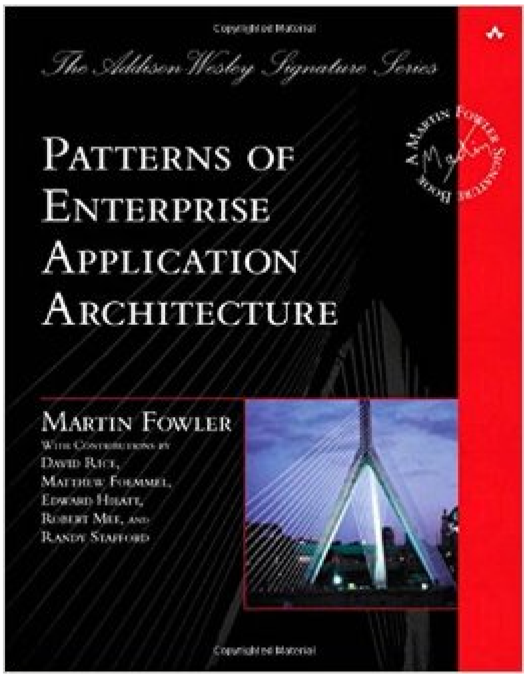
The book was found

Patterns Of Enterprise Application Architecture



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&#39;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

Hardcover: 560 pages

Publisher: Addison-Wesley Professional; 1 edition (November 15, 2002)

Language: English

ISBN-10: 0321127420

ISBN-13: 978-0321127426

Product Dimensions: 7.5 x 1.4 x 9.3 inches

Shipping Weight: 2.5 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 starsÂ Â See all reviewsÂ (112 customer reviews)
Best Sellers Rank: #43,288 in Books (See Top 100 in Books)   #1 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Computer Design   #15 inÂ Books > Computers & Technology > Hardware & DIY > Personal Computers > PCs   #18 inÂ Books > Computers & Technology > Hardware & DIY > Design & Architecture

Customer Reviews

I agree wholeheartedly with an above post which pointed out that the subject material is mostly known to the average enterprise developer. I am at best an average developer and found I'd already thought of much of this stuff myself.One thing I would like to add is that this book was still excellent reading and skimming through the patterns sparked my creative energies. I find that when I read through it, even if I ’know’ the patterns already, it helps me explore their organization and consequences.I was disappointed that I wasn't blown away with helpful new concepts, but quite happy with my purchase all the same. Buy this if you want a thorough guide to EAA and maybe some enjoyable afternoon reading.(The following was added about 2 months after the original review) After owning this book for awhile, I've found it more and more indispensible. My original review, above, mentions that few of the concepts seem new, however, now that I've read it more thoroughly and applied some of the concepts, I don't think that ’mind-blowing originality’ is what I should have been looking for.Fowler's ’Refactoring’ is another example of a great book without any stunningly original concepts. Like Refactoring, PEAA can serve as a great guide to page through when you're stuck on a project and need to review your options.

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. Reading the Unit Of Work, Data Mapper, Repository, Lazy Load and Identity Map chapters helped *immensely* in that process. Likewise reading the Front Controller pattern gave me some new ideas on how best to utilize the Struts framework. I totally disagree with the notion that "learning about the patterns that are associated with these frameworks will provide little value". Ignorance is definitely not bliss here.Finally, the idea that because the book ’just’ collects and names patterns that already exist somehow decreases its value is hogwash. These are tried and true patterns that many developers have found useful. Naming and clearly describing common patterns is very helpful. This is exactly what the original ’Design Patterns’ book did. By this logic, I guess the original reviewer would have given ’Design Patterns’ only 3 stars.It’s a great book.

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. This probably is a good thing as the J2EE architectures tend to be more mature and contain good ideas...but you should be aware of it if you are a .NET programmer. Not all of the patterns will be immediately useful to .NET, but will require a careful implementation.4)Fowler passes on some of the tough questions. On page 93, while discussing RPC calls versus XML based messaging, he says "my preference is for a message - based approach that’s inherently asynchronous." I agree. (RPC style is not the really suited for async, whereas document style messaging is) He then says it is too large of a topic, so does not cover it. Well, if it is the best way, it needs to be covered in an Enterprise level book, no?While I may seem to be critical of Fowler’s book, I really think it is excellent with fine ideas. But, especially if you are from .Net, implement the patterns with care, and then only you have checked against Microsoft best practices. This is not to say Microsoft is always right, but get the second opinion anyway.

Download to continue reading...

Patterns of Enterprise Application Architecture Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 practices Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML Enterprise Java Developer’s Guide with CDROM (McGraw Hill Enterprise Computing) Houses Coloring Book: An Adult Coloring Book of 40 Architecture and House Designs with Henna, Paisley and Mandala Style Patterns (Architecture Coloring Books) (Volume 1) Safari Animal Patterns: 30 Exotic Safari Animal Patterns to Feel the Wildlife World (Safari Animal Patterns, animal designs, zendoodle) Oracle ADF Enterprise Application Development - Made Simple : Second Edition ASP.NET Core Application Development: Building an application in four sprints (Developer Reference) Adobe ColdFusion 9 Web Application Construction Kit, Volume 2: Application Development Adobe ColdFusion 8 Web Application Construction Kit, Volume 3: Advanced Application Development Wiley GAAP: Interpretation and Application of Generally Accepted Accounting Principles 2011 (Wiley GAAP: Interpretation & Application of Generally Accepted Accounting Principles) Girls Life Application Study Bible NLT (Kid’s Life Application Bible) The Handbook of Bible Application (Life Application Reference) An Introduction To Enterprise Architecture: Third Edition The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise Modeling Enterprise Architecture with TOGAF: A Practical Guide Using UML and BPMN (The MK/OMG Press) Enterprise Architecture As Strategy: Creating a Foundation for Business Execution Developer’s Guide to Microsoft Enterprise Library, C# Edition (Patterns & Practices) Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions Enterprise Model Patterns: Describing the World (UML Version)