



UML 2 AND THE UNIFIED PROCESS

SECOND EDITION

PRACTICAL OBJECT-ORIENTED ANALYSIS AND DESIGN

JIM ARLOW
ILA NEUSTADT



Praise for *UML 2 and the Unified Process, Second Edition*

“OMG’s UML 2 standard defines UML very systematically and thoroughly, but lacks how to actually apply UML 2 in a real-world project. That is where *UML 2 and the Unified Process, Second Edition*, comes in. The book manages to convey the practical use of UML 2 in clear and understandable terms with many examples and guidelines. Even for people not working with the Unified Process, the book is still of great use. *UML 2 and the Unified Process, Second Edition*, is a must-read for every UML 2 beginner and a helpful guide and reference for the experienced practitioner.”

—Roland Leibundgut, Technical Director, Zühlke Engineering Ltd.

“The authors provide great detail on the constructs of the UML and how they support the Unified Process. This book is a good starting point for organizations and individuals who are adopting UP and need to understand how to provide visualization of the different aspects needed to satisfy it.”

—Eric Naiburg
Market Manager, Desktop Products
IBM Rational Software

“Many books today cover either UML or the Unified Process (UP), but not both. Arlow and Neustadt have filled this void with a book that is a perfect blend of UML and the UP. The authors provide a wealth of experience invaluable to novice modelers and seasoned OO analysts and designers. The logical organization based on the UP workflows and the unique presentation style of using activity diagrams to provide book and chapter overviews make reading less demanding. Both practitioners and students will certainly keep this book within close reach all the time.”

—Ishan De Silva
Software Engineer
Millennium Information Technologies, Sri Lanka

“If you are looking for a cookbook, read something else. This book will make you think! It explains all the syntactic elements of UML, but more importantly it gives practical advice on how and when to use—or not to use—UML. You will learn to think about the role that modeling plays in the development process. This knowledge enables you to decide for yourself how and when to use UML to gain the optimal situation for your project. *UML 2 and the Unified Process, Second Edition*, will prepare you for successfully using UML.”

—Jos Warmer
Ordina System Integration & Development, The Netherlands

“The authors have crafted a book that brings two important topics, UML and the Unified Process, together in one volume. *UML 2 and the Unified Process* is an excellent reference for UML 2. It tells you what UML can do and how to apply it to the analysis and design disciplines of the Unified Process. This book belongs on every professional’s bookshelf.”

—Gary Pollice
Professor of Practice, Computer Science
Worcester Polytechnic Institute

UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design

Table of Contents

Contents

Acknowledgments

Preface

Part 1 Introducing UML and UP

1 What is UML?

- 1.1 Chapter roadmap
- 1.2 What is UML?
- 1.3 The birth of UML
- 1.4 MDA the future of UML
- 1.5 Why unified?
- 1.6 Objects and UML
- 1.7 UML structure
- 1.8 UML building blocks
- 1.9 UML common mechanisms
- 1.10 Architecture
- 1.11 What we have learned

2 What is the Unified Process?

- 2.1 Chapter roadmap
- 2.2 What is UP?
- 2.3 The birth of UP
- 2.4 UP and the Rational Unified Process
- 2.5 Instantiating UP for your project
- 2.6 UP axioms

Table of Contents

- 2.7 UP is an iterative and incremental process
- 2.8 UP structure
- 2.9 UP phases
- 2.10 What we have learned

Part 2 Requirements

3 The requirements workflow

- 3.1 Chapter roadmap
- 3.2 The requirements workflow
- 3.3 Software requirements metamodel
- 3.4 Requirements workflow detail
- 3.5 The importance of requirements
- 3.6 Defining requirements
- 3.7 Finding requirements
- 3.8 What we have learned

4 Use case modeling

- 4.1 Chapter roadmap
- 4.2 Use case modeling
- 4.3 UP activity: Find actors and use cases
- 4.4 UP activity: Detail a use case
- 4.5 Use case specification
- 4.6 Requirements tracing
- 4.7 When to apply use case modeling
- 4.8 What we have learned

5 Advanced use case modeling

- 5.1 Chapter roadmap
- 5.2 Actor generalization
- 5.3 Use case generalization
- 5.4 «include»
- 5.5 «extend»
- 5.6 When to use advanced features

Table of Contents

5.7 Hints and tips for writing use cases

5.8 What we have learned

Part 3 Analysis

6 The analysis workflow

6.1 Chapter roadmap

6.2 The analysis workflow

6.3 Analysis artifacts metamodel

6.4 Analysis workflow detail

6.5 Analysis model rules of thumb

6.6 What we have learned

7 Objects and classes

7.1 Chapter roadmap

7.2 What are objects?

7.3 UML object notation

7.4 What are classes?

7.5 UML class notation

7.6 Scope

7.7 Object construction and destruction

7.8 What we have learned

8 Finding analysis classes

8.1 Chapter roadmap

8.2 UP activity: Analyze a use case

8.3 What are analysis classes?

8.4 Finding classes

8.5 Creating a first-cut analysis model

8.6 What we have learned

9 Relationships

9.1 Chapter roadmap

9.2 What is a relationship?

9.3 What is a link?

Table of Contents

9.4 What is an association?

9.5 What is a dependency?

9.6 What we have learned

10 Inheritance and polymorphism

10.1 Chapter roadmap

10.2 Generalization

10.3 Class inheritance

10.4 Polymorphism

10.5 Advanced generalization

10.6 What we have learned

11 Analysis packages

11.1 Chapter roadmap

11.2 What is a package?

11.3 Packages and namespaces

11.4 Nested packages

11.5 Package dependencies

11.6 Package generalization

11.7 Architectural analysis

11.8 What we have learned

12 Use case realization

12.1 Chapter roadmap

12.2 UP activity: Analyze a use case

12.3 What are use case realizations?

12.4 Use case realization elements

12.5 Interactions

12.6 Lifelines

12.7 Messages

12.8 Interaction diagrams

12.9 Sequence diagrams

12.10 Combined fragments and operators

Table of Contents

12.11 Communication diagrams

12.12 What we have learned

13 Advanced use case realization

13.1 Chapter roadmap

13.2 Interaction occurrences

13.3 Continuations

13.4 What we have learned

14 Activity diagrams

14.1 Chapter roadmap

14.2 What are activity diagrams?

14.3 Activity diagrams and the UP

14.4 Activities

14.5 Activity semantics

14.6 Activity partitions

14.7 Action nodes

14.8 Control nodes

14.9 Object nodes

14.10 Pins

14.11 What we have learned

15 Advanced activity diagrams

15.1 Chapter roadmap

15.2 Connectors

15.3 Interruptible activity regions

15.4 Exception handling

15.5 Expansion nodes

15.6 Sending signals and accepting events

15.7 Streaming

15.8 Advanced object flow features

15.9 Multicast and multireceive

15.10 Parameter sets

Table of Contents

- 15.11 «centralBuffer» node
- 15.12 Interaction overview diagrams
- 15.13 What we have learned

Part 4 Design

16 The design workflow

- 16.1 Chapter roadmap
- 16.2 The design workflow
- 16.3 Design artifacts metamodel
- 16.4 Design workflow detail
- 16.5 UP activity: Architectural design
- 16.6 What we have learned

17 Design classes

- 17.1 Chapter roadmap
- 17.2 UP activity: Design a class
- 17.3 What are design classes?
- 17.4 Anatomy of a design class
- 17.5 Well-formed design classes
- 17.6 Inheritance
- 17.7 Templates
- 17.8 Nested classes
- 17.9 What we have learned

18 Refining analysis relationships

- 18.1 Chapter roadmap
- 18.2 Design relationships
- 18.3 Aggregation and composition
- 18.4 Aggregation semantics
- 18.5 Composition semantics
- 18.6 How to refine analysis relationships
- 18.7 One-to-one associations
- 18.8 Many-to-one associations

Table of Contents

- 18.9 One-to-many associations
- 18.10 Collections
- 18.11 Reified relationships
- 18.12 Exploring composition with structured classes
- 18.13 What we have learned

19 Interfaces and components

- 19.1 Chapter roadmap
- 19.2 UP activity: Design a subsystem
- 19.3 What is an interface?
- 19.4 Provided and required interfaces
- 19.5 Interface realization vs. inheritance
- 19.6 Ports
- 19.7 Interfaces and component-based development
- 19.8 What is a component?
- 19.9 Component stereotypes
- 19.10 Subsystems
- 19.11 Finding interfaces
- 19.12 Designing with interfaces
- 19.13 Advantages and disadvantages of interfaces
- 19.14 What we have learned

20 Use case realization design

- 20.1 Chapter roadmap
- 20.2 UP activity: Design a use case
- 20.3 Use case realization design
- 20.4 Interaction diagrams in design
- 20.5 Modeling concurrency
- 20.6 Subsystem interactions
- 20.7 Timing diagrams
- 20.8 Example of use case realization design
- 20.9 What we have learned

Table of Contents

21 State machines

- 21.1 Chapter roadmap
- 21.2 State machines
- 21.3 State machines and the UP
- 21.4 State machine diagrams
- 21.5 States
- 21.6 Transitions
- 21.7 Events
- 21.8 What we have learned

22 Advanced state machines

- 22.1 Chapter roadmap
- 22.2 Composite states
- 22.3 Submachine states
- 22.4 Submachine communication
- 22.5 History
- 22.6 What we have learned

Part 5 Implementation

23 The implementation workflow

- 23.1 Chapter roadmap
- 23.2 The implementation workflow
- 23.3 Implementation artifacts metamodel
- 23.4 Implementation workflow detail
- 23.5 Artifacts
- 23.6 What we have learned

24 Deployment

- 24.1 Chapter roadmap
- 24.2 UP activity: Architectural implementation
- 24.3 The deployment diagram
- 24.4 Nodes
- 24.5 Artifacts

Table of Contents

24.6 Deployment

24.7 What we have learned

Part 6 Supplementary material

25 Introduction to OCL

25.1 Chapter roadmap

25.2 What is the Object Constraint Language (OCL)?

25.3 Why use OCL?

25.4 OCL expression syntax

25.5 Package context and pathnames

25.6 The expression context

25.7 Types of OCL expressions

25.8 The expression body

25.9 OCL navigation

25.10 Types of OCL expression in detail

25.11 OCL in other types of diagrams

25.12 Advanced topics

25.13 What we have learned

Appendix 1: Example use case model

Appendix 2: XML and use cases

Bibliography

Index