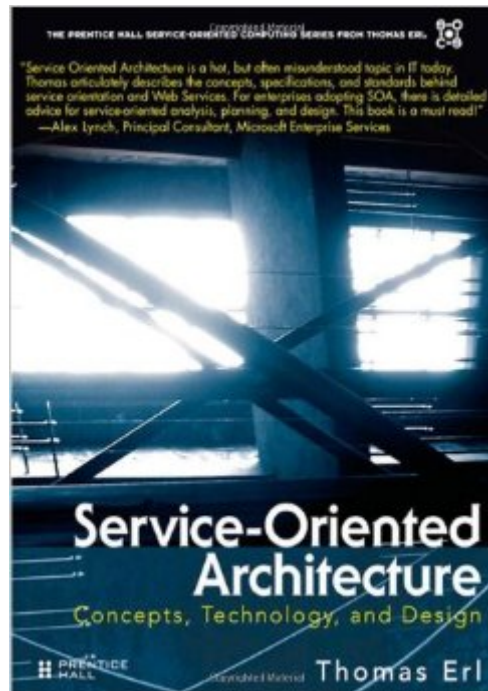


The book was found

Service-Oriented Architecture (SOA): Concepts, Technology, And Design



Synopsis

The first start-to-finish, step-by-step guide to modeling and designing SOA. Using Service-Oriented Architecture (SOA), enterprises can deliver high-value business services more rapidly and effectively, and gain unprecedented flexibility and value from existing IT infrastructure. SOA has earned the support of virtually every major software provider, and some 75% of enterprises surveyed are now investing in SOA technology and expertise. In *Service-Oriented Architecture: Concepts, Technology, and Design*, the author of 2004's best-selling SOA book presents the first end-to-end-tutorial for modeling and designing successful service-oriented architectures from the ground up. Writing in plain English, Thomas Erl provides step-by-step process descriptions for analyzing and designing any service or service-oriented business process definition.

Book Information

Hardcover: 792 pages

Publisher: Prentice Hall (August 12, 2005)

Language: English

ISBN-10: 0131858580

ISBN-13: 978-0131858589

Product Dimensions: 7.4 x 1.8 x 9.5 inches

Shipping Weight: 3.7 pounds (View shipping rates and policies)

Average Customer Review: 4.0 out of 5 stars [See all reviews](#) (69 customer reviews)

Best Sellers Rank: #253,797 in Books (See Top 100 in Books) #37 in [Books > Computers & Technology > Programming > Languages & Tools > XML](#) #72 in [Books > Computers & Technology > Web Development & Design > Web Services](#) #366 in [Books > Textbooks > Computer Science > Software Design & Engineering](#)

Customer Reviews

This book is superb. I have read every SOA book available (up until Apr/06) because it's part of my job as a technology research analyst and all-around techno-geek. From those that I have read and studied, this is the only one I feel compelled to write a review about. AND - because I did have to go through it in such detail I'm going to raid my research notes and share with you a detailed review of not just the book, but each of its chapters.

Chapter 1 - Introduction

Nothing special here, this is just a chapter that introduces the rest of the book. Call it a glorified table of contents if you will. At first I felt like skipping it altogether, but then I did what I'm supposed to do for my job and that is read each and every part. In the end, I'm glad I took the time for two reasons: By reading a summary of each

of the chapters I got a good feel for what this book was going to cover and what it wasn't going to cover. Secondly, I liked the author's intro stuff about ideal and not so ideal (real) SOA. Kind of insightful and stinging at the same time. Still, though, this is still just a description of other chapters. It's also a chapter you can get for free at the book's web site.

Chapter 2 - Case Studies Here the author provides background information for the two companies he uses as case studies. If you're into case studies, then you'll definitely want to read through this. But - I found the subsequent samples pretty easy to follow and I think you could get away with skipping this chapter if you really wanted to.

Chapter 3 - Introducing SOA Here's where I started getting into the meat of the book. If you think you don't understand what soa is or what the industry's made of it or turned it into then you need to read this chapter. It breaks it all down and builds it all up again in a very systematic manner. Make sure you leave this chapter with an understanding of how primitive and contemporary variations of soa are different because the author uses these terms later.

Chapter 4 - The Evolution of SOA Finally someone who makes a distinction between specification and standard and gets it right. This chapter talks about the soa industry and how vendors are responsible for soa but are also causing problems at the same time. How standards organizations are working for soa but also competing at the same time. Pretty interesting stuff and even though this was the least technical chapter, not once was I bored. It ends by comparing Ssoa with older architectures. I especially like how the author differentiates between soa and "traditional" distributed architecture that uses web services. (hint: rpc has a lot to do with it)

Chapter 5 - Web services and primitive soa I read the author's first soa book last year and this chapter seemed to repeat a few sections from that. But if I remember correctly it goes into more detail and provides case study examples that the first book didn't have. If you're a web services veteran you can probably skip this one.

Chapter 6 - Web Services and Contemporary SOA (Part I: Activity Management and Composition) Here he goes up a gear and dives right into that scary thing we've been calling ws-* Everything from transactions to context mgmt to orchestration and so on is covered. I really felt the author did a brilliant job building this chapter up by starting with simple meps and building up to activity management and bpm and so on. He really showed how each adds a layer over the other and how all add layers to soa.

Chapter 7 - Web Services and Contemporary SOA (Part II: Advanced Messaging, Metadata, and Security) Yup, the rollercoast ride continues here as he gets into addressing, reliable messaging, security and other ws-* specs. All of these are specs I had already heard about and I think this type of coverage is appropriate for where soa is going. I forgot to mention that in this chapter and 6 he introduces 'in plain english' sections that are hilarious. They are humorous analogies that compare these complex technologies to analogies he writes about a car wash. Good,

fresh writing in the usual dull and dry techno world.

Chapter 8 - Principles of Service-Oriented Architecture

Essentially a whole bunch of theory about designing services and then eight specific 'principles' (dos and don'ts) about how to design services the right way for SOA. I had to go back and reread this chapter after I finished the book. I sort of glanced thru it at first but then found out that later chapters really use these principles. When I went through it again I actually thought this was pretty important stuff. This really is the next oo. You can get this chapter for free at the book web site too.

Chapter 9 - Service Layers

Study this if you're an application architect or enterprise architect. It shows what you can do with services built with service-orientation. The diagrams showing different types of layers combined together are pretty cool.

Chapter 10 - SOA Delivery Strategies

If you're a PM you'll love this chapter. It gets into the different phases in a SOA project and how you can reorganize them using 'delivery strategies' depending on your budgets and priorities. I'd pay extra close attention to the pros and cons parts where, after documenting these strategies in abstract, the author points out their true colors.

Chapter 11 + 12 - Service-Oriented Analysis I + II

Don't know where to start when it comes to figuring out your services? Well, the author lays it all out here. He provides a process for systematically breaking down your business logic and dividing it up into services. Chapter 12 is like an instruction manual about service model. Being from a web services background this was all new to me.

Chapter 13 - 16 - Service-Oriented Design I, II, III, IV

Roll up your sleeves man, because here is where you get into the real muck of building web services for an SOA. There are a bunch of processes that hash out the nitty gritty of WSDL, XSD, and BPEL and show you how to build services for the types of layers set up in ch.9. Tons of code and case study samples and tips for design. This is probably the most valuable part of the book for developers and architects.

Chapter 17 - Fundamental WS-* Extensions

I forgot to mention that in chapters 6 and 7 no code samples are given. He only covered WS-* specs conceptually. All of the corresponding code is placed in this chapter. A bit inconvenient if you're a developer who wants to see the code while learning about the spec, but not tragic. The neat thing is he ties the code samples into the case studies. This was my first experience with WS-* in real world type scenarios.

Chapter 18 - SOA Platforms

The author documents J2EE and .NET frameworks here first in total abstract and then about how they support the different parts of SOA. This was very interesting because it related a lot of the concepts stuff to actual technology and let you compare different technologies in how they support SOA. I recommend this book to colleagues and clients and I'm recommending it here. If you have questions about SOA then this book probably has the answers you're looking for. I say that because by the time I finished reading it I ran out of questions myself.

It's hard to understand how the same author wrote this and "SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)" and "Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)", both of which had more useful information in a much more compact package. The only real use I can think of for this book is perhaps to quote in a sales context regarding the benefits of SOA to someone who hasn't heard of it. That said, although I believe in SOA as a powerful mechanism, I believe the claims in the book are less well supported than the heft of the book might imply. Other technical details like the importance of UDDI are largely out of date. I disagree with some of the other reviewers who call the book overly theoretical: I would not give it that much credit. Theory would call on or reference solid research; this book provides anecdotal evidence at best. Aside from some potential use to sales folks (perhaps why Sun, IBM and MS endorse the book), I think most will want to pass on this one.

This book might be best described as SOA for managers. Most of the book covers high level concepts. In some parts everything is presented as an abstraction, leaving the reader to wonder what the connection with the real world is. Even as a book focused on a high level overview this book doesn't work. This should have been a 300 page book. Who has time to put up with an extra 400 pages? If your interest is in actually implementing something, you'll need to go far beyond this book. I've given it 2 stars instead of 1 because I did learn a few things from it.

The author should be a politician not a writer. This book is painful to read because it goes on-and-on but doesn't actually say that much. The same material could have easily been covered in 300 pages.

If you want a book that covers most of the SOA standards in one place, this might be helpful. I think you could get that from Wikipedia. Lots regurgitation of SOA platitudes, not much value add. If you're looking to make the light go on about key SOA concepts, this isn't the book. It would make a good management summary of the technology, if it was about 1/3 as long.

I have not in my long career in IT encountered anything quite as complex and confusing as what is being called service-oriented architecture or service-oriented computing. It's still distributed the way technology architectures have been for several years now. It still seems to use the same kind of technologies that internet apps have been using for the past half decade. So what exactly is a

service-oriented system? And, more importantly, what makes it better? I looked to this book to answer these and many more questions and to address a great deal of skepticism I had regarding this whole SOA trend. Erl's book provided me with an education in SOA and everything that surrounds it. The book systematically breaks down content into an intelligent sequence of sections and chapters that gradually and smoothly transition through basic to intermediate to advanced topics. It's a superbly written tutorial that blends coverage of technologies and theory with case studies and code examples. I honestly couldn't remember the last time I read such an organized and informative book. I think this book will be important because SOA is becoming the next big thing (yes, I am a believer now) and because there is nothing out there even close to providing this level of insight and guidance. A classic in the making and highly recommended.

[Download to continue reading...](#)

Service-Oriented Architecture (SOA): Concepts, Technology, and Design Service-Oriented Architecture: Concepts, Technology, and Design The Food Service Professional Guide to Controlling Restaurant & Food Service Operating Costs (The Food Service Professional Guide to, 5) (The Food Service Professionals Guide To) The Food Service Professional Guide to Controlling Restaurant & Food Service Food Costs (The Food Service Professional Guide to, 6) (The Food Service Professionals Guide To) Object Success : A Manager's Guide to Object-Oriented Technology And Its Impact On the Corporation (Object-Oriented Series) El Soñador: (Spanish language edition of The Dreamer) (Spanish Edition) Food Service Menus: Pricing and Managing the Food Service Menu for Maximum Profit (The Food Service Professional Guide to Series 13) Civil Service Exam Secrets Study Guide: Civil Service Test Review for the Civil Service Examination (Mometrix Secrets Study Guides) Reusable Software : The Base Object-Oriented Component Libraries (Prentice Hall Object-Oriented Series) Blockchain: The Comprehensive Guide to Mastering the Hidden Economy: (Blockchain Technology, Fintech, Financial Technology, Smart Contracts, Internet Technology) The Object-Oriented Approach: Concepts, Systems Development, and Modeling with UML, Second Edition Object Oriented Perl: A Comprehensive Guide to Concepts and Programming Techniques Good Deeds, Good Design: Community Service Through Architecture Object-Oriented Technology and Computing Systems Re-Engineering Visual Object-Oriented Programming Using Delphi With CD-ROM (SIGS: Advances in Object Technology) Object-Oriented Technology: From Diagram to Code with Visual Paradigm for UML N-Layered Domain-Oriented Architecture Guide with .NET 4.0 Drills: Science and Technology of Advanced Operations (Manufacturing Design and Technology) Solid Wood: Case Studies in Mass Timber Architecture, Technology and Design The Laws of Simplicity: Design, Technology, Business, LifeDesign,

