Book Review

*Developing Applications with JAVA and UML*
by Paul R. Reed, Jr.

Addison-Wesley, 2002
ISBN: 0-201-70252-5
Cover Price: US$44.99
504 Pages

As Paul Reed explains in the Preface to this book, there are many publications on the market that cover Java-enabled extensions such as JavaBeans, servlets, and JavaServer Pages, but none of them really covers project planning, software process, and the methodology for building enterprise level Java applications. And that is what this book seeks to address. Using a holistic approach toward enterprise software development, Reed applies the Unified Process model, Unified Modeling Language (UML) notation, and the latest in mainstream Java technology as he walks readers through an extended sample project for a hypothetical musical instrument manufacturer. The book illustrates and emphasizes the benefits of this approach versus coding from an informal set of requirements and a sketchy whiteboard design. And along the way, Reed offers tips on design considerations and component deployment -- based on his own experience -- which might be helpful for novice developers.

This book is a follow-on to Reed's *Developing Applications with Visual Basic and UML* and probably shares some of the same UML content. It would be of interest to developers who:

- Are working on a project in Java.
- Would like to use best practices and the latest set of modeling tools.
- Would like to use Java Server Pages (JSPs), Java servlets, and the Enterprise Java Beans (EJB) 2.0 framework.
- Need help in understanding how to make all of the above work for their project.

You do need some Java experience to get the most out of this book, but prior knowledge of UML or the Unified Process is not necessary, as the book covers all the core concepts and artifacts. It also helps to have a basic understanding of JSPs and EJBs, but there is little in the book that couldn't be understood after a quick review on the Java technology Web
site. Some knowledge of Structured Query Language (SQL) and relational databases would also be helpful in order to fully appreciate the Data Access Objects (DAO) constructs Reed presents (as well as the lack thereof in his preferred solution).

**Review in Early Chapters**

The first two chapters emphasize the importance of having an iterative, incremental, risk-based software development process. Although Reed introduces both the Unified Process and his own Synergy Process, the rest of the book actually uses the Unified Process model, providing details about the Synergy process in an Appendix. UML is introduced along with a core set of key deliverables.

These chapters also provide a review on Java and how UML diagrams map to Java. As the Preface explains, this review is not aimed at teaching a beginner how to program in Java. Moreover, these chapters may not appeal to readers, myself included, who dislike terse previews of material that will be developed later in the book. I found the later chapters to be much more enjoyable. If you are a reader who likes to learn from examples, then the rest of the chapters are made for you. It is here that the book excels. Although some of the content is introductory, Reed is good about providing references for more detailed information on the technologies he discusses, including JSPs and EJBs.

**Sample Project**

The sample project that Reed introduces in these latter chapters is an order entry, billing, and inventory system for musical instruments. The goal is to have a lightweight client browser at the front end sending requests to a Web server, which then acts as a broker in a container product that implements the application's business rules. The Web server uses servlets and JSPs to intercept the requests. For the business rules, alternate solution strategies involving JavaBeans and EJBs with Bean-managed/container-managed persistence are discussed, implemented, and compared.

Although the application is simple, it makes use of a core set of UML diagrams and the Java framework. Reed introduces and explains UML artifacts on an as-needed basis, and develops and refines the UML diagrams as the book progresses. Rational Rose is used for modeling in UML and later for Java prototype generation. For areas in which he feels UML is lacking, Reed introduces some of his own artifacts.

For this example, estimates for the number of person-hours needed to complete the project are based on an approximation heuristic. Reed provides additional details about this method in the Appendix; he approximates the numbers representing weight factors for actors, technical know-how, and so on; then he plugs these into preset formulae to obtain results. (The book also provides a reference for further reading.)

These example chapters are well constructed; they build upon each other nicely, and the content is clearly presented, easy to understand, and self-
contained. Rarely do you encounter a loose end that goes unexplained. I would put myself in the beginner category regarding some of the material discussed in this book, and I am happy to say that at times I found it hard to put the book down. The reading was really enjoyable, and I didn't want to interrupt the flow of information.

One small complaint is that some of the later chapters do not list page numbers or table numbers for the JavaBeans and diagrams they reference in previous chapters. That meant that I had to search the Index or thumb back through the pages to find these things. Another small complaint is that the book ends with the Elaboration phase of the Unified Process (Chapter 12), after having walked the reader through nothing besides normal (untroubled) pathways for the sample use-cases. Although it may not have been a goal of the book to see the project through to its Transition phase, I would have liked to see some more tips on the last steps as well as typical problem-solution scenarios for use cases that do not proceed so perfectly.

In general, however, I enjoyed learning from Paul Reed. If you are new to UML or JavaBeans, JSPs, and EJBs, then this book may help you get up to speed more quickly.

-Kaveesh Mishra
Staff Software Engineer
Rational Software

Read Chapters 3 and 4 of Developing Applications with JAVA and UML in this month's issue of The Rational Edge!

For more information on the products or services discussed in this article, please click here and follow the instructions provided. Thank you!