1 Introduction

1.1 Introduction

This book introduces and guides you through the use of the Unified Modeling Language (UML) and the Unified Process (both originally devised by Grady Booch, James Rumbaugh and Ivar Jacobson) and their application to Java systems. This means that the book will present you with the notation used in the UML and the steps described by the Unified Process with particular reference to the Java environment (including the classes and the syntax).

The book itself is structured in two parts. The first part introduces object-oriented analysis and design and the Unified Process. The UML is introduced, as necessary, to support the Unified Process steps. The second part of the book provides a detailed worked case study. The case study follows the whole design process through from inception to implementation in Java.

The first part of the book is structured in the following manner:

Chapter 2: Object-Oriented Analysis and Design
This chapter surveys the most significant object-oriented design and analysis methods to emerge since the late 1980s.

Chapter 3: An Introduction to the UML and the Unified Process
This chapter provides the background to the UML and the Unified Process. It also presents a summary of both.

Chapter 4: Software Architecture and Object-Oriented Design
This chapter explains and justifies why an architecture is essential to the successful design and implementation of a large object-oriented system.

Chapter 5: Requirements Workflow: Use Case Analysis
This chapter introduces the requirements workflow (which may also be known as Use Case Analysis). This workflow attempts to identify what the functionality of the system will be. These use cases will be essential as the backbone to the whole design process.

Chapter 6: The Analysis Workflow
This chapter considers the analysis of the requirements as described by the use cases. This process helps to identify the primary system requirements necessary to support the use cases.

Chapter 7: The Design Workflow: System and Class Design
The design workflow chapter moves the results of the analysis workflow forward to the actual design of the system.

Chapter 8: Implementation Phase
Having produced a design it is then necessary to move the design into an implementation in Java.

Chapter 9: The Test Workflow: How it Relates to Use Cases
Testing is a huge subject in its own right. This chapter therefore focuses on the use of use cases as the driving force behind the identification of test cases.

Chapter 10: The Four Phases
The Unified Process is made up of four phases that a project may cycle through during its lifetime. These phases apply the workflows described above. This chapter considers and describes the four phases and highlights the focus of each phase.

Chapter 11: Software Patterns
This chapter presents a detailed look at the design pattern concept.

The second part of the book relates to the real-world Unified Process and UML and has the following format:

Chapter 12: The JDSync Case Study
This chapter presents a detailed worked case study.

Chapter 13: Are UML Designs Language Independent?
This chapter considers UML and Java and how UML can be mapped into Java.

Chapter 14: Customizing the Unified Process for Short Time-Scale Projects
This chapter discusses the issue of customizing the Unified Process for short-term projects (rather than the long-lived projects assumed by default in the Unified Process).

Chapter 15: Augmenting the Unified Process with Additional Techniques
This chapter describes how the Unified Process can be augmented with additional techniques.

Chapter 16: Inheritance Considered Harmful!
This chapter discusses when it is appropriate to take advantage of component-based reuse rather than inheritance.

1.2 Why UML and the Unified Process?

A question which should be answered straight away is "why use the UML and the Unified Process?". The simple answer to the first part of this is that the UML, or Unified Modeling Language, has become the *de facto* standard. This is not necessarily a cast-iron reason for adopting a particular approach or notation. However, in this case the *de facto* standard has been adopted by the Object Management Group (the OMG) and by (almost) all vendors of object modelling tools. This allows a common language to be used whether you are working with Rational's Rose, Select's Enterprise Modeller or indeed Visio Enterprise. As the UML is not tied to a particular modelling approach you can also apply it via whichever design method you wish.

This is actually a very important point – the UML is a notation, not a method. I personally have heard numerous people talk about adopting or applying the UML method. This is a warning flag that these people probably don't know very much about object-oriented design. As the UML is a notation you cannot say you are going to apply the UML method – it is just plain nonsense.

The final reason for adopting the UML is that it has been a long time in gestation and has been open to public review for a number of years now. This has ensured that many people worldwide have been able to have input into the UML, rather than just a few behind the closed doors of some university or company. The result is not necessary the final word on notations, but it is certainly better than anything else around at present.

This should have justified the presentation of the UML in this book, but what about the Unified Process? Like the UML, the Unified Process was developed by the "three amigos", Booch, Rumbaugh and Jacobson, with support from Rational, their employer. It is explicitly designed to work with the UML and indeed was developed in tandem with the UML, but has taken longer to come out into the public domain. This is partly due to the fact that it is easier to produce a notation than it is to produce a whole design method which covers the majority of the life cycle of a software product. However, the ambitious goal of the Unified Process is to do just that!

So, other than the fact that the same people who developed the UML developed the Unified Process, is there any other reason for choosing the Unified Process? Actually, yes. I have been recommending for some time that projects should adopt a hybrid approach to their object-oriented analysis and design, primarily based around the Object Modeling Technique (OMT), some elements of Objectory and a bit of Booch and Fusion. However, this was a rather informal hybrid, directed more by personal judgement than by an explicit process. Nevertheless, this is essentially what the Unified Process does, but it does it far more formally than I ever did and goes much further than I ever went. It addresses many of the areas with which I was not comfortable in my own efforts and introduces techniques to deal with issues which I had not even considered. I therefore decided to adopt this design method.

1.3 Why this Book?

We have now covered what this book is about and justified the choice of the UML and Unified Process as the notation and method respectively that we have adopted. What we have yet to cover is why I felt that it was necessary to write this book in the first place. Essentially I was moved to write it, as, having waded through the weighty tomes on the Unified Process and read many books on the UML, I felt that I had not been presented with a clear guide on how to apply the notation and the method for the project work I am involved in, which concentrates on Java. This book therefore focuses on applying the notation and the method to Java. This does not mean that it is without value to someone who is not interested in Java. A major problem that I have had with the Rational books on the Unified Process is that I do not find them very accessible. They are written (to my mind at least) more as academic texts than as a practitioner's workbook. This book therefore attempts to focus on actual

practitioners and on providing an easily accessible step by step guide to applying UML and the Unified Process.

1.4 Where to Get More Information

The following books are useful additional references on the Unified Process and the Unified Modeling Langauge.

Jacobson, I., Booch, G. and Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley, Reading, MA.
Booch, G., Rumbaugh, J. and Jacobson, I. (1999). *The UML User Guide*. Addison-Wesley, Reading, MA.
Eriksson, H. and Penker, M. (1998). *UML Toolkit*. John Wiley, New York.
Fowler, M. and Scott, K. (1997). *UML Distilled*. Addison-Wesley, Reading, MA.
Hunt, J. (1999). *Java for Practitioners*, Springer-Verlag, London.

1.5 Where to Go Online

One important point about both the UML and the Unified Process is that both have Web sites dedicated to them. The places you should start with include the OMG (Object Management Group) and Rational Corp.

OMG for UML:

```
http://www.omg.org/
```

Rational for UML and Unified Process:

```
http://www.rational.com/
```

UML User Group:

```
http://www.valtech.com/about/umlug.htm
```

Unified Modeling Language Revision Task Force:

```
http://uml.systemhouse.mci.com/
```

Object References:

```
http://www.jaydeetechnology.co.uk/
```