Chapter 1

Getting to Know SOA

In This Chapter

- ▶ Finding out why you should care about SOA
- ▶ Liberating business from the constraints (and tyranny) of technology
- ▶ Illustrating the need for SOA
- ▶ Saving bundles by using what you have
- ▶ Expanding your SOA to customers, partners, and suppliers
- ▶ Focusing on function

Service oriented architecture (SOA) is a hot topic being bandied about by IT vendors across the globe. IBM, Hewlett-Packard, Software AG, Oracle, SAP, and Microsoft (just to drop a few names) are all singing from the SOA songbook, and hundreds of vendors are adding their tunes as we speak.

"What's SOA?" you ask. We suspect that you've already skimmed a dozen articles and read (or deleted) hundreds of e-mails from vendors pushing SOA, but the answers you've gotten so far have been, well, vague and inadequate.

The short answer is that SOA is a business approach to building IT systems that allows businesses to

- ✓ Leverage existing assets
- ✓ Create new ones
- ✓ Easily enable the inevitable changes required to support the business

For you impatient readers out there, we expand on this short answer in Chapter 5. However, right now, we think the more important question is, "Why should I care about SOA?" In this chapter, we try to answer this question.

The promise of service oriented architecture is to liberate business from the constraints of technology, unshackling technologists and business leaders from the chains they themselves have forged. ("IT workers of the world, unite! You have nothing to lose but your chains!" as it were.) This has major implications both for the business and for the IT structure that supports the business.

From our perspective, one of the most important aspects of SOA is that it's a *business* approach and methodology as much as it is a *technological* approach and methodology. SOA enables businesses to make business decisions *supported* by technology instead of making business decisions *determined* by or *constrained* by technology. And with SOA, the folks in IT finally get to say "yes" more often than they say "no."

TIP

We pronounce *SOA* to rhyme with boa (***bow**-uh*). Stretching it out by clearly articulating each letter (S-O-A) is perfectly acceptable but might leave you stymied when we say things like, "SOA what?"

Business Lib

Executives have come to rely on technology — in terms of reporting, text analytics, projections, graphical representations, risk analysis, and other analytical tools — to make informed decisions for their company. The day-to-day operations of a company have slipped, little by little, into the hands of IT. Quite simply, more and more of the activities of an organization are supported by increasing levels of business process automation — whether its business is to build ships, sell insurance, or manage cities — and since IT implements the automation of business process, business decision makers have become more dependent upon IT. While this increasing use of technology has helped the business in so many ways, technology has also created significant constraints. At many companies, business and IT management operate in very separate worlds without the benefit of a common unifying language. Unfortunately, as organizations become more diverse and complex through mergers, acquisitions, globalization, and the need to manage lots more data, the supporting IT infrastructure has become more cumbersome and brittle after being stretched in so many different ways to keep up with all the changes. This is not good for business, and neither is it good for IT.

We're not advocating that business leaders should (or can) take control of the technology from the hands of IT. Modern businesses are inextricably tied to technology. No sizable business can function without IT — it's as simple as that. However, we are advocating a new world order. Indeed, we advocate that business leaders and IT work together to create this new world order. *Together,* business leaders and IT will communicate how the automated processes of its business should be facilitated, and work together to make it a reality by using SOA. *Together,* IT and business leaders can determine a strategy that both liberates business from IT and allows IT to create maintainable, extensible, compliant systems to support the initiatives determined by business leaders.

Tech Lib

Just because business has become constrained by technology, don't think that the folks in IT are having a jolly old time basking in their new-found power. On the contrary, the IT staff gets to spend its time in endless meetings accounting for why projects are late, explaining why applications can't easily be adapted to changing business conditions, and pleading for more staff. When some clever marketer presents a new concept for selling more widgets via the Internet or mobile devices or some other new channel, IT management is always the wet blanket, having to explain why (despite the company's investment in all the latest software and hardware) it will take 18 months to implement the new plan.

With SOA, business and IT have a way to meet each other half-way and collaborate using a business focused approach to develop new ways to use technology to grow the firm, help to spot new trends and opportunities, and see new ideas to fruition. But before you go marching off to save the world, though, we have some more explaining to do. A story will help.

A SOA Case Study

Once upon a time, there was an insurance company called ABC Insurance Incorporated. When ABC was born — oh, maybe 150 years ago — it began by selling insurance policies to factories and manufacturers. In those days, there were no computers to mess things up. The company followed business processes that were pretty simple. A nice person sent a letter inquiring about a policy. A smart person set a rate, sold a policy, and hoped that nothing caught fire or blew up. ABC thrived for more than 100 years.

But then, things got complicated. Other companies started stealing ABC's business. Customers were asking for insurance for different kinds of risk. ABC had to change or die.

ABC was an early user of punch-card accounting systems. In the 1960s, ABC bought computers, hired programmers, and built software applications to support its business. In the 1980s, it bought software packages from different suppliers to help it continue to compete. It bought or built business applications to solve problems all over the company — one at a time. For example, it bought an application for the corporate finance department, created one to handle customer claims, and procured other applications to manage research information about what type of accidents were most common under what circumstances.

This worked well for many years, until the 1990s, when ABC found itself competing against financial services companies who decided *they* could sell insurance, too. Suddenly, ABC needed to find new ways to compete so it could sell a larger variety of products to current customers and also find some new customers. Its leaders thought up exciting new solutions based on the knowledge of their business and their customers.

In addition, management thought ABC could expand its business by acquiring other insurance companies with complementary products. ABC could sell these new products to existing ABC customers and sell ABC's products to the customers of the companies they acquired. These smart guys and gals understood business strategy. Everyone got really excited until . . .

Management talked to IT, and IT said, "This is really, really exciting, but we have a *small* problem."

"What could it be?" asked management.

"It's this," said IT. "We can no longer simply buy or build more software applications to implement our innovative plans for new products and services. The business policies and processes that we follow have become more complex. Everything we want to do has to work in concert with what we already have. The very running of our company depends on all the business applications that we built and acquired over years working together smoothly — such as the programs that tally the premiums people pay; administer the claims we process; and make risk analysis, payroll, invoicing, and sales commission calculations. When you come right down to it, our company is the aggregation of all our programs. Everything we need to carry out our day-to-day business functions — including information about our customers, our products, and our risk performance — is locked inside these programs and processes."

"Well," said management, "You can just write new programs to tie everything together. We'll *integrate,* and we'll all be very happy."

And IT said, "Yes, it is possible to *integrate,* but integrating will take *a very, very long time:* at least 18 months. Maybe two years. And by then, you might want more changes that will take another 18 months or two years. By then, it might be too late. And," IT continued, "*it will cost lots and lots of money.*"

Management and IT were very sad. They knew that ABC wouldn't survive if they couldn't find a new way of thinking about business process and technology. So they began asking everyone they knew of any way to save ABC. They searched, and they studied, and they prayed — until one day, a package arrived. In that package were several copies of a yellow-and-black book titled, *Service Oriented Architecture For Dummies,* 2nd Edition.

Both management and IT took copies of the book and read. They were very excited to discover that they didn't have to throw away valuable assets and that they could reap benefits in a short time. In the end, they came up with a *new* strategy, one based on five key elements:

- The IT organization will partner with the business managers to create a high-level map of the business processes, followed by each line of business. This will help identify the similarities, differences, and interrelationships across the business lines for the company.
- The IT organization will create a flexible structure that will turn key IT software assets into reusable business services that can be used no matter how the business changes. These business services will include everything from business processes and best practices to consistent data definitions to code that performs specific business functions.
- The IT organization will begin replacing the hundreds of redundant business services locked in old software with these new reusable services.
- The IT organization will use only accepted industry standards to link these software assets.
- The IT organization will use the service oriented architecture concept described in the rest of this book to begin to create business services that are consistent with how the business operates.

Together, management and IT began a journey. As far as we know, they are living happily ever after. In Part V, we give you many real-life case studies from companies you might recognize that indeed are alive and well and living happily on their journey to SOA.

Better Living through Reuse

One of the biggest deals in the SOA world is the tenet that you don't have to throw away things. You take the stuff (software assets) that you use every day — well, the *best* of the stuff you use every day — and package it in a way that lets you use it, reuse it, and keep on reusing it.

One problem common of many large companies that have been around for a while is that they have lots of similar programs — software applications — representing commonly used business processes. Every time a department wants something slightly different, that department builds its own version of the software so that across a particular company, you might find umpteen versions of more or less the same processes — with, of course, slight variations. Many IT shops have policies and procedures designed to prevent this sort of thing, but when deadlines loom and budgets are tight, it's often easier and quicker to write something from scratch that fills the need rather than to coordinate with other divisions. This sort of duplication becomes a nightmare when one company acquires another and finds that they have similar (but not identical) applications purporting to do the same thing.

These slight variations are precisely what make systems very complicated and expensive to maintain — even one business policy change might affect lots of different applications. In situations like this, it's very difficult to find every instance in every application that needs to be changed. The testing required for this type of application change management takes time away from more innovative development work and can inhibit businesses from getting to market quickly with new products.

With SOA, these important business processes — such as creating an invoice, calculating an interest rate, securing a reservation — become business services. Briefly, a *business service* is a sealed container of software code that describes a specific business process that can be connected to other business processes. (We talk more about this in Chapter 5.) You end up with one single business service for a given function that gets used everywhere in your organization. With SOA, when you need to change a business policy, you change it in only one place. And, because the same service is used everywhere, you have consistency throughout your organization.

For example, you know that if you decide to create a new department in your organization, you're not going to create new Accounting, Human Resources, Legal, Cleaning, Training, and Travel departments to go along with it. Even if you need to add staff, you'll likely use your existing Accounting, HR, Cleaning, Training, and Travel departments to *service* — note the word *service* — this new department.

The problem is that over time, IT ends up embedding redundant function in programs everywhere in the organization. That redundancy — just like having separate Accounting, HR, Legal, Cleaning, Training, and Travel departments for every department — is what SOA ultimately eliminates. This lack of redundancy gives you the same obvious benefits of scalability, consistency, and maintainability.

With SOA, business managers work with IT to identify business services. Together, they determine policy and best practices. These policies and best practices become *codified business services* that represent honed company business processes. No need, for example, to have 30 different variations on an exchange rate translation application, each used by a different department and all requiring IT time for ongoing maintenance. One business service will do. Onward, the new world order!

Moving in Tandem with SOA

In any formal dance, from the cha-cha to the waltz, form matters. The *form* is what allows you to dance with someone you've never met. When both partners truly know the form, they move in tandem, are flexible, and navigate with ease and grace.

SOA is form. It enables the business to move, change, partner, and reinvent itself with ease and grace. In the beginning, mastering new steps requires focus and attention. Over time, the steps become second nature.

Implicit in the notion of form is standards. Using industry standard interfaces and creating business services without dependencies (more on that later, we promise) allows the business vastly more flexibility than it enjoys today to change its business model, reorchestrate itself, and partner dynamically.

Redundant reiteration, again

For any IT old-timers out there who have labored long and hard in the IT trenches, the concept of software reuse isn't new. You're familiar with subroutine libraries and the great theme of object orientation, and you extol the virtues of standardization. "What's the big deal with SOA?" you ask. "Aren't we already doing this?" Well, yes and no. Yes, because the world of SOA depends on a good understanding of reuse and on the building of reusable components. No, because SOA extends the idea of reuse not only to *Web services* but also to *business services.* (For definitions of *business services* and *Web services,* look in Chapters 5 and 6.) In the world of SOA, the level of granularity shifts profoundly. No longer are we talking simply about reusable low-level components: We're talking about reusable high-level business services. This shift, and its implementation, is no mean feat either for business managers or for IT, but the rewards for everyone are dramatic.

Here's a real-world application. Electrical appliances that you plug in at home today plug in equally well at the office or if you move across town. If you travel abroad, though, you likely need electrical adapters. When standard interfaces don't agree, you must adapt. Likewise, working with industry standards set forth by standards bodies enable autonomous entities (partners, customers, and suppliers) to dance at the ball.

Sweeping Unsightly Technology under the Rug

In the next chapter, we talk a lot about architecture. For those of you who already know a lot about systems architecture and want more nuts and bolts, we suggest you skim quickly through the "conceptual" chapters in Part II to make sure you understand what we mean by the terms we use. Then dive headlong into Part III, which we promise puts meat on the bones and gives you a lot to chew on — metaphorically, of course.

One big reason we think business managers are going to like SOA is because business gets to focus more on business and less on technology, SOA technology has the potential to become more invisible at the business layer, like the plumbing in a well-designed home. In this chapter, we give you an overview of what the business can expect from SOA.

REMEMBER

SOA enables business managers and IT to talk in business terms that both sides understand. Without SOA, the IT developer and business manager typically use very different words for the same process: for example, creating an invoice. The IT developer is concerned with APIs (application program interfaces) and how to go about creating customer records from ten different Oracle database tables. The business manager describes the actual *business* process used to create an invoice. With SOA, a business service is a business service is a business service. How that business service is implemented in the technology layer is the purview of IT, and business managers need not worry about it or its associated technical jargon.

Understanding Why SOA Is Different

Perhaps you're skeptical. Perhaps, for as long as you can remember, the software industry has been promising yet another silver bullet to rid you of all business woes. We think now's a good time to repeat that SOA is not about "out with the old, in with the new." SOA is about *reuse:* taking what you have and structuring it to allow you not only to continue to use it, but to use it securely knowing that future change will be simple, straightforward, safe, and fast. SOA is indeed a journey; it can't be built overnight. But organizations can begin SOA now and can benefit now. Ultimately, SOA renders a business more flexible — and IT more reliable, sustainable, extensible, manageable, and accountable.

We think SOA is the most important mandate facing business and IT today. And because SOA is a joint venture between business managers and IT, we present the basics necessary for everyone to come to the table with a good grounding from a conceptual level.