Book Review

*Developing Enterprise Java Applications with J2EE and UML*
by Khawar Zaman Ahmed and Cary E. Umrysh

Addison Wesley, 2002
330 Pages

Ahmed's and Umrysh's book provides a high-level overview of Java Platform 2, Enterprise Edition (J2EE); Unified Modeling Language (UML); and how UML-based representations of J2EE systems can be evolved using an analysis-and-design process that is a subset of the Rational Unified Process® (RUP®). The book can be a useful addition to your technical collection if:

- You are a Java developer looking for an easy-to-read introduction to J2EE and UML.
- You are a J2EE developer who needs an initial introduction to UML.
- You have a good understanding of UML, and you need an introduction to J2EE technologies.

If you are looking for a reference that provides technical depth on subjects such as the details of J2EE source code or how to address common J2EE issues using standard design patterns, then this is not the final book for you. It still can, however, provide background knowledge and understanding that will enable you to digest the more in-depth technical references.

Organization

Although the book is not formally organized as such, it can be divided into three conceptual sections:

- Concepts definition
- Process overview
- Concepts application
The first two sections prepare the reader to benefit from the third, which contains the true meat of the book. There, Ahmed and Umrysh employ a simple analysis-and-design process to progressively evolve UML diagrams that describe a piece of a J2EE-based system.

**Concepts Definition**

The first four chapters of the book provide some useful conceptual background on the enterprise applications problem space addressed by J2EE, the key technologies and APIs that compose J2EE, UML basics, and UML representation of basic Java constructs.

Java developers should be able to absorb the UML introduction easily. Ahmed and Umrysh present only the pieces of UML that are used in diagrams they build across later chapters. Further, they illustrate many of the UML concepts using figures that include Java code and the resulting UML representation.

One type of UML diagram they do not illustrate in the foundations section is statecharts. This might cause some confusion for readers who are not familiar with state-based modeling, because the discussions of component lifecycles in later chapters often document lifecycle models using statecharts. Readers should refer to one of the more complete UML texts (e.g., *The Unified Modeling Language Reference Manual* by Rumbaugh et al., Addison-Wesley, 1999) for background information on these diagrams.

**Process Overview**

Chapters 5 and 6 provide brief surveys of different software development processes and means for describing system architecture. Chapter 6 is particularly useful for readers who are struggling to justify why they should spend time defining and documenting a consistent systems architecture. This chapter also provides useful discussion of key concepts, such as decomposition, frameworks, patterns, and layering, that are central to defining and describing such architectures.

Chapters 7 and 8 provide details about the analysis-and-design approach the book adopts in walking through the diagrams. This process is a subset of the activities defined for RUP's Analysis and Design Discipline. Specifically, the authors:

- Start with the assumption that the development team has defined requirements in the form of system use cases.
- Use RUP's Use-Case Analysis activity to define how high-level analysis elements interact to satisfy the system's functional requirements.
- Use RUP's Use-Case Design activity to transform the analysis elements into implementation elements. This transformation progresses throughout the Concepts Application section of the book, as the authors present aspects of J2EE that address different systems issues.
The process steps used in the book are sufficient to support the book's purpose, which is to illustrate how to represent a J2EE system using UML. The authors do point out that their process is just a small portion of RUP, but some readers might have been better served if the authors had described the full RUP analysis and design discipline from which they extracted their process.

**Concepts Application**

The major value of this book resides in Chapters 10-14, which have a consistent format:

- Presentation of a specific J2EE technology -- either Servlets, Java Server Pages (JSPs), Session Enterprise JavaBeans (EJB)s, Entity EJBs, or Message EJBs.
- Types of issues addressed by the technology.
- Modeling the technology using UML.
- Adding instances of that technology into the diagrams that describe the sample system.

The authors have made a concerted effort to adopt either generally accepted practice or approved specifications for their modeling guidance. Specifically:

- For the most part, the chapters on modeling Servlets and JSPs (Web components) follow the guidance provided in Jim Conallen's book, *Building Web Applications with UML*. Conallen's work is largely accepted as the de facto standard approach for Web application modeling.
- The authors' UML-to-Java mappings follow the version of Java Specification Request-26 (JSR-26) that was undergoing final review while the book was being written. Ahmed's and Umrysh's approach to illustrating the mappings is easier to follow than is the specification itself.

It is gratifying to see this usage of recognized best practices. Our industry needs convergence rather than divergence in approaches.

Chapter 15 provides the first clear discussion that I have seen regarding using UML to model the deployment of J2EE-based systems. This chapter also provides an enlightening discussion of how the simple UML model that has been developed throughout the book facilitates the traceability of artifacts from requirements through deployed modules.

Finally, Chapter 16 presents the case study that is used throughout the book.

**Things I'd Like to See in the Second Edition**

I've talked with a number of other readers of this book; all have found it a
delightful introduction to the subject matter. There are a few things, however, that I would like to see in a second edition:

- Better in-text attribution of original references. Although the reference list is reasonably comprehensive, the authors provide limited in-text attributions to these references. For more conscientious readers, this makes it difficult to identify which particular reference to examine for more in-depth understanding of a given topic or technique.

- Update the EJB modeling guidelines to reflect the contents of the version of the JSR-26 specification that currently is under review. One unfortunate aspect of documenting a quickly evolving technology such as J2EE is that the lifespan of any definitive reference is fairly short. This is true of the initial release of JSR-26, which is based upon the EJB 1.1 standard. JSR-26 is being revised to incorporate new features of the EJB 2.0 standard and to clean up some holes in the first release of the specification. The changes are not radically different, but it would be great to have an updated edition of the book-under-review that provides popular access to the new aspects of JSR-26.

- Incorporate discussion of the Core J2EE Patterns into the book. In the time since the authors wrote their book, Sun has published a series of design patterns that are recommended for addressing commonly occurring J2EE system design issues. The authors discuss design patterns at many points in their book, so recasting these discussions in terms of the de facto standard J2EE design patterns would be in line with their emphasis on applying recognized best practices.

**Final Thoughts**

*Developing Enterprise Java Applications with J2EE and UML* most likely will retain a spot on my reference bookshelf for the foreseeable future. Its high-level approach to the UML and J2EE make it a quick and easy read for brushing up on the key aspects of modeling J2EE systems. It is a recommended read for anyone who needs an overview either of J2EE or UML, or on how to use the two together effectively. It will not be a primary reference for readers who need technical depth on those subjects, but it provides an initial foundation upon which deeper study can be built.

-Todd Dunnavant
Rational Software

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

Copyright Rational Software 2002 | Privacy/Legal Information