Preface

This book is about patterns for the Java 2 platform, Enterprise Edition (J2EE). These J2EE patterns provide solutions for problems typically encountered by designers of software applications for the J2EE platform. All the patterns documented in the catalog have been discovered in the field, where they have been used to create successful J2EE applications for our customers.

This book describes proven solutions for the J2EE platform with a particular emphasis on such key J2EE technologies as: Java Server Pages (JSP), Servlets, Enterprise JavaBeans (EJB) components, Java Message Service (JMS), JDBC, and Java Naming and Directory Interface (JNDI). We offer solutions for recurring problems for the J2EE platform through the J2EE Pattern Catalog and J2EE refactorings. You can apply these ideas when developing new systems or when improving the design of existing systems. The patterns in this book will help you quickly gain the proficiency and skills to build robust, efficient enterprise applications.

Today, as in the past, many of us naively assume that *learning a technology* is synonymous with *learning to design* with the technology. Certainly learning the technology is an important part to being successful in designing with the technology. Many existing Java books are excellent at explaining technology details, such as API specifics and so forth, but at the same time they give no insight on applying the technology. Learning to design comes from experience and from sharing knowledge on best practices and bad practices.

The experiences we have conveyed in this book are derived from the work we have done in the field. We are part of Sun Microsystems, Inc.'s Sun Java Center (SJC) consulting organization. In our work, we often encounter situations where, because technology is moving so quickly, designers and developers are still struggling to understand the technology, let alone how to design with the technology.

It is not good enough to tell designers and developers to write good code, nor is it sufficient to suggest using Servlets and JSP for developing the presentation tier and EJB components[1] for developing the business tier.

[1] If you are new to the J2EE platform, we discuss the platform and these technologies in Chapter 2, "J2EE Platform Overview".

So, given this scenario, where does an aspiring J2EE architect learn not only what to do, but what not to do? What are the best practices? What are the bad practices? How do you go from problem to design to implementation?

Sun Java Center and the J2EE Pattern Catalog

Since its inception, SJC architects have been working with clients all over the world to successfully design, architect, build, and deploy various types of systems based on Java and J2EE. The SJC is a rapidly growing consulting organization constantly adding new hires to its ranks of experienced architects.

Recognizing the need to capture and share proven designs and architectures, we started to document our work on the J2EE platform in the form of patterns in 1999. Although we looked in the existing literature, we could not find a catalog of patterns that dealt specifically with the J2EE platform. We found many books dealing with one or more of the J2EE technologies, and these books do an excellent job of explaining the technology and unraveling the nuances of the specifications. Some books offered extra help by providing some design considerations.

Since we first publicly presented our ideas on J2EE patterns at the JavaOne Conference in June 2000, we have received an overwhelming response from architects and developers. While some individuals expressed great interest in learning more about the patterns, others confirmed that they had applied the patterns, but had never named or documented them. This interest in patterns for the J2EE platform further motivated us to continue our work.

Thus, we put together the J2EE Pattern Catalog., which was initially made available to the entire J2EE community in beta form via the Java Developer Connection in March, 2001. Based largely on community feedback, the beta documentation evolved into the release you see in this book.

We hope these patterns, best practices, strategies, bad practices, and refactorings for the J2EE platform, provide the same benefits to you as they do for us.

What This Book is About?

This book is about:

- *Using patterns for the J2EE Platform.*

  Based on our collective J2EE platform experience, we have assembled the pattern catalog in this book. The J2EE Pattern Catalog describes various best practices related to architecting and designing applications for the J2EE platform. This book focuses on the following four J2EE technologies: Servlets, JSP, EJB components, and JMS.

- *Using best practices to design applications that use JSP, Servlet, EJB components, and JMS technologies.*

It is not sufficient to merely learn the technology and the APIs. It is equally important to learn to design with the technology. We have documented what we have experienced to be the best practices for these technologies.

- *Preventing re-inventing-the-wheel when it comes to design and architecture for the J2EE platform.*

  Patterns promote design reuse. Reusing known solutions reduces the cycle time for designing and developing applications, including J2EE applications.

- *Identifying bad practices in existing designs and refactoring these designs to move to a better solution using the J2EE patterns.*

  Knowing what works well is good. Knowing what does not work is equally important. We have documented some of the bad practices we have experienced when designing applications for the J2EE platform.

What This Book Is Not?

This book is not about:

- *How to program with Java or J2EE technologies*

  This book is not about programming. While this book is heavily based on the J2EE technologies, we do not describe the specific APIs. If you wish to learn about programming using Java or using any of the J2EE technologies, there are a number of excellent books and online resources from which to learn. The online tutorials on the official Java home page at http://java.sun.com are highly recommended if you wish to learn about individual technologies. The official specifications for J2EE technologies are also available from the Java home page.

- *What process and methodology to use*

  We do not suggest any type of process or methodology to use since the material presented in this book is not related to either. Hence, this book does not teach you about a process or methodology to follow in your projects. If you would like to learn more about processes and methodologies, there are a good number of books that deal with various object-oriented methodologies and new books on lightweight processes, such as Extreme Programming.

- *How to use Unified Modeling Language (UML)*

This book is not going to teach you about UML. We use UML extensively (specifically class and sequence diagrams) to document the patterns and describe the static and dynamic interactions. If you want to learn more about UML, please refer to the UML User Guide [Booch] and the UML Reference Manual [Rumbaugh] by Grady Booch, Ivar Jacobson and James Rumbaugh.

Who Should Read this Book?

This book is for all J2EE enthusiasts, programmers, architects, developers, and technical managers. In short, anyone who is remotely interested in designing, architecting and developing applications for the J2EE platform.

We have attempted to distinguish this book as a training guide for J2EE architects and designers. We all recognize the importance of good designs and well-architected projects, and that we need good architects to get there.

The use of well-documented patterns, best practices, and bad practices to share and transfer knowledge and experience can prove invaluable for teams with varied experience levels, and we hope that this book answers some of these needs.

How This Book is Organized

This book is organized into three parts.

Part 1—"Patterns and J2EE", consists of Chapter 1 and Chapter 2.

Chapter 1: "Introduction" is a brief discussion on various topics, including patterns, J2EE platform, defining a pattern, and pattern categorization. It ends by introducing the J2EE Pattern Catalog.

Chapter 2 : "J2EE Platform Overview" provides a high level overview of the J2EE platform for those readers unfamiliar with J2EE, or who wish to refresh their knowledge of the J2EE platform.

Part 2—"Design Considerations, Bad Practices, and Refactorings" deals with design considerations for JSP, Servlets, and enterprise beans. This part also includes bad practices and refactorings for the J2EE platform. This part is comprised of Chapter 3, 4, and 5.

Chapter 3 "Presentation Tier Design Considerations and Bad Practices" and Chapter 4 "Business Tier Design Considerations and Bad Practices" discuss the design considerations and bad practices for the presentation tier and business/integration tiers respectively. The design considerations are issues that a J2EE developer/designer/architect needs to consider while working with the J2EE platform. The topics presented in these chapters point the reader to other sources (such as official specifications and well written books on these topics) for more detailed information on these issues.

Chapter 5: "J2EE Refactorings" includes some of the refactorings we have experienced in our work in the field that has enabled us to move our design from a less optimal solution to a better solution. The refactorings provide another way to think about the material in the rest of the book, providing what we believe to be valuable companion material to the pattern catalog. This chapter shows how we have been influenced by Martin Fowler and his book "Refactoring" [Fowler]. For those readers who are familiar with the Refactoring book, the format of this chapter will be very familiar. However, the content of this chapter is entirely in the context of J2EE technologies, whereas Martin Fowler addresses refactoring at a different level.

Part 3—"J2EE Pattern Catalog" presents the J2EE pattern catalog. The catalog contains the fifteen patterns that form the core of this book. This part is comprised of Chapter 6, 7, 8, and 9.

Chapter 6: "J2EE Patterns Overview" provides an overview of the J2EE pattern catalog. This chapter begins with a high level discussion of the pattern ideas and explains the way the patterns are categorized into tiers. It also explains the J2EE pattern template, which is used to present all patterns in this book. The chapter discusses all the J2EE patterns and uses a diagram to show their inter-relationships. It also provides what we have termed a roadmap to the pattern catalog. This roadmap presents common J2EE design and architecture-related questions with references to patterns or refactorings that provide solutions to these questions. Understanding the pattern relationships and the roadmap is key to using these patterns.

Chapter 7: "Presentation Tier Patterns" presents six patterns that pertain to using Servlets, JSP, JavaBeans, and custom tags to design web-based applications for the J2EE platform. The patterns describe numerous implementation strategies, and address common problems such as request handling, application partitioning, and generating composite displays.

Chapter 8: "Business Tier Patterns" presents seven patterns that pertain to using EJB technology to design business components for the J2EE platform. The patterns in this chapter provide the best practices for using the EJB and JMS technologies. Where relevant, these patterns include discussion on other technologies, such as JNDI and JDBC.

Chapter 9: "Integration Tier Patterns" presents two patterns that pertain to integrating J2EE applications with the resource tier and external systems. The patterns deal with using JDBC and JMS to enable integration between business tier and resource tier components.

Epilogue: “J2EE Patterns Applied” discusses realizing sample use cases with the patterns. This chapter discusses and demonstrates how patterns are combined and work together. This chapter reinforces the idea that patterns exist in a community, and that each pattern supports, and is supported by, other patterns.

Companion Website and Contact Information

The official companion website where we will provide updates and other material is http://www.phptr.com/corej2eepatterns.

The J2EE Patterns interest group, j2eepatterns-interest@java.sun.com is available for public subscription and participation. To subscribe to the interest group and review the discussion archives, please visit:

http://archives.java.sun.com/archives/j2eepatterns-interest.html