

ENTERPRISE SERVICE ORIENTED ARCHITECTURES

Enterprise Service Oriented Architectures

Concepts, Challenges, Recommendations

by

JAMES MCGOVERN

OLIVER SIMS

ASHISH JAIN

MARK LITTLE

 Springer

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN-10 1-4020-3704-X (HB)

ISBN-13 978-1-4020-3704-7 (HB)

ISBN-10 1-4020-3705-8 (e-book)

ISBN-13 978-1-4020-3705-4 (e-book)

Published by Springer,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

www.springer.com

Printed on acid-free paper

All Rights Reserved

© 2006 Springer

No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, microfilming, recording or otherwise, without written permission from the Publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Printed in the Netherlands.

The condition of an enlightened mind is a surrendered heart.
Alan Redpath

Author Team

To those who are savage in the pursuit of excellence ...

James

To my wife Sherry and sons James and Sylvester who provide and replenish the energy necessary for me to complete the exciting work as well as the mundane. To Mom and Dad, thanks for the encouragement and persistence.

Oliver

To my wife, Heather, for just about everything, and to my children Christopher, Richard, and David, of whom I am inordinately proud, and who have kept me firmly rooted in reality.

Ashish

To my wife Nishma and children Eshan and Ronit for their love, patience and support. To my parents, for their encouragement throughout the years.

Mark

I'd like to send my love to all my family, particularly my wife Paula and our children Daniel and Adam, who have acted as an anchor for me through the years, keeping me sane throughout the storms.

TABLE OF CONTENTS

ENDORSEMENTS	xi
ABOUT THE SERIES	xiii
FOREWORD	xvii
PREFACE	xxi
ABOUT THIS BOOK	xxv
ACKNOWLEDGEMENTS	xxxi
ABOUT THE REVIEWERS	xxxiii
 CHAPTER 1: UNDERSTANDING SERVICE-ORIENTED ARCHITECTURE	 1
Introduction	3
1. Introducing Service-Oriented Architectures	5
2. Service-Based Collaboration through Federation	19
3. The Platform	32
4. Transitioning to ESOA	45
5. Summary	48
 CHAPTER 2: COMPONENT-BASED SERVICES	 49
1. Component-Based Software Engineering (CBSE)	51
2. A Component Definition	55
3. Component Granularity	64

4. From Requirements to Design	81
5. Summary	94
CHAPTER 3: ORCHESTRATION	95
1. Workflow and Business Process Management	97
2. The Business Process Execution Language (BPEL)	101
3. A Worked Example of Web Services Orchestration	128
4. Design-Time Demonstration	129
5. Run-Time Demonstration	145
6. Summary	148
CHAPTER 4: WORKING WITH REGISTRY AND UDDI	151
1. Introducing the Registry	152
2. Universal Description, Discovery and Integration (UDDI)	154
3. Programming UDDI	169
4. Internationalization	179
5. Summary	187
CHAPTER 5: UNDERSTANDING ENTERPRISE SECURITY	189
1. Need for a Message Level Security Solution	191
2. Security Concepts	193
3. Security Technologies	196
4. Web Services Security (WSS)	225
5. WS-Policy	230
6. WS-Trust	231
7. WS-Privacy	232
8. WS-SecureConversation	232
9. WS-Federation	233

10. WS-Authorization	233
11. Summary	233

CHAPTER 6: SOA MANAGEMENT 235

1. Problem Space	236
2. Systems Management	244
3. Alerting	250
4. Provisioning	252
5. Leasing	253
6. Billing	254
7. Pricing/Chargeback Models	255
8. Lifecycle Management	257
9. Management Architecture	271
10. Policy Architecture	277
11. Framework Vendors	279
12. Summary	280

CHAPTER 7: TRANSACTIONS 281

1. What Are ACID Transactions?	281
2. Why ACID Is Too Strong for Web Services	288
3. A Brief History of Web Services Transactions	290
4. The Coordination Frameworks	291
5. Web Services Transactions	299
6. Security Implications	312
7. Interoperability Considerations	314
8. Summary	315

CHAPTER 8: EVENT-DRIVEN ARCHITECTURE	317
1. Overview	319
2. Events	320
3. Agents	323
4. Threads	329
5. Alternative Pattern-Based Approaches	332
6. Language Specific Constructs	338
7. Finite State Machines	344
8. Event Notification	347
9. Practical Considerations	352
10. Summary	355
OUTTRO	357
APPENDIX A: UNDERSTANDING DISTRIBUTED COMPUTING	359
1. Distributed Computing	360
2. Practical Considerations	385
3. Summary	385
APPENDIX B: QUALITY ATTRIBUTES	387
1. System Qualities	387
2. Design vs Run-Time	391
APPENDIX C: REFERENCES	395
APPENDIX D: ADDITIONAL READING	403
APPENDIX E: UPCOMING BOOKS	405

ENDORSEMENTS

*You can't live a perfect day without doing something
for someone who will never be able to repay you.*

John Wooden

"Enterprise SOA is well written and insightful. This book covers enormous ground and readers will find it the best, single source on SOA. Highly recommended!"

Ron Widitz
Enterprise Architect
Discover Financial

"This book was truly a guide for all levels of individuals developing business applications in today's global, open market. It clearly summarizes key concepts for executive management, provides the framework of architectural guidelines and standards, as well as provided detailed coding examples for entry level developers. This book should be a must read for all interested in leading their organization's business model into the future."

Damon Rothstein
Enterprise Network Architect
Piper Jaffray and Companies

"Concise, readable and useful as a tool in the running of a business. You truly pull concepts together with real world examples."

W.M. Douglas Crawford
VP NASDAQ Technology & Operations
Advest

"Enterprise SOA provides architects and developers with an excellent source of much needed information on understanding how to utilize enterprise technologies such as

SOA, orchestration, components, and registries to solve problems that currently face the enterprise. Understanding these technologies will help architects develop systems that can solve current problems as well as lay down an architecture that will adapt to on going changes in the business environment.”

Suneet Shah
CTO and Chief Architect
Diamelle

“Enterprise Service-Oriented Architectures provides a unique and worthwhile lifecycle-perspective to realizing a SOA. A number of concepts such as components, registries, web-service security, management, business processes, etc. are addressed in the context of different stages during the realization of a SOA, including : translating SOA requirements to design, design to implementation, and implementation to deployment.”

Sekhar Sarukkai
Technical Evangelist
Oblix

“This book is an outstanding and insightful work on the perspectives and potential of service-oriented architecture. A must read for every Enterprise Architect who needs to know how to succeed in the face of architectural challenges presented as part of his/her daily chores.”

Nitin Narayan
CEO
Mavenz, India

“This book is the product of some of the leading thinkers in Information Technology today. The concepts included in this book are being debated and analyzed by most of the Information Officers in the world right now. This book provides a history of how we got to SOAs, what they mean today, and where they will lead tomorrow. The implications of SOAs, Web Services, Federation, BPEL, and Grid computing, will revolutionize the IT industry. We are living in truly interesting times. Those of us in the IT community have our work cut out for us to lead our companies and customers into the next generation of computing. Thank you for this great book to help spearhead the charge!”

Joe Gibson
Senior Practice Director, Integration
East Area Technology Services
Oracle

ABOUT THE SERIES

Before you can inspire with emotion, you must be swamped with it yourself. Before you can move their tears, your own must flow. To convince them, you must yourself believe.

Winston Churchill

The new enterprise requires a new type of leadership based on the logical continuation of historical effort, while not doing what has been done just because it has been done that way in the past. Agility and leadership when combined is a cohesive presentation of common sense mined from the few truly successful projects as opposed to the aggregation of every (predominately failed) effort declared complete not only successful projects but projects that really add value to a business imperative. We are living in a new era where one seeks uncommon approaches while maintaining common virtues.

The art of leadership is about knowing and influencing people so that they can come to share common values resulting in more efficiency in achieving the strategic vision for the enterprise. Leadership must also embrace diversity which will lead to a much more lively dialectic.

The Enterprise Series has earned its place as a valuable resource to those who want to ramp up quickly and stay ahead of the curve. The authors of books within this series are not writers hired to cover the “hot” topic of the minute. Instead they are thought leaders and expert practitioners who bring insight to the community at large based on real-world experience. More importantly, they are not theorists but actually practice what they preach.

This series is founded on the conviction that enterprises should differentiate themselves, their architecture and their people based on the way they think as much as on the products or services they sell. Thought leadership is all about time-honored expertise and comprehensive capabilities. Its inflection point however is new thinking and original perspectives.

We hope you find this series and this book to be a practical guide and trusted advisor leading you successfully on your journey.

James McGovern
Rajanish Dass
Anthony Finkelstein
John Götze

Series Editors

James McGovern

James is an industry thought leader and the co-author of several recent books on service-oriented architectures, enterprise architectures and technology leadership. He is employed as an Enterprise Architect for The Hartford Financial Services Group, Inc. He holds industry certifications from Microsoft, Cisco and Sun. James is a popular speaker at elite technology conferences around the globe. He is member of the Java Community Process and the Worldwide Institute of Software Architects.

Rajanish Dass

Rajanish is a Fellow at IIM Calcutta and his primary research and teaching interests are in the area of Information Systems Management and its applications to various arenas of doing business. Particular interests lie in the area of real-time Data Mining and Business Intelligence techniques, developing algorithms and heuristics for performance enhancement, applications of Business Intelligence techniques for various areas of business like supply chain management, network intrusion detection, privacy and security enhancement, anti-spamming techniques, retailing, finance and business policy making, Competitive Intelligence, etc.

A rank holder throughout his career; he has experiences in working for high-ended projects at the IBM Research Laboratory India and at the Tata Consultancy Services. He has published a number of research papers in forums of international repute and of national significance. He is currently doing research in collaboration with world renowned research labs of MIT, Oracle Centre of Excellence, Messagelabs, etc.

Anthony Finkelstein

Anthony is Professor of Software Systems Engineering at University College London. He is a founder of London Software Systems a new joint research institute established jointly by UCL and Imperial College. He has contributed to

research on software specification methods, software development processes, tool and environment support for software development. Recent work has included significant contributions to work on tools for managing model integrity in software development, specification from multiple viewpoints and requirements engineering. His current interest's area is the managing of distributed information in software development. He is the Head of the Department of Computer Science and founder of Systemwire a technology spinout company.

John Gotze

John is chief consultant and enterprise architect in the Danish Ministry of Science, Technology and Innovation, where he since 2001 has been heavily involved in establishing and managing the Danish government's enterprise architecture program, as an important part of the national e-government policy. Recent work has included policy documents as well as academic publications on cross-governmental interoperability, architecture frameworks, service-oriented architecture and open standards. He is also a non-tenured associate professor at Copenhagen Business School and the IT University of Copenhagen, where he teaches masters classes and supervises theses. He holds a PhD and MSc from the Technical University of Denmark.

FOREWORD

We live in an interesting (in the sense of the “*May your life be interesting*” Chinese proverb) moment of the IT industry. Finally, many of the disciplines required to manage IT in a structured and agile way are converging together. We now have the theoretical concepts and practical experience to align IT to the business, manage IT as a business, define enterprise architectures, and align IT initiatives and individual projects (not only as far as development is concerned, but also with regards to outsourcing, deployment, integration, governance, and the other IT concerns). Nearly any imaginable technology IT challenges can currently be solved (or has been solved) by some company in the industry. Companies willing to invest the time and money and having the experience (or the luck ...) to do it can truly take advantage of IT to achieve their business objectives. And we have many failures and successes to learn from, and to build upon.

A key architectural and technology element of this convergence is an architectural style that became known in the year 2000 as “Service-Oriented Architecture”. Since 2003, this has started to morph toward an architectural style with the potential of impacting the vast majority of IT: “an Enterprise Service-Oriented Architecture” (ESOA). In its widest meaning, this term indicates the architectural style enabling an interoperability layer reducing the costs of integration, creating a technical and functional decoupling between applications, and supporting an enterprise-wide rationalization of IT applications. When ESOA is fully adopted, enterprises can align IT to their business processes and create a transitioning path toward a normalized IT portfolio. As such, an ESOA has the potential to profoundly impact both business and IT.

It is indeed the first time in our industry that the IT disciplines, technologies, approaches are coming together to this extent. And still, IT is today more complex than ever: addressing IT this way is still beyond the capability of most organizations, both due to the costs and to the complexity of the task. The industry still needs to mature to bring costs and complexity down. For this to happen, among other things

we need to see books addressing the required architectural elements. This is one of those books.

In 1996, Oliver Sims and I were working together on a large product development (over 500 hundred developers building an ERP product with thousands of tables). Oliver introduced a concept he called “semantic messages”. A semantic message was a message that contained not only the data that needed to be sent, but also tags describing the data being sent. For a couple of years, we explored together the many challenges of addressing interoperability through “semantic messages” to address development, interoperability, and deployment “in-the-large”. We built a simple language for describing these semantic messages. This was the first time I had met the concept of what I later called a “strongly-tagged language”. We also built the infrastructure, patterns, architectures, modeling tools and code-generation tools, repositories, and processes to define these semantic messages in an architecturally consistent way. As often happens in our industry, in time I discovered that many other teams had been working with similar approaches in their companies.

These approaches were not standards, but rather very proprietary approaches. So when XML came out, many people were happy to finally see a “tagged language” that was being standardized and adopted industry-wide: something for which we had unsuccessfully lobbied in various standard bodies for years. Now finally the industry had a standard as a basis for what we believed was the best way to address many interoperability issues.

Of course, XML is just a very basic language, alone it cannot do much: to achieve its potential, it needed a whole set of additional standards, technologies, infrastructures, frameworks and tools to cover the whole spectrum of what is needed for interoperability. Today, this is reflected in the many Web Services standards, and in the many products being sold to support Web Services. But, once these basic technical layers are addressed, any serious project needs to address, among other things, the architectural issues. The point is not (and has never been) the technology: the point is how to use these technologies in the various architectural choices we have to make to address the requirements.

The industry has come a long way. Many Web Services technology providers have come and gone. The industry is stabilizing and consolidating. The Open Source movement has brought costs down. Second-generation architectural approaches provide maturity models for component blueprints and models enabling faster and more reliable software manufacturing. Now we have much experience with Web Services infrastructures, and, not only in specific companies but industry-wide, we know what works and what doesn't. For the business, this promises the elimination of costly, redundant, and proprietary approaches, and the ability to integrate applications

quickly and easily. IT is rapidly commoditizing, and ESOA helps drive value higher and higher up the IT chain.

And finally, we start to see books not treating the problem as a simple “standard” or technology problem, not focusing only on the basic Web Services bricks, but positioning Web Services within the larger architectural perspective. The book you hold in your hands is a fine example of this.

This book has the advantage of having been written by industry practitioners covering many perspectives in IT: the authors together have the right mix of technology product perspective, consultant perspective, and large IT shop perspective. The book covers many of the most important topics: components, registries, security, management, transactions, and events. As such, it addresses an important need: bridging the gap between technology and architecture.

Enterprise Service-Oriented Architectures are one of the most significant evolutions in the IT industry in the past few years. They share the spotlight with other significant evolutions and trends. These include the wide-spread adoption of enterprise architectures, the creation of an enterprise architecture discipline that looks well beyond “software architecture” to address the many business, functional, structural and technical aspects of IT today, the maturity of governance and compliance process frameworks, and the application of agile concepts to all aspects of IT (including deployment of packaged software and outsourcing). But software architecture remains the critical prerequisite for success in IT and in ESOA in particular: this book addresses this prerequisite.

Thanks, guys, for putting together a fine and timely book.

Peter Herzum
President, Herzum Software

PREFACE

*Don't tell people how to do things, tell them what to do
and let them surprise you with their results ...*

George S. Patton

Service-oriented architectures (SOA) fundamentally changes the way enterprise software is developed and deployed. SOA enterprise applications evolve. The change will morph existing software, as we currently know it away from using monolithic approaches. Instead, SOA will enable virtualized on-demand execution models that break the current economic and technological bottleneck caused by traditional approaches.

Software as a service has become pervasive as a model for forward looking enterprises to streamline operations, lower cost of ownership and provides competitive differentiation in the marketplace. SOA is not a silver bullet that will address all of the deficiencies within an enterprise related to integration, reusability or the elimination of redundant systems. SOA can assist in these problem domains in an incremental manner while providing guidance on business architecture as well.

Service orientation has become a viable opportunity for enterprises to increase the internal rate of return on a project-by-project basis, react to rapidly changing market conditions and conduct transactions with business partners at will. Loosely coupled, standards-based architectures are one approach to distributed computing that will allow software resources available on the network to be leveraged. Applications that separate business processes, presentation rules, business rules and data access into separate loosely coupled layers will not only assist in the construction of better software but also make it more adaptable to future change.

Service-oriented architectures will allow for combining existing functions with new development efforts, allowing the creation of composite applications. Leveraging high-quality low-cost applications that can be procured from third party suppliers to

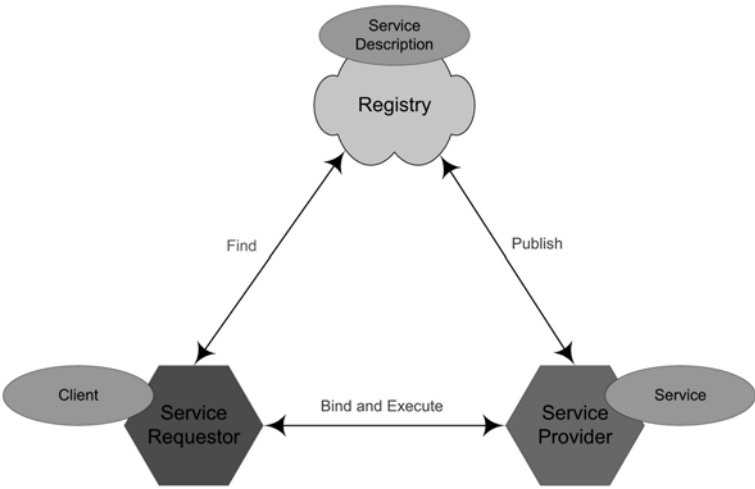


Figure 1. Service-Oriented Architecture

replace lower quality components lowers the risks in software development projects. By reusing existing functions that are of high quality, it leads to faster deliverables and increases the potential for overall higher quality.

Loose coupling helps preserve the future by allowing parts of the business and the technology that provides support to change at their own pace without the risks linked to costly migrations using monolithic approaches. SOA allows business users to focus on business problems at hand without worrying about the technical constraints within existing IT applications as they can choose to replace components incrementally. For the individuals who develop solutions, SOA helps in the following manner:

- Business analysts focus on higher order responsibilities in the development lifecycle while increasing their own knowledge of the business domain.
- Separating functionality into component-based services that can be tackled by multiple teams enables parallel development.
- Quality assurance and unit testing become more efficient; while errors using SOA have the potential to be discovered much later in the lifecycle, the overall quantity will be reduced to support better service level agreements (SLA) and development specifications.
- Component-based approaches when used within a service-oriented architecture can aid in becoming reusable assets for systems and software that are constructed using proper software development techniques.

- Functional decomposition of services and their underlying components with respect to the business process helps preserve the flexibility, future maintainability and eases integration efforts.
- Security rules defined for consumers and providers based on role, function, context and method of interface that helps solve many security considerations within the enterprise.

Implementing a service-oriented architecture will not happen overnight nor requires enterprises to throw existing technology investments. The challenge will be in deciding which services are core to your business and investing in them and leveraging services provided by others as appropriate. In all technology revolutions the prior generation usually remains a vital part of the infrastructure but not the only part. Software built to support services significantly reduces the need for enterprise software; not eliminate it. Service-oriented architecture is an economically responsive model to the needs of IT and enables the corporation to focus more on the business process and the applications rather than the deployment of more enterprise software.

Service-oriented architectures also provide an opportunity for an organization to mature their enterprise architecture perspective. Enterprise architecture provides the necessary framework for categorizing everything for the purpose of supporting future business planning. Both business and IT executives should be able to look at the enterprise architecture and understand the tradeoffs in using various systems, processes, technologies and so on. SOA can be the next level of specification and provide guidance on the best ways to realize many of the goals of enterprise architecture.

ABOUT THIS BOOK

The great revolution in our generation is that of human beings, who by changing the inner attitudes of their minds, can change the outer aspects of their lives.

Marilyn Ferguson

The goal of this book is to share insight gathered by industry thought leaders in a practical easy to read manner. This book contains many leading edge examples that illustrate how agile approaches to enterprise architecture can be applied to existing business and technology issues. It will help one focus on how to think concretely about enterprise architecture while providing solutions to today's problems.

Within the covers of this book, you will learn about the following topics:

- Fundamentals of a Service-Oriented Architecture,
- Component-Based Services,
- Orchestration,
- Registries,
- Management,
- Transactions,
- Event-Driven Architecture, and
- Understanding Distributed Computing.

Audience

This book is for every Java and .NET developer and architects of Fortune 1000 enterprises and the consultancies that service them who have the drive to spend extra hours feverishly seeking bits of wisdom on their craft and who want to gain the latest insights and strategies for leveraging emerging enterprise SOA disciplines for value creation, increased business agility and strategic competitive advantage.

This book does assume that one has significant IT experience under their belt and have worked on projects that were both large and small; on time as well as those which are over budget using different project management, software development and infrastructure paradigms. This book is not for those who desire all the information they require in a single book as this is an impossible goal to achieve; rather this is best suited for those who want to gain insight from thought leaders and are willing to be savage in the leap from good to great.

Finally, this book is aimed at the people who must create applications in the real world day-in, day-out. Many of the best practices books treat each tip as the sole focus of a chapter with no discussion of integrating it into a real application. Real applications are tenuous at best, requiring lots of moving parts in order to function together. Concepts and recommendations within this book are presented in context of a living enterprise with all the places that the real world intersects with the academia of the problem space at hand.

The hardest part of starting an enterprise service-oriented architecture initiative is knowing where to begin. We hope that our insights will be a useful starting point for a set of well-managed endeavors for many an architect.

What This Book Is Not!

First, this is not a book for beginners. If you are looking for introductory material on service-oriented architectures, we recommend searching for articles on the Internet using search engines such as Google. Second, while this book may sporadically use SOAP, WSDL and UDDI snippets to describe SOA concepts, this book is not about web services. If you require a book on web services, we recommend *Java Web Services Architecture* published in 2003 by Morgan Kaufmann.

Many of the examples contained within this book use XML. If you do not have a thorough understanding, we recommend the following books:

- Learning XML, Second Edition – O'Reilly.

- *Effective XML: 50 Specific Ways to Improve your XML* – O'Reilly.

The authors recommend that the readers have a thorough understanding of general software development, design patterns, enterprise integration and network infrastructure as a foundation to truly understanding the material contained within. Some good books are:

- *Enterprise Integration Patterns* – Addison Wesley.
- *Pattern-Oriented Software Architecture*, Volume 1 – Wiley.
- *Network Security Architectures* – Cisco Press.
- *Introduction to Networking* – McGraw-Hill.
- *Software Architecture in Practice* – Addison Wesley.

Finally, if you have not programmed using a modern language such as Java and/or .NET, we recommend the following books as a good starting point:

- *Java 2 Enterprise Edition Bible* – John Wiley & Sons.
- *Bitter Java* – Manning Publications.
- *Java 2: A Beginner's Guide* – McGraw Hill.
- *Beginning VB.NET (Programmer to Programmer)* – Wrox.

How to Use This Book

The authors have taken deliberate action to avoid filling this book with lots of code and specification information. This book strives to provide vivid insight into the dynamics of SOA and a strategic approach to successfully integrate technology into business decisions and the day-to-day actions of the workforce, business partners and the consumer to reduce waste of resource and enhance the deliverables of the enterprise.

The goal of this book is to strive to be both practical and philosophical. Sections of the book may discuss general principles with examples that illustrate them: examples drawn not from trivial demonstration programs (i.e. pet stores, shopping carts, etc.) but rather actual architectures used in production systems in industry applications such as financial services, supply chains, manufacturing, defense and telecommunications.

Motivation for Writing This Book

There is an abundance of books on Web Services on the market many of which are bestsellers. The vast majority of texts simply explain vendor product offerings, APIs and very brief tutorials without providing the reader with sufficient understanding of the problems that exist within enterprises. The authors felt it is not sufficient to merely learn the technology and APIs but to understand the principles behind them so that they can be appropriately incorporated into existing systems and applications.

Many books also explain the happy path to software development. Knowing what works is good (design pattern), but knowing what doesn't work is better (anti-pattern). Where appropriate, this book will outline potential pitfalls. The authors of this book have been doing service-oriented software development before it became a hot topic in industry magazines and journals. The authors themselves are relentless in the pursuit of finding better ways to develop software, which relies on thinking about problems of the past differently. Success is bred by not simply understanding technology but also by understanding agile methods for realizing the business goal.

The author team's primary motivation in writing this book is to share our working experience to bridge the gap between the knowledge of industry gurus and newcomers to service-oriented architectures. Many powerful techniques lay hidden in magazine articles, conference proceedings, vendor marketing materials and academic papers for years before becoming recognized by the masses.

NOTE: For those who are not familiar with Agile Methods, we recommend you read the Agile Manifesto at <http://www.agilemanifesto.org>.

Disclaimer

The advice, diagrams and recommendations contained within this book may be used as your heart desires, with the sole discretion that you may not claim that you were the author. The publisher, authors or their respective employers do not provide any form of warranty or guarantee its usefulness for any particular purpose.

This book is 100% error free! Not! If you believed us even for a second, I have a suspension bridge in my backyard that I would like to sell you. The author team and editors have worked hard to bring you an easy to understand, accurate guide on Enterprise Service-Oriented Architectures. If you find any mistakes in this book, we would appreciate your contacting us via email at serviceorientedbook@yahoogroups.com.

This book may use from time to time examples of a fictitious organization. Any example companies, organizations, products, domain names, email addresses, people, places, events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, person, places or events is intended or should be inferred.

About the Authors

James McGovern

James is the co-author of several bestselling books, including *A Practical Guide to Enterprise Architecture*. He is employed as an Enterprise Architect for The Hartford, a Fortune 100 financial services firm. James is a popular speaker at elite technology conferences around the globe. He is a member of the Java Community Process and the Worldwide Institute of Software Architects.

Oliver Sims

Oliver is a recognized leader in the architecture, design, and implementation of service-oriented and component-based enterprise systems. Currently an independent consultant, Oliver has held a number of senior technical positions in software product and service companies. He was a founding member of the OMG Architecture Board, has been active in several OMG Task Forces, and has contributed to the development of MDA. Co-author of *Business Component Factory* and *Building Business Objects*, and author of *Business Objects*.

Oliver has many published articles to his credit. Oliver's experience in IT and software spans three decades, and he has accumulated wide practical experience in a number of roles and with many kinds of system. He was one of the first to prove the synergy between components and distributed objects, and was chief architect for a groundbreaking component container middleware product in the mid 1990s. Most recently Oliver has helped IT organizations in their transition to effective component-based high-productivity development of service-oriented systems.

Ashish Jain

Ashish is an enterprise architect with over 11 years of industry experience with expertise in the architecture, development and mentoring of the mid-tier and server-side components. He currently works as a Principal Architect with Ping Identity Corporation, a leading provider of solutions for identity federation. Prior to joining Ping Identity, he worked with BEA Systems where his role was to assist BEA customers in designing and implementing their e-business

strategies using solutions based on J2EE. He holds several industry certifications from SUN and BEA and is also a board member for the Denver BEA User Group. He is also the co-author of *The J2EE 1.4 Bible* (ISBN: 0-7645-3966-3, <http://www.amazon.com/exec/obidos/tg/detail/-/0764539663>).

Mark Little

Mark is Chief Architect for Arjuna Technologies Ltd., a spin-off company from Hewlett-Packard, specializing in the development of reliable middleware. At HP Mark was a Distinguished Engineer and architect working on their transaction and Web Service products. He is one of the primary authors of the OMG Activity Service specification and is on the expert group for the same work in J2EE (JSR 95). He is also the specification lead for JSR 156: Java API for XML Transactions. He is on the OMG's OTS Revision Task Force and the OASIS Business Transactions Protocol, OASIS WS-CAF and OASIS WS-TX Technical Committees, to name a few.

Before joining HP he was for over 10 years a member of the Arjuna team within the University of Newcastle upon Tyne (where he continues to have a Visiting Fellowship). His research within the Arjuna team included replication and transactions support, which include the construction of an OTS/JTS compliant transaction processing system. Mark has published extensively in the *Web Services Journal*, *Java Developers Journal* and other journals and magazines. He is also the co-author of several books including *Java Transactions Processing: Design and Implementation* and *The J2EE 1.4 Bible*.

ACKNOWLEDGEMENTS

When the character of a man is not clear to you, look at his friends.

Japanese Proverb

A book like this is never just the work of those whose names appear on the cover. Like the academy awards, “there are so many people we would like to thank ...” We are immensely grateful for all those who have encouraged us, provided practical advice, debated the finer points on controversial topics and whose insights have honed our own.

The authors have benefited immensely in their own professions by reading papers from other industry thought leaders including but not limited to (in no particular order):

Jeff Schneider
Doug Barry

Martin Fowler
Doug Kaye

Anne Thomas Manes
Peter Herzum

The author team would also like to thank other writers we have worked with in the past and desire to work with in the future (in no particular order):

Per Bothner
Jason Gordon
Leeanne Phillips
Sameer Tyagi
Elias Jo
Nitin Narayan

Kurt Cagle
James Linn
Sunil Mathew
Alan Williamson
Scott W. Ambler
Rahul Tyagi

Yakov Fain
Lynn Denoia
Vaidyanathan Nagarajan
Vikas Sharan
Dave Hollander
Kito Mann

Finally, the author team would like to thank our editor, Robbert van Berckelaer, for allowing our ideas to be published in a manner the community sorely needs and most importantly our copy editor, Jolanda Karada, for painstakingly copyediting and ensuring this book is error-free.

James McGovern

Best wishes to the reluctant warriors within the Armed Forces who spread freedom throughout the planet. Prayers to the families in Palestine, Israel and other parts of the Middle East who seek peace and those who have lost their lives in pursuit of it. To all of my coworkers at The Hartford who have been merciless in the support of speed, agility and balance in our daily lives. To Democrats, who make thievery and cowardice sound so romantic. To Republicans, who make Democrats look principled.

Regardless of land, religion or language, there is just but one God. I must thank our creator whom has bestowed upon me many favors. I am grateful for being blessed with a great family, doing the things I love and working for such a great employer. To say that I am not worthy of such blessings is obvious to anyone who knows me, which makes me all the more grateful.

Oliver Sims

Such insights as I may have developed over the years have been mainly due to the many valued colleagues with whom I have had the honor of working. In particular, I would like to thank Martin Anderson, Alan Boother, Roger Brown, Peter Eeles, David Frankel, Mike Guttman, Peter Herzum, Haim Kilov, Wojtek Kozaczynski, Maurice Perks, Dave Roberts, Mike Rosen, Trevor Sharpe, Sandy Tyndale-Biscoe, Rick Williams, and Bryan Wood.

Ashish Jain

I would like to thank all my colleagues at Ping Identity for helping me learn everyday. In particular, I would like to thank Darren Platt and Brian Whitney for taking the time to share their real world experiences with me. I would also like to thank my ex-colleagues at BEA, John Funk and Bob Webster, for reviewing the content and their invaluable comments.

Mark Little

I would like to thank all of my colleagues at Arjuna Technologies, who have helped to make it a great working environment over the years. In particular Stuart Wheeler, Barry Hodgson, Dave Ingham, Steve Caughey and the members of the transactions team, past and present. Many thanks go to Professor Santosh Shrivastava of the University of Newcastle upon Tyne, who started my career off in this direction and has been a constant friend over the many years. Thanks to my ex-Bluestone and Hewlett-Packard friends and colleagues, including Bob Bickel, ex-general manager of Hewlett-Packard Middleware, Al Smith, Greg Pavlik and Jon Maron, who showed me that the best things in life are free.

ABOUT THE REVIEWERS

Synergism is the simultaneous actions of separate entities which together have greater total effect than the sum of their individual effects.

Buchholz and Roth

The author team owes a debt of gratitude to all of the reviewers who provided guidance, feedback, constructive criticism, praise and encouragement throughout the manuscript writing process. Our reviewers came with diverse backgrounds: from people who believe in traditional processes to those who have embraced agile methods; from those whose native tongue is English, to those who speak joyous languages such as Arabic, Chinese, French, Hindi, Spanish, Urdu and others; from those who are lowly developers and project managers to those who are senior executives, this book would not be what it is without you.

The author team deeply appreciates all of the feedback received during the manuscript writing process from the following individuals (company affiliations are included where permitted):

Argentina

Daniel J. Biondi, Regional CT, EDS Latin America Solution Centres, Buenos Aires

Australia

Shaji Sethu, Solutions Architect, Volante, North Ryde, NSW

Belgium

Fanuel Dewever, Consultant, Human Capital Management, IBM Business Consulting Services

Robin Mulkers, Head Application Architect, Christelijke Mutualiteit

Canada

Henry Chiu, Architect/Trainer/Mentor, ObjectSoft Solutions, Ontario

Finland

Anders Aspнас, Solution Architect, Fujitsu Services OY

Germany

Stefan Tilkov, Managing Director, innoQ Deutschland GmbH, Ratingen

India

Naveen Gabrani, Software Project Manager, Computer Sciences Corporation

Rohit, S. Gulati, CISA, CSQA, Iflex Solutions, Bangalore

Shivprasad Koirala, AG Technologies, Nepal

Israel & Palestine

Issac Goldstand, Owner, Mirimar Networks, Jerusalem

Pakistan

Ammad Amjad, Systems Architect, Lahore Stock Exchange

Scotland

Jon C. Ferguson, PhD, Director and Architect, Omega Software, Ltd., Aberdeenshire

Singapore

Victor Putra Lesmana, IBM

Ukraine

Ruslan Shevchenko, Director, Grad-Soft Ltd., Kiev

United Kingdom

Max Kington, Technical Architect, Cantor Fitzgerald, London

United States

Wayne Allen, Product Development Manager, Corillian, Portland, Oregon

Matt Anderson, Senior Enterprise Architect, Great American Insurance Company, Ohio

Adolfo Lagomasino, Architect, Lucent Technologies, Landover, Maryland

Barbara McGowin, Principal, McGowin Enterprises, Goose Creek, South Carolina

Robert W. Meriam, Chief Infosec Officer, Catholic Health Services of Long Island, Melville, New York

David L. Nicol, Founder and CTO, TipJar LLC, Kansas City, Missouri

Donna M. Schaeffer, Associate Professor, University of San Francisco, California

Ravi Sharma, Senior Enterprise Architect, Systems Development Factory, General Motors, Detroit, Michigan