Enterprise Integration Patterns

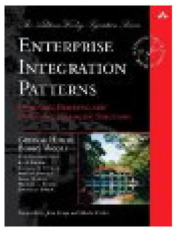
Home

Patterns and Best Practices for Enterprise Integration

This site is dedicated to making the design and implementation of integration solutions easier. The solutions described here are relevant for a variety of integration tools and platforms such as IBM WebSphere MQ, TIBCO, Vitria, WebMethods (Software AG), or Microsoft BizTalk, messaging systems such as JMS, WCF, Rabbit MQ, or MSMQ, ESB's such as Apache Camel, Mule, WSO2, Oracle Service Bus, Open ESB, SonicMQ, Fiorano or Fuse ServiceMix plus other SOA and Web-service solutions.

All content on this site is original and is maintained by Gregor Hohpe. I have been building integration solutions for large clients for many years and enjoy sharing my findings with the community. You can find the most recent content in my Ramblings. Please contact me if you have suggestions or feedback.

Enterprise Integration Patterns - The Book



Enterprise Integration Patterns

Gregor Hohpe, Bobb...

Best Price $24.03

or Buy New $49.30

Buy from amazon.com

Privacy Information

Enterprise integration remains harder than it should be. While integrating systems is inherently complex, I felt that one of the major stumbling blocks is the lack of a common vocabulary and body of knowledge around asynchronous messaging architectures, which are widely used to implement integration solutions. Under the guidance of Martin Fowler and Kyle Brown, Bobby Woolf and I documented such a language in the form of 65 integration patterns (see catalog on the right).

The book *Enterprise Integration Patterns* provides a consistent vocabulary and visual notation to design and document integration solutions independent from implementation technologies. It also explores the advantages and "gotchas" of asynchronous messaging architectures. You can learn how to design code that connects an application to a messaging system, how to route messages to the proper destination and how to monitor the health of a messaging system. The patterns in the book come to life with examples implemented in different messaging technologies, such as SOAP, JMS, MSMQ, .NET, TIBCO and other EAI Tools. See for yourself by Downloading a sample chapter!

Ramblings

My ongoing thoughts about the present and future of integration, SOA and Web services. [see all]

37 Things or "Where have all my ramblings gone?"

25 Years of OOP

How to Scale an Organization? The same way you scale a system!

Upcoming Events

IT-Architekten: Angelpunkt der Digitalen Transformation im Unternehmen
Sept 21-22, Oct 19-20, 2016
Frankfurt, Hamburg

Software Architecture Summit
Sept 26, 2016
Berlin

YOW!
Dec 1-9, 2016
Sydney, Melbourne, Brisbane

Articles & Interviews

20 Years of Patterns' Impact
(IEEE Software)

Conversations Between Loosely Coupled Services
(Video on InfoQ)

Developing in a Service-oriented World
(Video on InfoQ)

SOA Patterns - New Insights or Recycled Knowledge?
(Whitepaper)

Let's Have a Conversation
(IEEE Internet Computing)

Messaging Patterns

Table of Contents
Preface
Introduction
Solving Integration Problems using Patterns

Integration Styles
File Transfer
Shared Database
Remote Procedure Invocation
Messaging

Messaging Systems
Message Channel
Message
Pipes and Filters
Message Router
Message Translator
Message Endpoint

Messaging Channels
Point-to-Point Channel
Publish-Subscribe Channel
Datatype Channel
Invalid Message Channel
Dead Letter Channel
Guaranteed Delivery
Channel Adapter
Messaging Bridge
Message Bus

Message Construction
Command Message
Document Message
Event Message
Request-Reply
Return Address
Correlation Identifier
Message Sequence
Message Expiration
Format Indicator

Interlude: Simple Messaging
JMS Request/Reply Example
.NET Request/Reply Example
JMS Publish/Subscribe Example

Message Routing
Content-Based Router
Message Filter

Programming Without a Call Stack - Event-driven Architectures
(ObjektSpektrum, 2006)

Your Coffee Shop Does Not Use Two-Phase Commit
(IEEE Software)

Developing in a Service-Oriented World
(ThoughtWorks Whitepaper)

Integration Patterns with BizTalk Server 2004
(ThoughtWorks Whitepaper)

An Asynchronous World
(Software Development)

Test-Driven Development in Integration Projects
(ThoughtWorks Whitepaper)

XML Abuse
(Software Development)

News

"Starbucks Does Not Use Two-Phase Commit" selected for "The Best Software Writing"

Presentation Downloads

Nearfield Communication (NFC) in Japan
(OOP 2012)

Embedded OSS
(OOP 2012)

Distributed Programming the Google Way
(Devoxx 2009)

Developing in a Service-Oriented World
(SOACon 2007)

Programming without a Call Stack: Event-driven Architectures
(SOACon 2007)

Software Visualization and Model Extraction
(TheServerSide)

Conversations Between Loosely Coupled Systems
(SD West, 2006)

"The core language of EAI, defined in the book Enterprise Integration Patterns *by Gregor Hohpe and Bobby Woolf, is also the core language of defining ESB flows and orchestrations, as seen in the ESB's developer tooling."*
--Forrester Research

"If you are involved with the operation or development of an enterprise application, there will doubtless come a time when you will need to integrate your application with another using the emerging preferred approach of messaging. When that time comes, this book will be your most valuable reference."
--Randy Stafford, Oracle [More Testimonials]

Why Do We Need Integration?

Today's business applications rarely live in isolation. Users expect instant access to all business functions an enterprise can offer, regardless of which system the functionality may reside in. This requires disparate applications to be connected into a larger, integrated solution, which is generally achieved through some form of "middleware". Middleware provides the "plumbing" such as data transport, data transformation, and routing.

KPI Dashboard Free Trial
Monitor, Drill Down & Analyze. Create a KPI Dashboard Today!

What Makes Integration so Hard?

Architecting integration solutions is a complex task. There are many conflicting drivers and even more possible 'right' solutions. Whether the architecture was in fact a good choice usually is not known until many months or even years later, when inevitable changes and additions put the original architecture to test. Unfortunately, there is no "cookbook" for enterprise integration solutions. Most integration vendors provide methodologies and best practices, but these instructions tend to be very much geared towards the vendor-provided tool set and often lack treatment of the bigger picture, including underlying guidelines, principles and best practices.

Asynchronous Messaging Architectures

Asynchronous messaging architectures have proven to be the best strategy for enterprise integration because they allow for a loosely coupled solution that overcomes the limitations of remote communication, such as latency and unreliability. That's why most EAI suites and ESB's are based on asynchronous messaging. Unfortunately, asynchronous messaging is not without pitfalls. Many of the assumptions that hold true when developing single, synchronous applications are no longer valid. Vendor-independent design guidance helps developers avoid these pitfalls so they can build robust integration architectures based on asynchronous messaging.

Dynamic Router
Recipient List
Splitter
Aggregator
Resequencer
Composed Msg. Processor
Scatter-Gather
Routing Slip
Process Manager
Message Broker

Message Transformation

Envelope Wrapper
Content Enricher
Content Filter
Claim Check
Normalizer
Canonical Data Model

Interlude: Composed Messaging

Synchronous (Web Services)
Asynchronous (MSMQ)
Asynchronous (TIBCO)

Messaging Endpoints

Messaging Gateway
Messaging Mapper
Transactional Client
Polling Consumer
Event-Driven Consumer
Competing Consumers
Message Dispatcher
Selective Consumer
Durable Subscriber
Idempotent Receiver
Service Activator

System Management

Control Bus
Detour
Wire Tap
Message History
Message Store
Smart Proxy
Test Message
Channel Purger

Interlude: Systems Management Example

Instrumenting Loan Broker

Integration Patterns in Practice

Case Study: Bond Trading System

Concluding Remarks

Emerging Standards

Appendices

Bibliography
Revision History

Enterprise Integration Patterns (JAOO, 2003)

How can Patterns Help?

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 message-oriented integration . Each pattern tackles a specific problem by discussing design considerations and presenting an elegant solution that balances the forces. Often 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 that senior developers and architects have gained by repeatedly building solutions and learning from their mistakes. Thus, patterns are not "invented" but discovered and observed from actual practice.

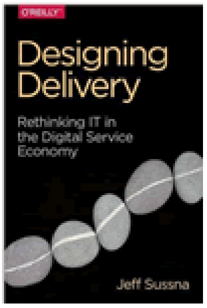
The patterns on this site cannot cover all aspects of integration. We focused on developing a cohesive set of patterns that would make a well rounded book. We continue to discover new patterns and plan to document them in the future.

What am I Reading Right Now?

37 Things One Architect Knows
A Chief IT Architect's Journey
Gregor Hohpe

NEW 37 Things One Architect Knows, Hohpe, Leanpub, to be published Summer 2016 (now 95%)

Transforming large-scale IT organizations and architecture requires superb technical, communication, and organizational skill. Chief architects or CTOs must play a key role in such transformations, but hardly any material caters to them. After orchestrating IT transformation as Chief Architect for 3 years, I decided to publish my own book DRM-free e-book (PDF, ePub, MOBI) on LeanPub.

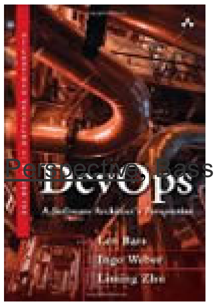

O'REILLY
Designing Delivery
Rethinking IT in the Digital Service Economy
Jeff Sussna

Designing Delivery: Rethinking IT in the Digital Service Economy, Sussna, O'Reilly 2015

This book strikes a great balance between drawing on theoretical foundations (cybernetics, complex systems theory) and real-world examples to explain why and how traditional IT organizations must transform to support the business in a digital world where customers expect new features to be delivered daily, but do not tolerate downtimes even though systems are becoming ever more complex. A book to hand to all IT managers.

DevOps: A Software Architect's

Building



Perspective, Bass, Weber, Zhu, Addison-Wesley 2015

This book that takes the (cloud) architecture viewpoint on DevOps, which is a great compliment e.g. to The Phoenix Project, which gives you the story behind DevOps and the motivation. The SEI titles can be a bit encyclopedic, but are thorough and this one is refreshingly close to real-world cloud solutions and tooling. A must read if you deal with development organizations that are moving (or should be moving) to the cloud.

Microservices, Sam Newman, O'Reilly, 2015

Sam cuts through the micro services hype with experience gathered from actual implementations. He takes us through the complete lifecycle of microservices from modeling via integration to deployment and testing. As microservices are not entirely new, the book covers both topics specific to micro services, e.g. breaking down monoliths into services, and some aspects not unique to just microservices, such as continuous delivery and virtualization. I feel that this rounds off the book quite nicely.