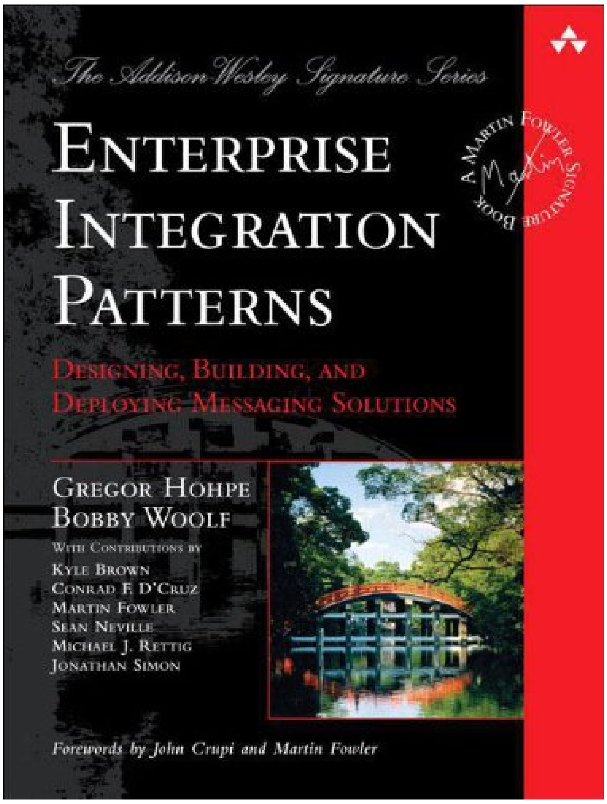
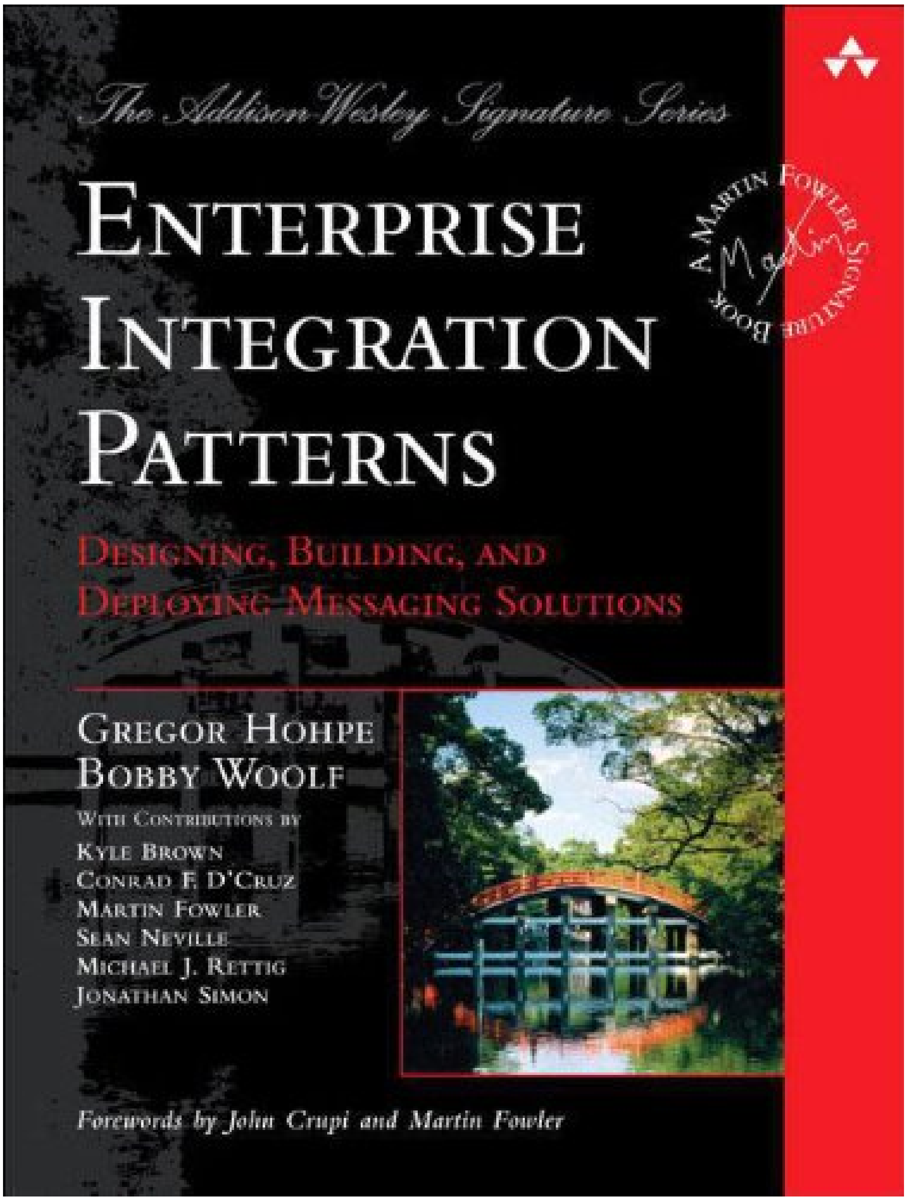
ENTERPRISE INTEGRATION PATTERNS: DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS BY GREGOR HOHPE, BOBBY WOOLF


The Addison Wesley Signature Series
ENTERPRISE INTEGRATION PATTERNS
DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS
A MARTIN FOWLER SIGNATURE BOOK
GREGOR HOHPE
BOBBY WOOLF
WITH CONTRIBUTIONS BY
KYLE BROWN
CONRAD F. D'CRUZ
MARTIN FOWLER
SEAN NEVILLE
MICHAEL J. RETTIG
JONATHAN SIMON
Forewords by John Crupi and Martin Fowler


The Addison Wesley Signature Series
ENTERPRISE INTEGRATION PATTERNS
DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS
GREGOR HOHPE
BOBBY WOOLF
WITH CONTRIBUTIONS BY
KYLE BROWN
CONRAD F. D'CRUZ
MARTIN FOWLER
SEAN NEVILLE
MICHAEL J. RETTIG
JONATHAN SIMON
Forewords by John Crupi and Martin Fowler
A MARTIN FOWLER SIGNATURE BOOK

ENTERPRISE INTEGRATION PATTERNS: DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS BY GREGOR HOHPE, BOBBY WOOLF PDF

The presented book Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf our company offer below is not sort of common book. You recognize, reviewing currently doesn't suggest to manage the published book Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf in your hand. You could obtain the soft documents of Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf in your gadget. Well, we suggest that guide that we proffer is the soft documents of guide Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf The material and all things are exact same. The difference is just the kinds of guide Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf, whereas, this problem will specifically be profitable.

From the Back Cover

How to manage and monitor a messaging system once it's in use as part of the enterprise. What This Book Does Not Cover We believe that any book sporting the word "enterprise" in the title is likely to fall into one of three categories. First, the book might attempt to cover the whole breadth of the subject matter but is forced to stop short of detailed guidance on how to implement actual solutions. Second, the book might provide specific hands-on guidance on the development of actual solutions but is forced to constrain the scope of the subject area it addresses. Third, the book might attempt to do both but is likely never to be finished or else to be published so late as to be irrelevant. We opted for the second choice and hopefully created a book that helps people create better integration solutions even though we had to limit the scope of the book. Topics that we would have loved to discuss but had to exclude in order not to fall into the category-three trap include security, complex data mapping, workflow, rule engines, scalability and robustness, and distributed transaction processing (XA, Tuxedo, and the like). We chose asynchronous messaging as the emphasis for this book because it is full of interesting design issues and trade-offs, and provides a clean abstraction from the many implementations provided by various integration vendors.This book is also not a tutorial on a specific messaging or middleware technology. To highlight the wide applicability of the concepts presented in this book, we included examples based on a number of different technologies, such as JMS, MSMQ, TIBCO, BizTalk, and XSL. However, we focus on the design decisions and trade-offs as opposed to the specifics of the tool. If you are interested in learning more about any of these specific technologies, please refer to one of the books referenced in the bibliography or to one of the many online resources.How This Book Is Organized As the title suggests, the majority of this book consists of a collection of patterns. Patterns are a proven way to capture experts' knowledge in fields where there are no simple "one size fits all" answers, such as application architecture, object-oriented design, or integration solutions based on asynchronous messaging architectures.Each pattern poses a specific design problem,

discusses the considerations surrounding the problem, and presents an elegant solution that balances the various forces or drivers. In most cases, the solution is not the first approach that comes to mind, but one that has evolved through actual use over time. As a result, each pattern incorporates the experience base that senior integration developers and architects have gained by repeatedly building solutions and learning from their mistakes. This implies that we did not "invent" the patterns in this book; patterns are not invented, but rather discovered and observed from actual practice in the field. Because patterns are harvested from practitioners' actual use, chances are that if you have been working with enterprise integration tools and asynchronous messaging architectures for some time, many of the patterns in this book will seem familiar to you. Yet, even if you already recognize most of these patterns, there is still value in reviewing this book. This book should validate your hard-earned understanding of how to use messaging while documenting details of the solutions and relationships between them of which you might not have been aware. It also gives you a consolidated reference to help you pass your knowledge effectively to less-experienced colleagues. Finally, the pattern names give you a common vocabulary to efficiently discuss integration design alternatives with your peers.The patterns in this book apply to a variety of programming languages and platforms. This means that a pattern is not a cut-and-paste snippet of code, but you have to realize a pattern to your specific environment. To make this translation easier, we added a variety of examples that show different ways of implementing patterns using popular technologies such as JMS, MSMQ, TIBCO, BizTalk, XSL, and others. We also included a few larger examples to demonstrate how multiple patterns play together to form a cohesive solution.Integrating multiple applications using an asynchronous messaging architecture is a challenging and interesting field. We hope you enjoy reading this book as much as we did writing it.About the Cover Picture The common theme for books in the Martin Fowler Signature Series is a picture of a bridge. In some sense we lucked out, because what theme would make a better match for a book on integration? For thousands of years, bridges have helped connect people from different shores, mountains, and sides of the road.We selected a picture of the Taiko-bashi Bridge at the Sumiyoshi-taisha Shrine in Osaka, Japan, for its simple elegance and beauty. As a Shinto shrine dedicated to the guardian deity for sailors, it was originally erected next to the water. Interestingly, land reclamation has pushed the water away so that the shrine today stands almost three miles inland. Some three million people visit this shrine at the beginning of a new year.Gregor Hohpe
San Francisco, CaliforniaBobby Woolf
Raleigh, North CarolinaSeptember 2003
www.enterpriseintegrationpatterns.com

0321200683P10062003

ENTERPRISE INTEGRATION PATTERNS: DESIGNING, BUILDING, AND DEPLOYING MESSAGING SOLUTIONS BY GREGOR HOHPE, BOBBY WOOLF PDF

Make use of the innovative modern technology that human establishes this day to discover guide **Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf** quickly. However initially, we will certainly ask you, just how much do you enjoy to read a book Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf Does it consistently till finish? For what does that book read? Well, if you actually like reading, aim to review the Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf as one of your reading collection. If you just reviewed guide based upon demand at the time as well as unfinished, you have to attempt to such as reading Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf initially.There is without a doubt that book *Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf* will constantly make you inspirations. Even this is just a book Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf; you can locate several styles and sorts of publications. From delighting to adventure to politic, and sciences are all provided. As exactly what we state, right here we provide those all, from popular writers and author in the world. This Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf is among the compilations. Are you interested? Take it now. Just how is the way? Read more this article!When someone should visit the book establishments, search shop by shop, shelf by shelf, it is very frustrating. This is why we give guide collections in this website. It will relieve you to look guide Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf as you such as. By looking the title, publisher, or authors of the book you really want, you could find them promptly. In your home, office, and even in your way can be all best area within internet links. If you want to download and install the Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf, it is extremely easy then, considering that currently we extend the connect to purchase and also make offers to download Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf So very easy!

How to manage and monitor a messaging system once it's in use as part of the enterprise. What This Book Does Not Cover We believe that any book sporting the word "enterprise" in the title is likely to fall into one of three categories. First, the book might attempt to cover the whole breadth of the subject matter but is forced to stop short of detailed guidance on how to implement actual solutions. Second, the book might provide specific hands-on guidance on the development of actual solutions but is forced to constrain the scope of the subject area it addresses. Third, the book might attempt to do both but is likely never to be finished or else to be published so late as to be irrelevant. We opted for the second choice and hopefully created a book that helps people create better integration solutions even though we had to limit the scope of the book. Topics that we would have loved to discuss but had to exclude in order not to fall into the category-three trap include security, complex data mapping, workflow, rule engines, scalability and robustness, and distributed transaction processing (XA, Tuxedo, and the like). We chose asynchronous messaging as the emphasis for this book because it is full of interesting design issues and trade-offs, and provides a clean abstraction from the many implementations provided by various integration vendors.This book is also not a tutorial on a specific messaging or middleware technology. To highlight the wide applicability of the concepts presented in this book, we included examples based on a number of different technologies, such as JMS, MSMQ, TIBCO, BizTalk, and XSL. However, we focus on the design decisions and trade-offs as opposed to the specifics of the tool. If you are interested in learning more about any of these specific technologies, please refer to one of the books referenced in the bibliography or to one of the many online resources.How This Book Is Organized As the title suggests, the majority of this book consists of a collection of patterns. Patterns are a proven way to capture experts' knowledge in fields where there are no simple "one size fits all" answers, such as application architecture, object-oriented design, or integration solutions based on asynchronous messaging architectures.Each pattern poses a specific design problem, discusses the considerations surrounding the problem, and presents an elegant solution that balances the various forces or drivers. In most cases, the solution is not the first approach that comes to mind, but one that has evolved through actual use over time. As a result, each pattern incorporates the experience base that senior integration developers and architects have gained by repeatedly building solutions and learning from their mistakes. This implies that we did not "invent" the patterns in this book; patterns are not invented, but rather discovered and observed from actual practice in the field. Because patterns are harvested from practitioners' actual use, chances are that if you have been working with enterprise integration tools and asynchronous messaging architectures for some time, many of the patterns in this book will seem familiar to you. Yet, even if you already recognize most of these patterns, there is still value in reviewing this book. This book should validate your hard-earned understanding of how to use messaging while documenting details of the solutions and relationships between them of which you might not have been aware. It also gives you a consolidated reference to help you pass your knowledge effectively to less-experienced colleagues. Finally, the pattern names give you a common vocabulary to efficiently discuss integration design alternatives with your peers.The patterns in this book apply to a variety of programming languages and platforms. This means that a pattern is not a cut-and-paste snippet of code, but you have to realize a pattern to your specific environment. To make this translation easier, we added a variety of examples that show different ways of implementing patterns using popular technologies such as JMS, MSMQ, TIBCO, BizTalk, XSL, and others. We also included a few larger examples to demonstrate how multiple patterns play together to form a cohesive solution.Integrating multiple applications using an asynchronous messaging architecture is a challenging and interesting field. We hope you enjoy reading this book as much as we did writing it.About the Cover Picture The common theme for books in the Martin Fowler Signature Series is a picture of a bridge. In some sense we lucked out, because what theme would make a better match for a book on integration? For thousands of years, bridges have helped connect people from different shores, mountains, and sides of the road.We selected a picture of the Taiko-bashi Bridge at the Sumiyoshi-taisha Shrine in Osaka, Japan, for its simple elegance and beauty. As a Shinto shrine dedicated to the guardian deity for sailors, it was originally erected next to the water. Interestingly, land reclamation has pushed the water away so that the shrine today stands almost three miles inland. Some three million people visit this shrine at the

beginning of a new year.Gregor Hohpe
San Francisco, CaliforniaBobby Woolf
Raleigh, North CarolinaSeptember 2003
www.enterpriseintegrationpatterns.com

0321200683P10062003 Most helpful customer reviews111 of 118 people found the following review helpful.
Patterns - revisited
By Amazon Customer
To do justice in reviewing this book, I should depict every single pattern and give you multiple examples on how it would apply to your job as a Project Manager, Software Architect, Technical Lead or a Developer. That would be a 500-page book all by itself. In short, this is one great book. The first book to actually take a complex and ever growing topic such as MOM, Message Oriented Middleware, and give you its benefits and the best practices/patterns all in one book.
The author starts by giving the reader the top reasons why messaging should be chosen for the next project:
1) Remote communication
2) Platform/Language Integration
3) Asynchronous communication
4) Variable timing
5) Throttling
6) Reliable Communication
7) Disconnected operation
8) Mediation
9) Thread Management
The author goes into detail on each of these reasons. These reasons would convince any software architect, but the author goes even further than that and reiterates the benefits of each of these reasons and elaborates on them thru out the book.
Chapter 3 of the book starts by breaking up a messaging system into its main components and briefly explaining each one:
1) Message Channel
2) Message
3) Pipes and Filers
4) Message Router
5) Message Translator
6) Message Endpoint
Each of these high level topics is then broken down and various patterns are shown for each section. Just like the GoF book, the reader can simply go the desired section and read the patterns that are associated with that "subsystem"
Each section is then followed by a full-blown example, which to me is priceless. The examples are shown using the most popular middleware vendors such as TIBCO, IBM, Microsoft, Web Methods, SeeBeyond and a couple JMS vendors. The examples show the similarities and differences in implementation but clearly show how EACH pattern that was just covered in the previous section applies to the example.
Having worked with many of the MOM vendors covered in this book, Chapter 7, Message Routing, is my favorite chapter. The author breaks down this topic into 14 different patterns:
i) Pipes and Filers
ii) Message Router
iii) Content-Based router
iv) Message Filter
v) Dynamic Router

vi) Recipient List
vii) Splitter
viii) Aggregator
ix) Resequencer
x) Composed Message Processor
xi) Scatter-Gather
xii) Routing Slip
xiii) Process Manager
xiv) Message Broker

The chances are, not many of us need to write a MOM due to the fact that there are many vendors out there that are doing that already! But one could certainly use this section for education purposes, and/or use it a checklist of "nice-to-haves" when shopping around for a MOM vendor. By reading the book, you can figure out what "features" apply to you, your application and your enterprise, and take that list and see which vendor has implemented that feature.

In summary, Gregor Hohpe and Bobby Woolf have done a fantastic job depicting a very complex topic. I have made a place for this book right next to the original GoF Design Patterns book.43 of 45 people found the following review helpful.

The best technical book of 2004

By David Bridgeland

I had been waiting for this book for several years. There are many good books on software architecture using synchronous communication, but nothing on asynchronous communication --- the typical scheme when connecting existing applications. This is surprising since the underlying products (MQ, MSMQ, WebMethods, Vitria, etc.) have been around for a while, some for more than 10 years, and the techniques have become increasingly well understood by the practitioners. There are even some books on the individual products --- several on MQ for example --- but nothing more general about how to use messaging, message routing, and message transformation to build a larger system.

This is the book I had been waiting for. Furthermore the authors have avoided the usual three pitfalls of technical books: it is well organized, it well written, and it is deep treatment, not at all superficial.

The book is organized into 65 patterns (in the manner of the classic _Design Patterns_). Each pattern shows one typical problem in integrating applications, and how it is solved. Each pattern gives enough implementation details so it is clear how it would work, and an example or two so it is clear how it works in practice. For example the Message Expiration pattern addresses the problem of "How can a sender of a message indicate when a message should be considered stale and thus shouldn't be processed?"

The writing in this book is clear. For example "A Message Expiration is like the expiration date on a milk carton. After that date, you shouldn't drink the milk." The authors have also invented icons for each of their patterns. Their icon language allows a integration architecture to be visuallized in a way that UML does not provide.

Amongst the 11 pattern-describing chapters are 3 "interludes", chapter-length examples that explain a problem, show how patterns can combined to solve it, and then provide implementations in different technologies (JMS, .Net, TIBCO, MSMQ, etc.).

My only beef with this book is that it is long and dense: almost 700 pages. I bought it in late December 2003 and I am only finishing it now. But it is hard to say what should have been cut. Certainly none of the patterns are unnecessary, and the decription of each feels like about the right length. The interludes are also useful for seeing how the patterns fit together. So maybe this book just needs to be 700 pages.27 of 28 people found the following review helpful.

Great Message Pattern Language

By Ray Ye

This a book about enterprise integration solutions, authors claim that they are technology neutral, it is true. In the examples and implementations, they chose 3 most popular messaging frameworks to illustrate the patterns. However, they are pretty biased toward messaging as the "better" solution to enterprise integration strategy. It may have a lot of edges over the other approaches, sometimes it is just easy to use a simple wrapper/facade to do the integration. But I guess authors really intend to push their messaging solutions as the subtitle indicates.

Having said that, this is an excellent book of message pattern language, which I believe is the first one introducing the interesting topic. The books touches from the architectural patterns, e.g., messaging bus, pipe and filters, to common design patterns, e.g., publish/subscribe, request/reply, to some patterns that most MOMs provide as integrated solutions, e.g., durable subscriber, message filter, message expiration etc. With all these patterns at hand, a system architect would be able to craft a messaging pattern-oriented enterprise integration architecture by applying the appropriate patterns compositely.

The book would be better if authors describe some patterns implementation in more detail. E.g., it would be interesting to see how the message expiration is implemented, does the message contain a timer or the message channel monitor each individual message from start up? How does the channel interact with the message and check the expiry? Guaranteed delivery is another example. I know most of these implementation details only interest MOM developers, whereas pattern users are only interested in how and when to apply the patterns, but now that the book is about patterns themselves, implementation details would be appreciated.

Since all the patterns introduced in the book form a messaging pattern language, knowing each pattern's strength and limitation under the context, scope and different forces, and how it interacts with other patterns to form a bigger(composite) pattern are essential to grasp the pattern language. A collaboration diagram to show each pattern's transition/migration/composition to each other would be helpful.See all 71 customer reviews...

How to manage and monitor a messaging system once it's in use as part of the enterprise. What This Book Does Not Cover We believe that any book sporting the word "enterprise" in the title is likely to fall into one of three categories. First, the book might attempt to cover the whole breadth of the subject matter but is forced to stop short of detailed guidance on how to implement actual solutions. Second, the book might provide specific hands-on guidance on the development of actual solutions but is forced to constrain the scope of the subject area it addresses. Third, the book might attempt to do both but is likely never to be finished or else to be published so late as to be irrelevant. We opted for the second choice and hopefully created a book that helps people create better integration solutions even though we had to limit the scope of the book. Topics that we would have loved to discuss but had to exclude in order not to fall into the category-three trap include security, complex data mapping, workflow, rule engines, scalability and robustness, and distributed transaction processing (XA, Tuxedo, and the like). We chose asynchronous messaging as the emphasis for this book because it is full of interesting design issues and trade-offs, and provides a clean abstraction from the many implementations provided by various integration vendors.This book is also not a tutorial on a specific messaging or middleware technology. To highlight the wide applicability of the concepts presented in this book, we included examples based on a number of different technologies, such as JMS, MSMQ, TIBCO, BizTalk, and XSL. However, we focus on the design decisions and trade-offs as opposed to the specifics of the tool. If you are interested in learning more about any of these specific technologies, please refer to one of the books referenced in the bibliography or to one of the many online resources.How This Book Is Organized As the title suggests, the majority of this book consists of a collection of patterns. Patterns are a proven way to capture experts' knowledge in fields where there are no simple "one size fits all" answers, such as application architecture, object-oriented design, or integration solutions based on asynchronous messaging architectures.Each pattern poses a specific design problem, discusses the considerations surrounding the problem, and presents an elegant solution that balances the various forces or drivers. In most cases, the solution is not the first approach that comes to mind, but one that has evolved through actual use over time. As a result, each pattern incorporates the experience base that senior integration developers and architects have gained by repeatedly building solutions and learning from their mistakes. This implies that we did not "invent" the patterns in this book; patterns are not invented, but rather discovered and observed from actual practice in the field. Because patterns are harvested from practitioners' actual use, chances are that if you have been working with enterprise integration tools and asynchronous messaging architectures for some time, many of the patterns in this book will seem familiar to you. Yet, even if you already recognize most of these patterns, there is still value in reviewing this book. This book should validate your hard-earned understanding of how to use messaging while documenting details of the solutions and relationships between them of which you might not have been aware. It also gives you a consolidated reference to help you pass your knowledge effectively to less-experienced colleagues. Finally, the pattern names give you a common vocabulary to efficiently discuss integration design alternatives with your peers.The patterns in this book apply to a variety of programming languages and platforms. This means that a pattern is not a cut-and-paste snippet of code, but you have to realize a pattern to your specific environment. To make this translation easier, we added a variety of examples that show different ways of implementing patterns using popular technologies such as JMS, MSMQ, TIBCO, BizTalk, XSL, and others. We also included a few larger examples to demonstrate how multiple patterns play together to form a cohesive solution.Integrating multiple applications using an asynchronous messaging architecture is a challenging and interesting field. We hope you enjoy reading this book as much as we did writing it.About the Cover Picture The common theme for books in the Martin Fowler Signature Series is a picture of a bridge. In some sense we lucked out, because what theme would make a better match for a book on integration? For thousands of years, bridges have helped connect people from different shores, mountains, and sides of the road.We selected a picture of the Taiko-bashi Bridge at the Sumiyoshi-taisha Shrine in Osaka, Japan, for its simple elegance and beauty. As a Shinto shrine dedicated to the guardian deity for sailors, it was originally erected next to the water. Interestingly, land reclamation has pushed the water away so that the shrine today stands almost three miles inland. Some three million people visit this shrine at the

beginning of a new year.Gregor Hohpe
San Francisco, CaliforniaBobby Woolf
Raleigh, North CarolinaSeptember 2003
www.enterpriseintegrationpatterns.com

0321200683P10062003 The presented book Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf our company offer below is not sort of common book. You recognize, reviewing currently doesn't suggest to manage the published book Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf in your hand. You could obtain the soft documents of Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf in your gadget. Well, we suggest that guide that we proffer is the soft documents of guide Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf The material and all things are exact same. The difference is just the kinds of guide Enterprise Integration Patterns: Designing, Building, And Deploying Messaging Solutions By Gregor Hohpe, Bobby Woolf, whereas, this problem will specifically be profitable.