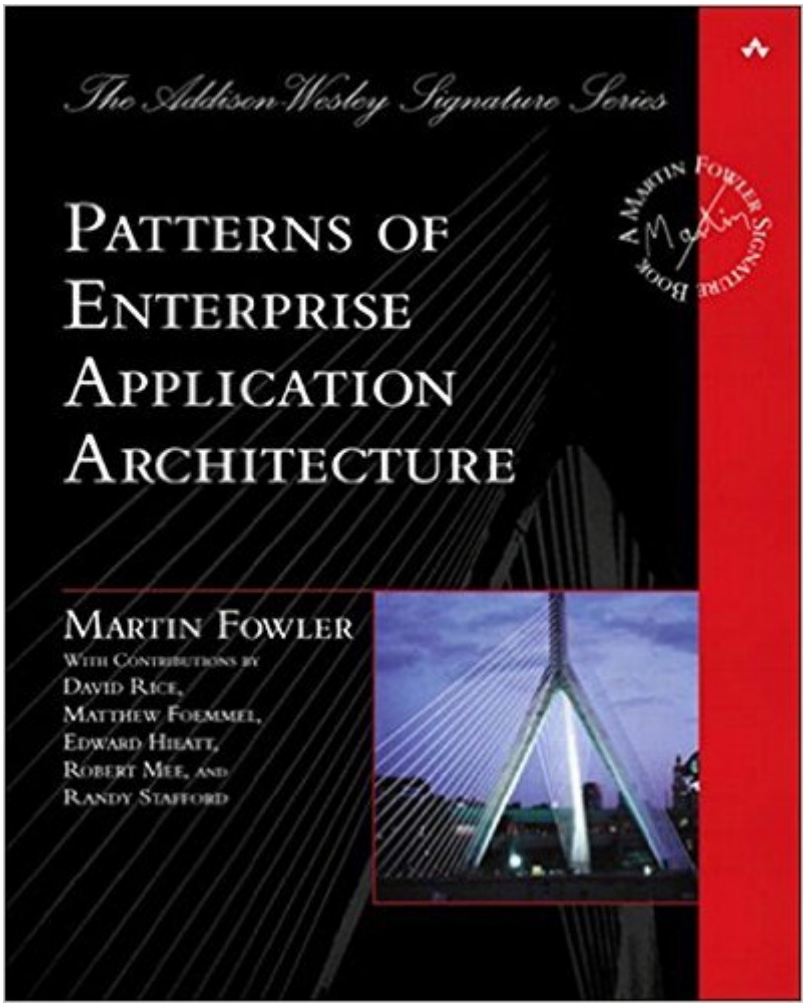
Patterns Of Enterprise Application Architecture (Addison-Wesley Signature Series (Fowler))



Synopsis

The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not easily implemented. Common failures in enterprise applications often occur because their developers do not understand the architectural lessons that experienced object developers have learned. Â Patterns of Enterprise Application Architecture is written in direct response to the stiff challenges that face enterprise application developers. The author, noted object-oriented designer Martin Fowler, noticed that despite changes in technology--from Smalltalk to CORBA to Java to .NET--the same basic design ideas can be adapted and applied to solve common problems. With the help of an expert group of contributors, Martin distills over forty recurring solutions into patterns. The result is an indispensable handbook of solutions that are applicable to any enterprise application platform. Â This book is actually two books in one. The first section is a short tutorial on developing enterprise applications, which you can read from start to finish to understand the scope of the book's lessons. The next section, the bulk of the book, is a detailed reference to the patterns themselves. Each pattern provides usage and implementation information, as well as detailed code examples in Java or C#. The entire book is also richly illustrated with UML diagrams to further explain the concepts. Armed with this book, you will have the knowledge necessary to make important architectural decisions about building an enterprise application and the proven patterns for use when building them. Â The topics covered include Â Â Dividing an enterprise application into layers Â Â The major approaches to organizing business logic Â Â An in-depth treatment of mapping between objects and relational databases Â Â Using Model-View-Controller to organize a Web presentation Â Â Handling concurrency for data that spans multiple transactions Â Â Designing distributed object interfaces

Book Information

File Size: 27645 KB

Print Length: 558 pages

Simultaneous Device Usage: Up to 5 simultaneous devices, per publisher limits

Publisher: Addison-Wesley Professional; 1 edition (March 9, 2012)

Publication Date: March 9, 2012

Sold by:Â Digital Services LLC

Language: English

ASIN: B008OHVDFM

Text-to-Speech: Enabled

X-Ray: Enabled

Word Wise: Not Enabled

Lending: Not Enabled

Enhanced Typesetting: Enabled

Best Sellers Rank: #46,135 Paid in Kindle Store (See Top 100 Paid in Kindle Store) #3 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Computer Design #15 inÂ Books > Computers & Technology > Hardware & DIY > Design & Architecture #25 inÂ Books > Computers & Technology > Hardware & DIY > Personal Computers

Customer Reviews

This is the best book I've found on J2EE and .Net patterns. I think it's destined to become a classic. I found the discussions on when to distrbute ('sell your favorite grandmother first'), Unit Of Work, Domain Model and Data Mapper patterns extremely useful. It has changed the way I think about enterprise applications.I think it fits somewhere between the original 'Design Patterns' book, by Gamma, et al, and a book like 'J2EE Patterns' in terms of its scope. 'Design Patterns' describes existing patterns that are applicable to any kind of application. 'J2EE Patterns' describes patterns in terms of one platform (although many of them apply to other platforms as well.) Fowler's book describes a set of patterns that work with a certain kind of application, business apps, but that are applicable to more than one platform.It's better than the 'J2EE Patterns' book, which doesn't do a good job explaining which parts of J2EE to avoid, and which 'patterns' are in fact workarounds for problems in the platform itself. (For example, the 'Composite Entity' pattern.)I have to strongly disagree with the first reviewer. Fowler does explain which patterns work best on which platform. The first section of the book gives a good road map for deciding which set of patterns to use for your app. He mentions explicitly that .Net pulls you in the direction of Table Module, but that with J2EE you would be less likely to use that pattern.As far as the patterns being available in frameworks, I still find it useful to know about the patterns the framework implements. That way you know which framework to select. We recently went through an O/R mapping tool selection process.

I am a fan of Fowler's and especially his "Refactoring" book, which I also rate as a must read for the serious programmer.Fowler's new book is an attempt to do for Enterprise Application Architecture what "Design Patterns" (i.e., GOF) did for OOP.Unfortunately,while it is an excellent book, there are

issues...1)First, Design Patterns is a very dense and scholarly read. It is also, frankly, a difficult read. However, after you have spent a couple of days trying to digest a pattern from Design Patterns, you realize, in many cases, you have had an experience with something profound. Even the GOF authors, in the preface, attempt to console readers by admitting "We didn't understand it all on the first writing!". Fowler's book, by contrast, is not on the same level, and can be understood on a first read. Perhaps this is what other reviewers were sensing when they indicated it was for the novice architect?2) Fowler does NOT address security. How then,does the word "Enterprise" get the priviledge of adorning the title of his book? Enterprise design should be secure design. But, this will usually require a trade off --- more secure, less performance...or less secure, more scaleable...Fowler does not consider this. Example: A chapter is devoted to the "Table Data GateWay" pattern. The gateway pattern might be OK for J2EE...but it is not the most secure, or the best for performance,in .Net... The problem is it constructs its SQL statements in line, rather than using stored procedures. This allows SQL insertion attacks if your coders are sloppy, and also does not take advantage of the precompiled nature of sprocs.3)There is a J2EE bias.

First off, let's start with some of the basics: This book is a classic. Like the "Gang of Four" book, "Design Patterns" this book forms something of a foundation of knowledge and lingo that software developers above a certain level are going to be expected to know and understand. If you're building a large system with experienced developers, you sure as heck better understand what things like Service Layer, Repository, Unit Of Work and Active Record are, to name just a few. This book, like all books about patterns, does more to serve as a foundation for communication, giving a common language for oft-repeated solutions. You can tell another member of your team "We're going to need a Domain Model here" and she will know what you are talking about in general, if not in specific. Or, you can name a class "WidgetPlugin" and people will understand what that class does (again, in general) without having to look at any code or read any documentation. In terms of ability to lift people up to this higher level, there are few books which compare.One complaint I do have (and because this book is from 2002 I don't take stars away) is that it seems to ignore some modern ideas about Dependency Inversion and testability. In 2015 we'd probably make a lot of design decisions around the ability to test classes. Testing usually involves injection of abstract dependencies, so many systems are designed with that in mind. Several of the patterns in this book explicitly go against this idea. Repository is one, and Active Record is another. In fact, his example code for Active Record uses a Repository internally as a static global to look up DB connectivity information.

Patterns of Enterprise Application Architecture (Addison-Wesley Signature Series (Fowler)) Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions (Addison-Wesley Signature Series (Fowler)) Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Adobe Reader) (Addison-Wesley Signature Series (Fowler)) More Agile Testing: Learning Journeys for the Whole Team (Addison-Wesley Signature Series (Cohn)) Enterprise Integration: An Architecture for Enterprise Application and Systems Integration Crochet: Easy Crochet Patterns: Crochet Patterns for Beginners (Crochet: Step by Step Crochet, Crochet Patterns, Easy Crochet Patterns, Crochet Patterns for Beginners, and Crochet Projects) Grand Theft Auto V Signature Series Strategy Guide: Updated and Expanded (Bradygames Signature Series) Final Fantasy XII Signature Series Guide (Bradygames Signature Guides) R for Everyone: Advanced Analytics and Graphics (Addison-Wesley Data & Analytics Series) First Principles of Discrete Systems and Digital Signal Processing (Addison-Wesley Series in Electrical Engineering) Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem (Addison-Wesley Data & Analytics Series) Apache Hadoop YARN: Moving beyond MapReduce and Batch Processing with Apache Hadoop 2 (Addison-Wesley Data & Analytics Series) TCP/IP Illustrated, Vol. 1: The Protocols (Addison-Wesley Professional Computing Series) TCP/IP Illustrated, Volume 1: The Protocols (Addison-Wesley Professional Computing Series) TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition) (Addison-Wesley Professional Computing Series) TCP/IP Illustrated, Vol. 2: The Implementation (Addison-Wesley Professional Computing Series) The Design and Implementation of the 4.4 BSD Operating System (Addison-Wesley UNIX and Open Systems Series) Advanced Programming in the UNIX Environment (Addison-Wesley Professional Computing Series) The Go Programming Language (Addison-Wesley Professional Computing Series) Advanced Programming in the UNIX(R) Environment (Addison-Wesley Professional Computing Series)