

Executive's Guide to Service-Oriented Architecture

ERIC A. MARKS

MICHAEL BELL



John Wiley & Sons, Inc.

Executive's Guide to Service-Oriented Architecture

ERIC A. MARKS

MICHAEL BELL



John Wiley & Sons, Inc.

This book is printed on acid-free paper. ©

Copyright © 2006 by Eric A. Marks and Michael Bell. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the publisher or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400, fax 978-646-8600, or on the web at www.copyright.com. Requests to the publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, 201-748-6011, fax 201-748-6008, or online at <http://www.wiley.com/go/permissions>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services, or technical support, please contact our Customer Care Department within the United States at 800-762-2974, outside the United States at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

For more information about Wiley products, visit our web site at <http://www.wiley.com>.

Library of Congress Cataloging-in-Publication Data:

Marks, Eric A.

Executive's guide to service-oriented architecture / Eric A. Marks, Michael Bell.

p. cm.

Includes index.

ISBN-13: 978-0-471-76894-4 (cloth)

ISBN-10: 0-471-76894-4 (cloth)

1. Business enterprises--Computer networks--Management. 2. Information technology--Management. 3. Computer network architectures. I. Bell, Michael, 1951- II. Title.

HD30.37.M3642 2006

004.068--dc22

2005034273

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Preface

In 2002, Web services were a hot topic and the concept of *service-oriented architecture* (SOA), while not a new idea, was beginning to pick up steam. It did not take long for organizations to realize that Web services mandated the concept and organizational model of SOA to guide their selection, design, implementation, and management. SOA, we know, is a critical discipline to make Web services, or services in the general sense, work together to help organizations achieve the business goals they are seeking. SOA is an important influence on information technology (IT) strategy and enterprise architecture.

This book is unlike any other SOA book on the market today. There are no XML snippets. There are no blocks of code. We seldom mention specific technology platforms or vendors. In this book we generalize SOA. We express business and technology issues of SOA so that they will apply to all industries, technology platforms, and operating environments, and cover all use cases. This book combines two critical SOA perspectives in one volume: the business perspective and the technical perspective. We examined SOA and services from two very different and complementary perspectives, yet we feel as if we have conquered many of those very barriers that create friction for business and IT organizations.

When we began this book, we established this mission statement for our efforts:

This book will represent the state of the art for SOA planning, business, organization, and services modeling, architecture design and implementation. This book will present a business and technology modeling approach that answers most of the critical questions asked by IT and business leaders in today's organizations: How do we get started with SOA? Where do we begin? Where should we focus our SOA efforts? What "services" should we begin with? How do we identify and expose them in our SOA? How do we measure results of our SOA efforts? This book will be a reference work for IT executives, architects, team leaders, and developers seeking to understand how to make SOA real for their organizations to enable desired business results.

As we discussed the outline, chapter ownership, and eventual integration editing, it became clear that what we were doing with this book reflected many of the organizational challenges that are

spurring business interest in SOA. We became a metaphor for an organization pursuing SOA—I was the business and Michael was the IT organization. As coauthors, we were merging two distinct approaches to SOA: one from a business and strategic perspective, more top-down in nature, and the other from a technical and architectural perspective. At first our language, perspectives, goals, and approaches were difficult to reconcile, but because we were committed to the project within a tight deadline, we had to work it out. And we did.

Many organizations wrestle with the semantic and linguistic barriers between the business community and the IT community, as well as between specific disciplines within the IT community. Often the overarching goals and objectives are shared, but the approaches to meeting those goals are quite different. After all, even different sides of the same coin are distinct and unique yet inseparable. And so it is with the concept of SOA.

SOA offers the potential to create a unified language of business based on a unit of analysis known as a service. In fact, we dedicate a chapter to the concept of services because they are indeed the fundamental unit of analysis for an SOA. The first SOA challenge is to establish shared meaning for services in a given organization. Our book is about *all* possible services in an organization, a subset of which will most likely be Web services. We are building a generalized model for services, and therefore in the remainder of this book we use the term “services” to also mean Web services.

Our goals for the book are lofty. We want to:

- Represent the state-of-the-art SOA planning and modeling book in the industry.
- Address “services” in the general case as opposed to targeting only Web services.
- Create an SOA business book with the technical aspects of SOA framed with “big picture” thinking and business results in mind.
- Be relevant beyond the expected 10-year SOA time horizon. In other words, we want to be a “lasting reference guide” for SOA.

Ultimately you, the reader, will tell us how we did. To answer the questions we hear all the time, we had to attack the SOA domain in three sections.

Part 1 of the book is all about SOA business concepts: the motivating forces for SOA, the importance of a general model of services, and SOA business modeling.

Chapter 1: Introduction to the SOA Business Model

Chapter 2: General Model for Services

Chapter 3: SOA Business Modeling

These three chapters set the stage for Part 2. Part 1 is focused on business concepts and will be very useful for business and IT executives, architects, and others seeking the business context and perspectives for SOA.

Part 2 delves into the topics of services identification, analysis and design, services integration using various classes of enabling technology, and achieving services reuse.

Chapter 4: Service Identification, Analysis, and Design

Chapter 5: SOA Technology and Services Integration Model

Chapter 6: Fundamentals of SOA Asset Reuse: Service Reusability Model

Part 2 is really targeted for chief technology officers, architects, and developers. These chapters contain some great business concepts and some innovative ideas to help with identification, modeling, and implementation of services for an organization.

Part 3 of the book focuses on the concepts of SOA governance, organizational models, and enterprise architecture, and also offers a new approach for harvesting return on investment (ROI) using SOA. These chapters address the aspects of SOA that require all of the constituents of an SOA—business management, process owners, IT leaders, architects, developers, business analysts—to work together.

Chapter 7: SOA Governance, Organization, and Behavior

Chapter 8: Architecture Organization Model

Chapter 9: SOA Business Case and Return on Investment Model

Chapter 7 is very thorough; we believe it is among the most comprehensive to date on the crucial topic of SOA governance. Chapter 8 addresses architecture, a critical capability for SOA success. And Chapter 9 creates a model for the realization of business results from SOA.

We hope you enjoy the book and engage in the SOA dialog with us. We believe we have filled a need in the industry with this work. SOA is a complex topic and a complex organizational goal. Our goal is to clarify, simplify, and enable your organization to realize its business objectives and goals through judicious implementation of SOA and services. Please let us know how we did.

Contents

PREFACE	vi
CHAPTER 1	
Introduction to the SOA Business Model	1
Elements of an SOA	1
Conceptual SOA Vision	2
Services	2
Enabling Technology	3
SOA Governance and Policies	3
Metrics	4
Organizational and Behavioral Change	4
SOA: Behavior and Culture	4
New SOA Conceptual, Architectural, and Organizational Models	6
SOA: Its Time Has Come	6
SOA: Evolved Enterprise Integration	11
SOA Means “Service-Oriented Agility”	20
Service-Oriented Business: SOA = Business-Oriented Architecture	25
Characteristics of a Zero-Integration Enterprise	27
What Are the Challenges of SOA?	29
CHAPTER 2	
General Model for Services	33
SOA Is All About Services	33
Services Address Persistent Challenges and Present New Opportunities	44
General Model of Services	47
Services in an SOA Context	51
Service Lifecycle	53
Summary	62
CHAPTER 3	
SOA Business Modeling	64
SOA Business Iteration Model	65
SOA Business Modeling	71
Identify Major Business Challenges: Imperative Analysis	78
Focusing Your SOA Efforts	91
Summary	98

CHAPTER 4**Services Identification, Analysis, and Design 99**

Begin with Candidate Business Services	103
Business Process Analysis	104
Core Entity Analysis	104
Opportunistic Approach via Already-Budgeted Projects	105
Business and Domain Expertise	106
Preexisting Services	106
Existing Business Applications	107
Top-Down or Bottom-Up? Both, Then Iterate	107
Focus on Candidate Business Services	107
Service Analysis and Design	108
Candidate Service Analysis	109
Service Design	123
Bottom-Up Design and Analysis Approach	147
Summary	149

CHAPTER 5**SOA Technology and Services Integration Model 150**

Service Integration Model Framework	152
Service Integration Generic Notation	153
Service Integration Principles	153
Service Integration Design	169
Service Integration Modeling	177
Service Product Mapping Component: Integration Product	
Mapping Strategy	202
Summary	206

CHAPTER 6**Fundamentals of SOA Asset Reuse: Service Reusability Model 207**

Major Influences on Service Reusability	209
Service Reuse Model	210
Service Reusability Strategy	211
Service Construction	217
Service Integration	226
Service Management	234
Summary	242

CHAPTER 7**SOA Governance, Organization, and Behavior 243**

Architecture's Role in an SOA	245
Dynamic Architecture versus Static Architecture	246
SOA: Spatial and Temporal Challenges	247

SOA Governance Overview	248
Organization of Governance	250
What Does SOA Governance Do?	252
SOA Governance Organizational Model	253
SOA Governance Processes	256
Closed Loop SOA Governance	256
What Is the SOA Governance Process?	260
SOA Policies: Where SOA Governance Gets Real	262
SOA Governance Implementation and Integration	266
Enabling Technology of SOA Governance	268
SOA Behavioral Model: Beyond SOA Governance	276
Summary	283

CHAPTER 8

Architecture Organization Model	284
When Should Enterprise Architecture Be Reviewed?	285
Key Questions	285
SOA-Driven Enterprise Architecture Model	287
Business Requirements	289
Technological Requirements	291
Architecture Disciplines	298
Architecture Practices	302
Architecture Roles	305
Architecture Role Models	308
Positioning of Architecture Organizations: An SOA Perspective	310
Architecture Organizational Management Structure	315
Summary	320

CHAPTER 9

SOA Business Case and Return on Investment Model	322
Achieving SOA Value	323
SOA Value Considerations	324
SOA Value Points	325
Big Hairy Audacious Value	325
SOA Process	327
Process Approach to SOA ROI	327
Exploring the SOA Value Model	329
SOA ROI Threshold Model	332
Targeting ROI with SOA Initiatives	342
SOA and Services Reuse	346
SOA Metrics and SOA Scorecards	354
Summary	358

Index

Introduction to the SOA Business Model

S*ervice-oriented architecture* (SOA) is a concept whose time has come. SOA is garnering great hype for such a simple concept, and we are here to tell you that SOA is more than hype. It is a concept with great promise for your information technology (IT) operations, for your business operations, and for your organization as a whole. We must remember, though, that SOA is a concept. Before we put our simple definition of SOA on the table, let's discuss what SOA is not.

SOA is not a product. SOA is not a solution. SOA is not a technology. SOA cannot be reduced to vendors' software products, much as they would like you to believe. SOA is not a quick fix for the IT complexity that has accumulated over 30-plus years. And finally, SOA does not address every IT challenge facing business and IT executives today. However, with proper planning and execution, SOA will deliver compelling business benefits to your organization in the short, medium, and long term. SOA is the right model for IT today, and for IT in the future. So, what is an SOA?

SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network. "Services" in an SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages.

ELEMENTS OF AN SOA

An SOA has many moving parts, not the least of which is the enabling technology that makes it work. The following list represents the essential ingredients of a successful SOA. Each ingredient is explained in the sections that follow.

- Conceptual SOA vision
- Services
- Enabling technology
- SOA governance and policies
- SOA metrics
- Organizational and behavioral model

Conceptual SOA Vision

An SOA is a business concept, an idea or approach, of how IT functionality can be planned, designed, and delivered as modular business services to achieve specific business benefits. The conceptual SOA vision includes clearly defined business, IT and architectural goals, and a governance model and policies to help enforce standards and technical requirements of the SOA over time. This is the definition of an SOA target state, the goal to be achieved over time.

Services

Yes, an SOA needs services, which as we said, means *all* possible services in the organization. Along with services comes a services design model to assure reusability, interoperability, and integration across all business processes and technology platforms. Services are the central artifact of an SOA. Services are the primary architectural asset of an SOA. As such, they merit significant attention throughout this book and throughout an organization's migration toward SOA through many projects and initiatives, each of which will most likely contribute services to the SOA over time.

Enabling Technology

While the technology of Web services and SOA generates lots of press, it is probably the easiest area to implement despite the vendor flux and standards volatility for various categories of technology solutions. The technology is essential to support realization of your SOA vision. However, the enabling technology is *not* your SOA. The enabling technology must be implemented to accomplish two objectives: (1) It must allow your services to operate reliably and securely in your enterprise in support of your stated business objectives; and (2) it must enable you to carry forward your existing IT architecture as well as enable your legacy systems to be leveraged to support your SOA goals. In many organizations, legacy main-frame systems and other applications are major contributors of services to an SOA.

SOA Governance and Policies

An SOA conceptual architecture cannot be realized unless it is communicated to the constituents of the SOA—business users, developers, architects, business and IT executives, and business analysts. In addition, communicating your SOA conceptual architecture to close trading partners is also advised. However, telling your SOA constituents what your conceptual architecture, vision, and goals are is one thing. Enforcing conformance to your SOA conceptual architecture, vision, and goals is another matter. SOA is not a big bang implementation model that we expect from large, packaged software applications. SOA is achieved incrementally through time at the project level by continuously defining and enforcing the standards that it will be based on. These standards are the policies that in the aggregate define your SOA conceptual architecture and, when implemented, help your organization achieve its SOA vision and business goals. An SOA governance model defines the various governance processes, organizational roles and responsibilities, standards and policies that must be adhered to in your SOA conceptual architecture.

Metrics

SOAs require a battery of metrics in order to measure the results you are achieving. These metrics include fine-grained metrics, such as service-level agreements (SLAs) for individual services, as well as usage metrics, policy conformance metrics, developer metrics, business and return on investment (ROI) metrics, and process metrics. Plan your metrics early, and don't forget them when you go live with services. You'll want the data, count on it.

Organizational and Behavioral Model

Your current IT architecture is the result of years of organizational behaviors, business decisions, and architectural choices. In order to achieve SOA, behavioral and organizational considerations must be understood and changed first; then over time will come gradual migration toward your SOA vision and goals. New organizational models and behavioral models will be essential to your SOA success.

SOA: BEHAVIOR AND CULTURE

SOAs contain a substantial amount of behavioral content because these initiatives are process-driven and span organizational boundaries. The “soft issues” of an SOA strategy must address the organizational issues and challenges that may help or inhibit SOA adoption, such as services ownership, the business and IT relationship, budgeting practices, and more. Organizational, cultural, and process issues thread through several facets of an SOA initiative. How do you organize your enterprise architecture functions and roles to support an SOA? How do you organize your developer resources to help ensure the realization of the goals and performance of your SOA initiative? What is the optimal IT structure for an SOA? Is centralized IT better? Or is a centralized enterprise architecture team optimal, supported by distributed developers embedded within specific business units? What are the skills, roles, and competencies of your architecture organization that will facilitate migration to and attainment of your SOA?

In addition, cultural and behavioral aspects are crucial to achieving SOA success. We will use a metaphor here. Imagine you're an

archaeologist. You're examining the artifacts of a long-since decimated culture—the physical remains, artifacts, tools, cooking utensils, and so forth—to ultimately make inferences about the behaviors that caused these artifacts to be frozen in their earthen matrix in the way that you've discovered them. That's what archaeologists do. They attempt to derive behavior from the physical remains and artifacts. Now, fast forward to your current IT architecture. It is comprised of legacy mainframes, distributed systems, desktop systems, software and documentation, user manuals, data models and schemas, which are all artifacts that resulted from the accumulated behaviors of your organization through time. These behaviors were a result of business and IT strategy and the various choices and decisions that caused your IT architecture to develop into its current state. So, if you plan to achieve SOA, you have to begin with behavioral, cultural, and other organizational factors that will lead to SOA success, and then architect your way toward SOA. You must enable and reinforce the behaviors that are more likely to result in the desired architectural outcome: SOA. If you start with enabling technology without changing behavior, years from now you'll end up with another layer of technology that an IT archaeologist will have to interpret.

Exhibit 1.1 depicts these elements of an SOA according to our model. As you can see, the SOA strategy drives the governance model and policies. Services are at the center of the model because they are the central asset and organizing principle of an SOA. They are the key asset of an SOA. The enabling technology surrounds the services, within the framework of the SOA governance model and policies. The SOA governance model also drives the metrics, the SOA architecture process, and finally the behavior and culture that must be addressed to ultimately realize the business and technology benefits of SOA.

Although these elements represent the essential ingredients of an SOA, there is much more to it. What most organizations will find is that they need new ways of managing various business and IT processes to meet the demands of an SOA initiative. This book represents a collection of models required to implement SOA. But why is SOA such an important concept now, and why is there so much interest in it these days? Simple. SOA offers too many business and IT benefits for business executives to ignore. Competitive advantage is at stake with SOA. First movers will have it; SOA laggards will not.

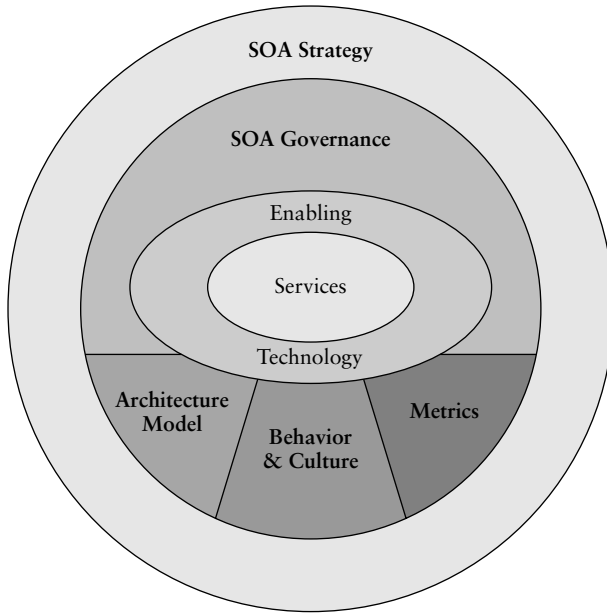


EXHIBIT 1.1 Elements of an SOA

NEW SOA CONCEPTUAL, ARCHITECTURAL, AND ORGANIZATIONAL MODELS

SOA initiatives will stress and in most cases break current operational and architectural models of IT organizations. SOA will require new ways of modeling and implementing various IT processes we have become accustomed to, such as services design models, integration models, reuse models, architecture processes, and enterprise architecture models. These other models, of course, augment the required SOA governance model.

SOA: ITS TIME HAS COME

One thing we know for sure, SOA is a concept whose time has come, and you do not need Jack Welch, Larry Ellison, Dr. Phil, or Oprah to tell you it's the right thing to do. If you are a business or IT executive and you are *not* thinking about how to implement an SOA in your

organization, you have already fallen behind your competitors. Your business and IT costs are higher. Your time to market is slower for new products and services. Your ability to implement IT solutions in support of business goals lags behind your competitors. And your legacy IT architecture is like a boat anchor embedded in the seafloor. You are at the mercy of the tides with no control of your destiny because you are beholden to your existing IT architecture. You are unable shed your legacy burden: the fixed costs, the outdated technology platforms, and the skills required to sustain it.

Over the years, your IT architecture has accumulated layer upon layer of complexity. When client-server architectures dominated the IT industry, client-server applications were layered over your mainframe platforms. When the Internet era rose to ascendance, Web-centric platforms were added on top of client-server solutions. And as these evolved into *n*-tiered architectures, the layers of IT complexity built up, building more modern complexity on top of legacy complexity.

Chances are you addressed the legacy systems problem with integration middleware, such as enterprise application integration (EAI) platforms or similar solutions. And this became yet another layer of complexity, or, as Brent Carlson, chief technology officer of LogicLibrary, calls it, “YALOT” (yet another layer of technology). These middleware or integration platforms were supposed to solve the legacy system integration problem and help simplify your IT architecture, but the reverse was true. These platforms became part of the very same problem, just more expensively and equally proprietarily.

Although IT architecture through the years is accretive, and nothing seems to ever go away, there are ways to “architect” your way out of this conundrum. One approach is to rip out all the legacy applications and replace them with modern ones. This rip-and-replace model is too expensive for most organizations. In addition, often it is not worth the risk and effort of replacing these still-working legacy systems with new software that will require significant modifications to suit your business model and business processes.

Another approach is to rewrite or refactor your legacy systems for modern application server platforms, such as J2EE or .NET. Even though rewriting systems is also very expensive, at least you know they will match your business processes when you are through. Like the rip-and-replace model, this approach usually is avoided, however,

not because it is not the right thing to do, but because it is expensive and difficult to cost justify to the business.

But there is a way out of this mess that avoids big bang system rewrites and expensive enterprise software projects. The way is *service-oriented architecture*. SOA is a simple concept, one that has the potential to alleviate many long-standing IT challenges and enable many coveted business goals that have until now been very elusive. SOA, by introducing a services layer into an existing IT architecture, can provide opportunities to isolate areas or elements of the architecture that are problematic, failure prone, or cost prohibitive. The services layering approach can enable the isolation, replacement, and/or potential consolidation of these architecture challenges while enabling the flexibility of reusable services. How often have business executives pronounced a desire to become more agile? When has time-to-market *not* been a mission-critical business requirement? Yet more often than not, these lofty business goals are constrained by outdated IT systems and incapable business processes that are subservient to tradition as well as to the digital concrete of today's enterprise software applications.

Why SOA Now?

SOA offers an avenue out of myriad business and IT challenges. However, before you leap into the SOA fray, you must understand a few things about it. First, SOA is not a new concept. It has, however, been refreshed with the advent of Web services, which have achieved more consensus from the vendor community than has been achieved in the history of computing.

SOA has also achieved trend status because of the degree of dissatisfaction that IT and business executives share with the current state of IT within their enterprises. Chief executives (CEOs) are fed up with hearing why they cannot expand into a new geographical location because the IT systems are not ready yet. They don't want to hear why the enterprise resource planning (ERP) system that cost \$30 million will not support the new business process targeted to launch in six months. Chief financial officers (CFOs) are tired of waiting for regulatory compliance issues to be resolved, and they certainly are not pleased with an overbudget IT organization. Chief operating

officers (COOs) resent being told they cannot get a report because data are spread across three different systems, all on different computing platforms. Chief technology officers (CTOs) are fed up with vendors pushing more new technology when the old technology is still underutilized and operating as islands of functionality. They are tired of the endless need to keep integrating systems when the integration models themselves become part of the problem—more legacy silos to maintain. And chief information officers (CIOs)? They are tired of explaining the same problems time and time again. They're tired of having their budgets cut. They wish they had more funding for strategic projects instead of being hamstrung by having 80 to 90% of their budget committed to maintaining legacy systems. CIOs could do much more for the business if they could shed their legacy systems and focus on forward-looking strategic solutions for the business. There has to be a way out of this quandary. SOA could well be an answer. SOA is not new, but it's here to stay. SOA is finally achievable thanks to three major factors:

1. *Standards consensus.* Microsoft and IBM agreed, and the rest fell in line.
2. *SOA enabling technology.* Finally, implementing standards-based services is possible and affordable.
3. *Integration fatigue.* There has to be a better way to achieve application and business integration.

Standards Consensus

For the first time in the history of IT, there is widespread agreement on major SOA and Web services standards by all IT vendors. This nearly unanimous agreement means that whether you move now or later, you most certainly are going to be using services in your organization. Your software and platform vendors are going to take you there whether you want to go or not. Our advice? Preempt your vendors with your own SOA strategy and roadmap. Rapidly accumulate SOA experience. And be prepared to fend off any proprietary platform-specific approaches to services. Implement industry standards in your SOA governance model and in specific policies that will govern services identification, design, and implementation. You may

have to dedicate some internal resources to tracking relevant standards, but the ROI on standards will be well worth it.

SOA Tools and Infrastructure

With the advent of new tools and infrastructure solutions that enable SOA and services in a cross-platform, reusable, and interoperable fashion, SOA is real. This is perhaps the most significant departure from previous SOA implementations, such as CORBA (Common Object Request Broker Architecture), COM/DCOM (Common Object Model/Distributed Common Object Model), DCE (Distributed Computing Environment), and other proprietary schemes for reusable services. Interoperability for services is largely due to the standards for Web services, primarily SOAP for messaging and WSDL (Web services description language) for service descriptions. The variety of tools for legacy systems enablement, services development and exposure, Web services management, and multiple run-time environments for services have made the SOA industry very interesting to watch. There are as many ways to enable services and SOA as there are legacy systems and platforms in your architecture. Of course, bear in mind that we refer to the general case of “services” in this book. Some of your services will be Web services based on XML (eXtensible Markup Language), SOAP, and WSDL, as well as the extended WS-* standards. However, do not limit your SOA total value by examining and considering Web services only. Think services first, and then specific implementation models later.

Integration Fatigue: “There Has to Be a Better Way”

IT “business as usual” is over. Business and IT executives are frustrated with the lack of integration of their internal systems, with their business processes, with their trading partners, and with their customers. We call this *integration fatigue*. The business is demanding change, and IT executives know there has to be a better way. The frustration with IT as we know it is at an all-time high. IT budgets continue to be stressed, and they rarely increase. The majority of IT budgets are focused on maintaining current systems and keeping the

lights on; very little IT budget is focused on strategic initiatives that may pay future dividends. It is a do-more-with-less environment.

Business continues to change while IT is saddled with maintaining the systems and architectures of the past. IT doesn't have the luxury of eliminating its legacy or underperforming assets. Those very IT assets contain business logic that is most likely running a mission-critical portion of the organization's business. Yet while the business logic is mission critical, nonetheless the logic and data are locked up within individual silos of systems and technology. You cannot afford to rewrite the application, and your integration strategy has proved to be supremely costly to implement and maintain. There has to be a better way, and there is. It's called *service-oriented architecture*.

SOA: EVOLVED ENTERPRISE INTEGRATION

A major impetus for SOA initiatives is solving the age-old problem of integration. For many executives, SOA holds the potential to eliminate, via industry standards and modern tools, the proprietary integration model they've become accustomed to. According to many analyst estimates, up to 30% of a typical IT budget is allocated to integration activities. What would business be like if there was less integration, or, rather, if the integration that was performed was directly related to process integration, enterprise integration, and mergers and acquisitions (M&A) integration? In other words, value-added integration. How would spending less money on integration change an organization's competitive advantage? Could that budget be shifted to more strategic projects?

Origins of the Integration Problem

Where did all this IT complexity come from? Why is 80 to 90% of your IT budget focused backward on maintaining the past rather than looking ahead to supporting the future? This "rearview mirror budgeting" problem is legendary among CIOs and is partly responsible for the lack of strategic IT investment by CIOs today. How backward committed is your IT budget? What percent of the IT budget is allocated to maintaining your legacy investments rather than focused

on forward-facing initiatives that may move the organization ahead? This is a real challenge for both business and IT executives today. If you feel as if you're managing your IT budgets using a rearview mirror, you're not alone.

The demand for IT and process integration is driven by business requirements, such as:

- Increased M&A activity
- Corporate reorganization or restructuring
- Application and/or system consolidation
- Data integration and data warehousing initiatives
- New business strategies leveraging current systems for new processes
- Achieving regulatory compliance (e.g., Sarbanes-Oxley, or HIPAA, the Health Insurance Portability and Accountability Act of 1996)
- Streamlining of business processes to improve productivity

Addressing the business drivers for integration is a great impetus for SOA and Web services. At what point does IT complexity become an obstacle to business goals and an impediment to achieving IT's goals? We believe that complexity becomes intolerable when organizations are considering or taking these actions:

- Hiring a chief architect
- Creating a central architecture team
- Acquiring or developing your own enterprise application integration (EAI) software
- Creating an internal integration team, or a middleware organization, to help solve the integration challenges for your organization

Now, this does not mean that hiring a chief architect is a bad thing. More than likely it is a good thing. Chief architects can help address the architectural complexity and pain your organization is facing. The other actions just listed also can help address these areas. The problems are symptoms of the IT challenges that SOA could address. If you have considered or taken one or more of these actions, your organization is at the point where the integration burden is consuming IT resources, compounding the existing complexity problem, and inhibiting business and IT effectiveness. You most

likely have a rearview-mirror IT budget, and you are ready to try a new approach.

Stop Integrating; Service-Enable Instead

Stop integrating now. What we mean is stop integrating the way you have been using proprietary middleware solutions, homegrown point-to-point integration techniques, and tactical integration approaches that are doomed from the outset. These techniques are almost always going to break and require a significant ongoing maintenance burden from the organization.

An organization should integrate without integrating. Eliminate all point-to-point integration projects in your enterprise and rearchitect these initiatives from an SOA point of view. Inventory the integration solutions in your organization. Identify the IT budget allocated to these solutions and projects, including support staff, maintenance, and infrastructure. Determine how many of these integration efforts can be eliminated using reusable services in an SOA. (Integration projects are very good services opportunities. Chapters 3, 4, and 5 discuss the process of identifying and modeling services opportunities for an SOA initiative.) Identify the consumers of these integrations and determine their satisfaction with the current approach. How often do these integrations break? How often must they be enhanced or modified to support changing business needs? Do the n^2 math to determine how many interfaces can be eliminated with a services approach to integration. There will be a cost savings when comparing SOA and services to your current integration strategy. A services approach will be a more flexible, reusable approach than the point-to-point model you've been using. Although service-oriented integration will require more discipline and planning than previous integration paradigms, the results will be well worth the investment. Stop integrating now. Service-enable instead.

Exploring an SOA Business Scenario Exhibit 1.2 depicts a hypothetical approach to SOA as an alternative to conventional integration. This exhibit shows a fictional insurance company with three business units—Group, Voluntary, and Individual—at the top. Corresponding to the three business units are their own collections of systems and

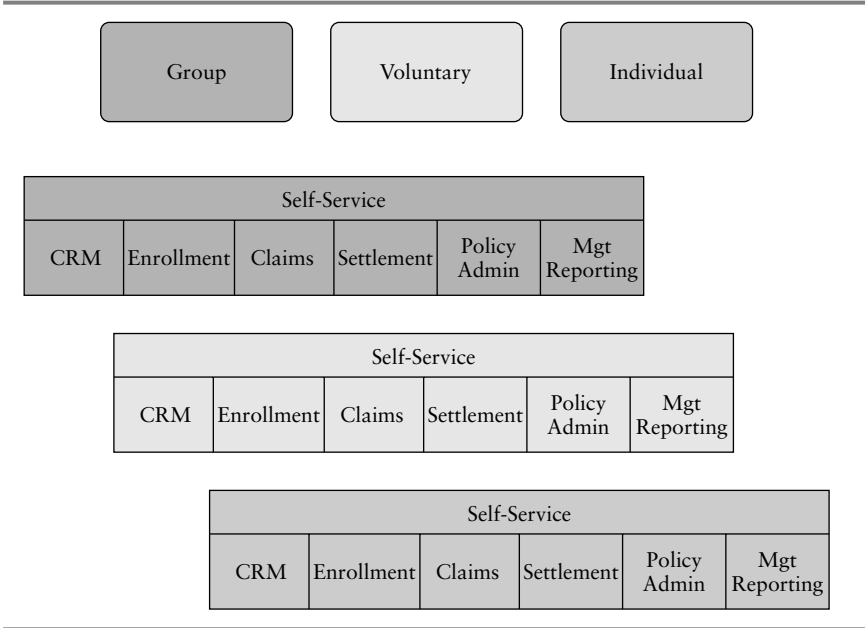


EXHIBIT 1.2 Typical IT System Silos

duplicate IT capabilities, accumulated over the years, and purportedly so unique that the business units must have their own systems to accomplish the very same business processes.

Each business unit has duplicate systems for sales and contact management, enrollment, claims, settlement, policy administration, and management reporting and controls. On top of these, all the business units have portal, intranet, and extranet capabilities to allow self-service for customers, agents, and brokers.

Identify SOA Business Opportunities Exhibit 1.3 examines this hypothetical business from an SOA perspective, exploring the potential business value that SOA may bring to this enterprise. What if the duplicate IT systems and business processes could be integrated and united as shared services by all three business units? What if the business processes across the three business units could be simplified to take advantage of shared IT services in an SOA model?

As shown in the exhibit, SOA offers several potential business benefits to this organization, such as product and process simplification,

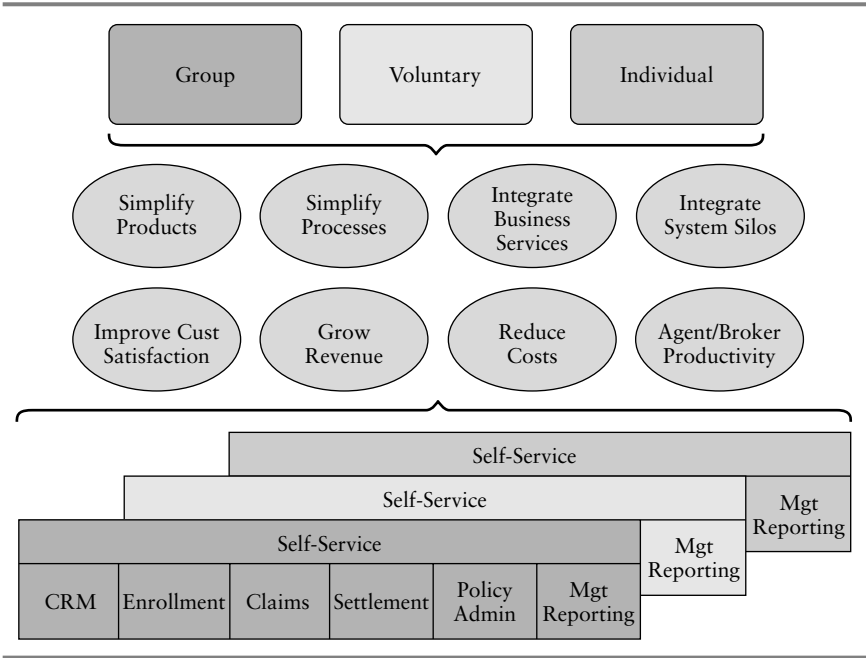


EXHIBIT 1.3 Identifying SOA Business Opportunities

integrated systems, integrated business services, better customer satisfaction, cost reductions, increased revenue and margin, and better agent and/or broker productivity. Of course, you cannot realize these potential benefits without performing proper analysis.

Examining the SOA Information Technology Potential Continuing the example, Exhibit 1.4 shows specific SOA opportunities that may apply to this insurance company. These opportunities include integration, process orchestration, better interoperability, services reuse, improved IT productivity, and achieving a real-time, event-driven enterprise.

Again, proper analysis and SOA planning of the business and SOA opportunities will determine the unique SOA value that will apply to any given organization.

Seeking Reuse Value from an SOA Initiative Let us assume that services reuse is a key SOA driver for this insurance company. Our SOA strategy, planning, and analysis will identify significant services reuse benefits in many areas of this company, across the multiple business units,

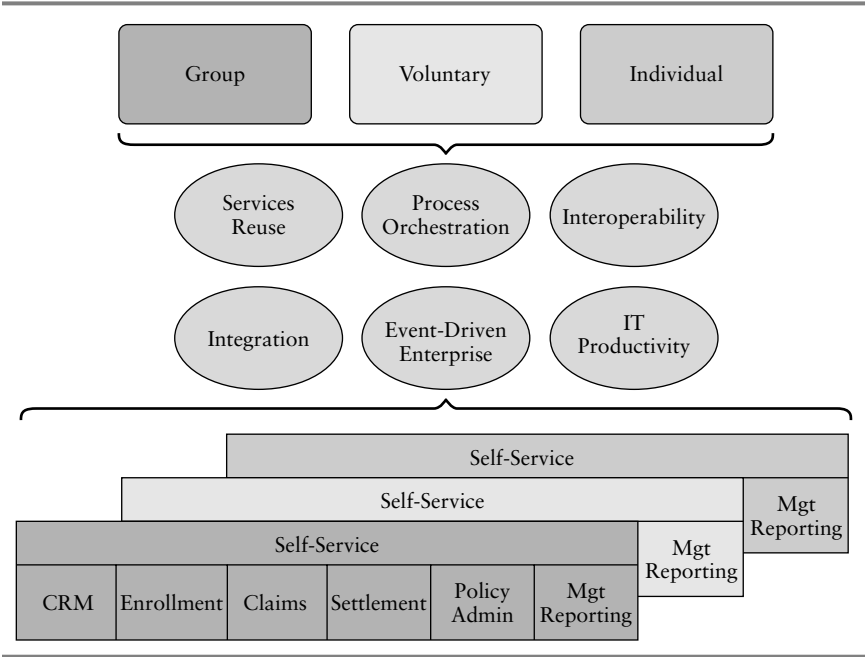


EXHIBIT 1.4 Identifying SOA Opportunities

and across the IT systems that support them. Exhibit 1.5 focuses this example on services reuse. In this insurance company, it was determined that reuse was essential to reducing costs, increasing IT productivity, and improving responsiveness to business demands.

This example shows how analysis might show that reuse of specific IT capabilities may offer tremendous business value. In this scenario, we focus on the enrollment process, which is supported by an excellent system but which can be improved by exposing its capabilities as shared reusable services in an SOA. Doing this allows reuse of a single IT service, enrollment, across three business units. This reuse eliminates IT complexity, increases IT productivity, and leads to simplified business processes in this organization. SOA applies as much to business processes and enabling better business functionality as it applies to IT systems—eliminating redundant systems, duplicate support infrastructure, and fragile and expensive integration strategies.

Service and Process Orchestration in an SOA Furthermore, this same organization is able to leverage its reusable services by orchestrating them

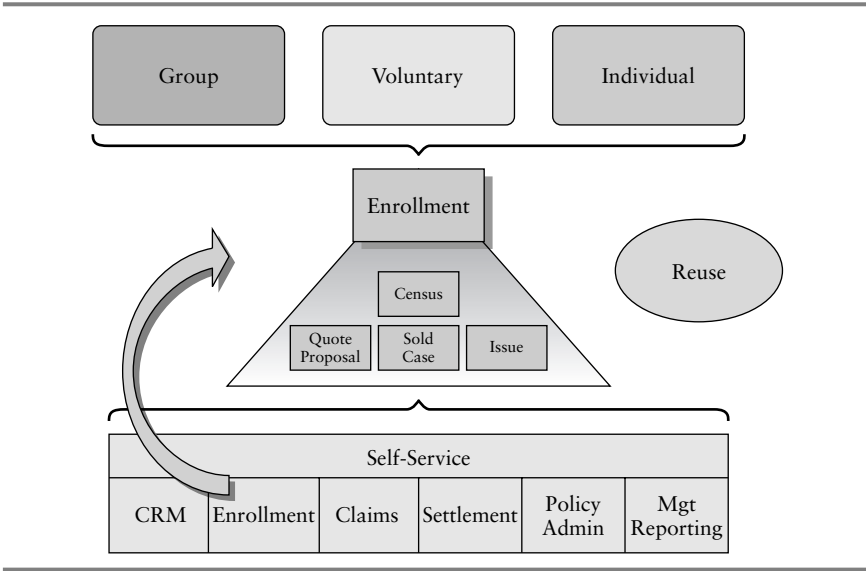


EXHIBIT 1.5 Services Reuse in an SOA

into business processes. Service and process orchestration extends reuse benefits by leveraging services as reusable composite applications and orchestrated process work flows. Exhibit 1.6 shows the enrollment process as a simplified series of services orchestrated into the business process work flow that the company decides is most reflective of how it wants to operate.

SOA allows an organization to stop integrating and instead recast its IT capabilities as shared reusable services. Once there are enough services to reuse in an SOA, further value can be harvested by orchestrating business processes based on these reusable services. In addition, the integration challenges that have historically plagued IT organizations can be avoided. Implementing the SOA scenario just described enables an organization to realize the many business benefits of SOA.

SOA: Competitive Advantage via Services

SOA is a concept that has direct business and competitive advantage implications for all organizations. For the business, SOA means increased customer satisfaction, real business agility, faster time to

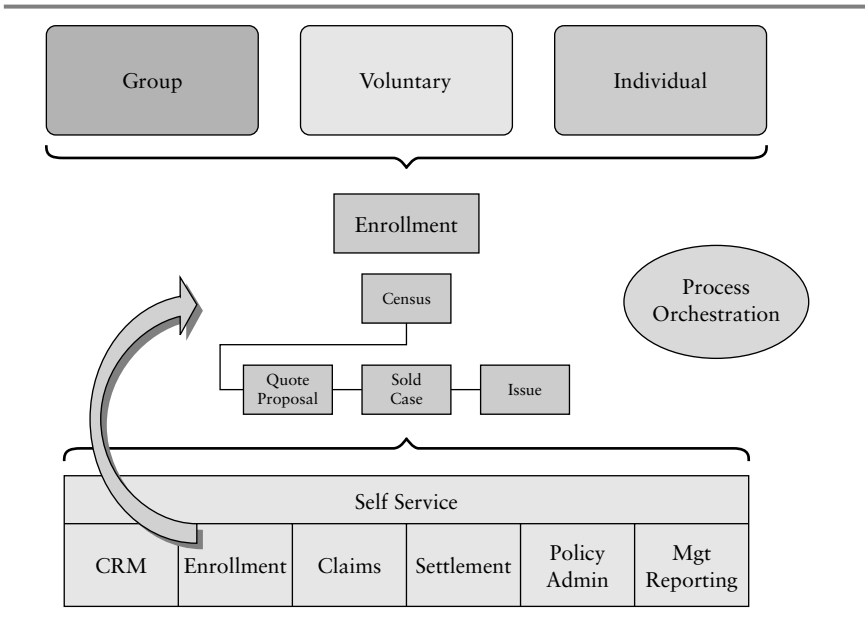
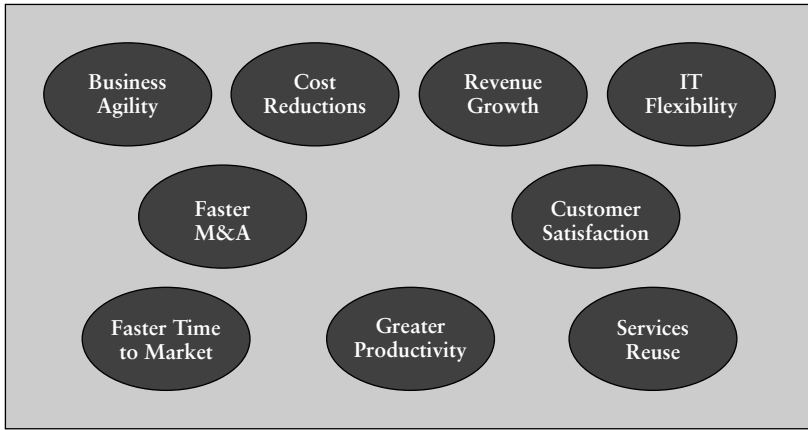


EXHIBIT 1.6 Orchestrating Processes in an SOA

market, ease of partnering, and lower business costs. Imagine that you can launch new products and services 30% faster than your competitors because you have eliminated friction within your enterprise, allowing better collaboration between your suppliers and your design engineers as well as better collaboration with your channel partners.

Your SOA has allowed you to speed up IT delivery to your business consumers. The time to implement needed system changes to support these new products has been cut by 25%, and you are able to make these changes using fewer resources: less development resources, less quality assurance resources, and less overall IT resources. Exhibit 1.7 depicts typical business benefits of SOA.

For IT organizations, SOA means greater productivity, faster time to market, greater asset reuse, agility, and lower IT costs. What if you could deliver IT services with less budget and few resources, while providing faster, higher-quality application support? SOA benefits an IT organization through faster application development, lower overall costs, greater software asset and services reuse, more

**EXHIBIT 1.7** Business Benefits of SOA

repeatable software development processes, higher-quality applications via pretested components and validated Web services interfaces, and overall faster response to business customer requests for system enhancements and modifications.

SOA benefits the business through greater flexibility, faster time to market for business initiatives, faster response to business changes, and closer mapping of IT services to business needs. Many analysts see SOA as a mechanism to help finally achieve alignment of business goals and objectives with IT services and capabilities. Better alignment is partly attributable to the speed with which Web services can be developed and deployed in an SOA as well as the flexibility from leveraging proven, tested software components and Web services.

SOA also benefits an organization by abstracting business services (or Web services) from the specific technologies they were originally developed with and the platforms they were meant to run on. This twofold abstraction has two key benefits: (1) an organization can modify the technology architecture without mandating changes to the services available, and (2) the business community can change business processes without causing the ripple effect of changes to the services and underlying IT systems. Doing this leads to greater business agility and IT flexibility.

SOA MEANS "SERVICE-ORIENTED AGILITY"

SOA holds promise to finally make the word "agility" real for organizations. That's why the "A" in SOA, which stands for "architecture," could just as well stand for "agility." Service-oriented agility is one of the most oft-articulated goals of SOA. SOA enables business and IT agility along a number of dimensions. Although nearly every business and IT executive for the last 30 years has wistfully dreamed of achieving business agility, there has been little real progress toward that end save for a few exceptional firms. For most organizations, business agility is a vision without reality. Until now. SOA and services provide a means to achieving true business agility. Business agility can come from two broad forms: the ability to change business processes to meet changing market demands and customer requirements, and reduce costs; or the ability to execute business processes faster or launch new processes, products, and services faster. Agility and speed are both real and tangible benefits of migrating to SOA and reusable services.

SOA can help an organization unshackle its business processes and data from the IT systems that support or, in many cases, constrain them. Using a services approach and SOA, enterprise software systems will be decoupled from business processes through the use of business services. Business services will be defined as abstracted entities separate from the business logic that is locked within enterprise applications such as SAP, Oracle, Siebel, and other monolithic enterprise applications. When application logic is exposed as a business service, it becomes a shareable and reusable software asset, or *service*, that can be coupled with services from other applications to create new sources of business process value, completely new business processes, and even more efficient versions of existing processes using business process management (BPM) tools.

Service-Oriented Agility: Speed

One aspect of agility is speed. The ability of an organization to hasten its response to market changes or competitive threats, or to quickly preempt competitive moves from the competition, is clearly an advantage. Speed consists of two dimensions: the total elapsed