Code	301
	Programming and the Development Process	302
	Mapping Designs to Code	304
	Creating Class Definitions from DCDs	304
	Creating Methods from Interaction Diagrams	307
	Container/Collection Classes in Code	309
	Exceptions and Error Handling	309
	Defining the Sale--makeLineItem Method	310
	Order of Implementation	311
	Test-First Programming	311
	Summary of Mapping Designs to Code	313
	Introduction to the Program Solution	313
<b>PART IV ELABORATION ITERATION 2</b>		
21	Iteration 2 and its Requirements	319
	Iteration 2 Emphasis: Object Design and Patterns	
	319 From Iteration 1 to 2	319 Iteration 2
	Requirements	321

TABLE OF CONTENTS

	Refinement of Analysis-oriented Artifacts in this Iteration	322
22	GRASP: More Patterns for Assigning Responsibilities	325
	Polymorphism	326 Pure
	Fabrication	329
	Indirection	332
	Protected Variations	334
23	Designing Use-Case Realizations with GoF Design Patterns	341
	Adapter (GoF)	342
	"Analysis" Discoveries During Design: Domain Model	345
	Factory (GoF)	346
	Singleton (GoF)	348
	Conclusion of the External Services with Varying Interfaces Problem	352
	Strategy (GoF)	353
	Composite (GoF) and Other Design Principles	358
	Facade (GoF)	368
	Observer/Publish-Subscribe/Delegation Event Model (GoF)	372
	Conclusion	380
	Further Readings	380
<b>PART V ELABORATION ITERATION 3</b>		
24	Iteration 3 and Its Requirements	383
	Iteration 3 Requirements	383
	Iteration 3 Emphasis	383
25	Relating Use Cases	385
	The include Relationship	386
	Terminology: Concrete, Abstract, Base, and Addition Use Cases	388
	The extend Relationship	389
	The generalize Relationship	390
	Use Case Diagrams	391
26	Modeling Generalization	393
	New Concepts for the Domain Model	393
	Generalization	396
	Defining Conceptual Superclasses and Subclasses	397
	When to Define a Conceptual Subclass	400
	When to Define a Conceptual Superclass	403
	NextGen POS Conceptual Class Hierarchies	403
	Abstract Conceptual Classes	406
	Modeling Changing States	408
	Class Hierarchies and Inheritance in Software	409
27	Refining the Domain Model	411
	Association Classes	411
	Aggregation and Composition	414
	Time Intervals and Product Prices—Fixing an Iteration 1 "Error"	418
	Association Role Names	419
	Roles as Concepts vs. Roles in Associations	420
	Derived Elements	421
	Qualified Associations	422
	Reflexive Associations	423
	Ordered Elements	423
	Using Packages to Organize the Domain Model	423
28	Adding New SSDs and Contracts	431
	New System Sequence Diagrams	431
	New System Operations	433
	New System Operation Contracts	434

## TABLE OF CONTENTS

29	Modeling Behavior in Statechart Diagrams	437
	Events, States, and Transitions	437
	Statechart Diagrams	438
	Statechart Diagrams in the UP?	439
	Use Case Statechart Diagrams	439
	Use Case Statechart Diagrams for the POS Application	441
	Classes that Benefit from Statechart Diagrams	441
	Illustrating External and Interval Events	443
	Additional Statechart Diagram Notation	444
	Further Readings	446
30	Designing the Logical Architecture with Patterns	447
	Software Architecture	448
	Architectural Pattern: Layers	450
	The Model-View Separation Principle	471
	Further Readings	474
31	Organizing the Design and Implementation Model Packages	475
	Package Organization Guidelines	476
	More UML Package Notation	482
	Further Readings	483
32	Introduction to Architectural Analysis and the SAD	485
	Architectural Analysis	486
	Types and Views of Architecture	488
	The Science: Identification and Analysis of Architectural Factors	488
	Example: Partial NextGen POS Architectural Factor Table	491
	The Art: Resolution of Architectural Factors	493
	Summary of Themes in Architectural Analysis	499
	Architectural Analysis within the UP	500
	Further Readings	505
33	Designing More Use-Case Realizations with Objects and Patterns	
507		
	Failover to Local Services; Performance with Local Caching	507
	Handling Failure	512
	Failover to Local Services with a Proxy (GoF)	519
	Designing for Non-Functional or Quality Requirements	523
	Accessing External Physical Devices with Adapters; Buy vs. Build	523
	Abstract Factory (GoF) for Families of Related Objects	525
	Handling Payments with Polymorphism and Do It Myself	528
	Conclusion	535
34	Designing a Persistence Framework with Patterns	537
	The Problem: Persistent Objects	538
	The Solution: A Persistence Service from a Persistence Framework	538
	Frameworks	539
	Requirements for the Persistence Service and Framework	540
	Key Ideas	540
	Pattern: Representing Objects as Tables	541
	UML Data Modeling Profile	541
	Pattern: Object Identifier	542
	Accessing a Persistence Service with a Facade	543
	Mapping Objects: Database Mapper or Database Broker Pattern	543
	Framework Design with the Template Method Pattern	546
	Materialization with the Template Method Pattern	546
	Configuring Mappers with a MapperFactory	552
	Pattern: Cache Management	552
	Consolidating and Hiding SQL Statements in One Class	553

## TABLE OF CONTENTS

29	Modeling Behavior in Statechart Diagrams	437
	Events, States, and Transitions	437
	Statechart Diagrams	438
	Statechart Diagrams in the UP?	439
	Use Case Statechart Diagrams	439
	Use Case Statechart Diagrams for the POS Application	441
	Classes that Benefit from Statechart Diagrams	441
	Illustrating External and Interval Events	443
	Additional Statechart Diagram Notation	444
	Further Readings	446
30	Designing the Logical Architecture with Patterns	447
	Software Architecture	448
	Architectural Pattern: Layers	450
	The Model-View Separation Principle	471
	Further Readings	474
31	Organizing the Design and Implementation Model Packages	475
	Package Organization Guidelines	476
	More UML Package Notation	482
	Further Readings	483
32	Introduction to Architectural Analysis and the SAD	485
	Architectural Analysis	486
	Types and Views of Architecture	488
	The Science: Identification and Analysis of Architectural Factors	488
	Example: Partial NextGen POS Architectural Factor Table	491
	The Art: Resolution of Architectural Factors	493
	Summary of Themes in Architectural Analysis	499
	Architectural Analysis within the UP	500
	Further Readings	505
33	Designing More Use-Case Realizations with Objects and Patterns	
507		
	Failover to Local Services; Performance with Local Caching	507
	Handling Failure	512
	Failover to Local Services with a Proxy (GoF)	519
	Designing for Non-Functional or Quality Requirements	523
	Accessing External Physical Devices with Adapters; Buy vs. Build	523
	Abstract Factory (GoF) for Families of Related Objects	525
	Handling Payments with Polymorphism and Do It Myself	528
	Conclusion	535
34	Designing a Persistence Framework with Patterns	537
	The Problem: Persistent Objects	538
	The Solution: A Persistence Service from a Persistence Framework	538
	Frameworks	539
	Requirements for the Persistence Service and Framework	540
	Key Ideas	540
	Pattern: Representing Objects as Tables	541
	UML Data Modeling Profile	541
	Pattern: Object Identifier	542
	Accessing a Persistence Service with a Facade	543
	Mapping Objects: Database Mapper or Database Broker Pattern	543
	Framework Design with the Template Method Pattern	546
	Materialization with the Template Method Pattern	546
	Configuring Mappers with a MapperFactory	552
	Pattern: Cache Management	552
	Consolidating and Hiding SQL Statements in One Class	553



TABLE OF CONTENTS

Transactional States and the State Pattern 554 Designing  
a Transaction with the Command Pattern 556 Lazy  
Materialization with a Virtual Proxy 559 How to Represent  
Relationships in Tables 562 PersistentObject Superclass and  
Separation of Concerns 563 Unresolved Issues 564

**PART VI SPECIAL TOPICS**

35 On Drawing and Tools 567  
On Speculative Design and Visual Thinking 567  
Suggestions for UML Drawing Within the Development Process 568  
Tools and Sample Features 571  
Example Two 573

36 Introduction to Iterative Planning and Project Issues 575  
Ranking Requirements 576  
Ranking Project Risks 579  
Adaptive vs. Predictive Planning 579  
Phase and Iteration Plans 581  
Iteration Plan: What to Do in the Next Iteration? 582  
Requirements Tracking Across Iterations 583  
The (Invalidity of Early Estimates 585  
Organizing Project Artifacts 585  
Some Team Iteration Scheduling Issues 586  
You Know You Didn't Understand Planning in the UP When... 588  
Further Readings 588

37 Comments on Iterative Development and the UP 589  
Additional UP Best Practices and Concepts 589  
The Construction and Transition Phases 591  
Other Interesting Practices 592  
Motivations for Timeboxing an Iteration 593  
The Sequential "Waterfall" Lifecycle 593  
Usability Engineering and User Interface Design 599  
The UP Analysis Model 599  
The RUP Product 600  
The Challenge and Myths of Reuse 601

38 More UML Notation 603  
General Notation 603  
Implementation Diagrams 604  
Template (Parameterized, Generic) Class 606  
Activity Diagrams 607

Bibliography 609 Glossary 615 Index 621