

# Chapter 1

## SOA What?

---

### *In This Chapter*

- ▶ 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
- 

**S**ervice oriented architecture (SOA) is the hottest topic being bandied about by IT vendors across the globe. IBM, HP, BEA, 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 recycled a tree’s-worth of junk mail 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 new approach to building IT systems that allows businesses to leverage existing assets and easily enable the inevitable changes required to support the business.

For you impatient readers out there, know that we expand on this short answer in Chapter 2. However, right now, we think the more important question is, “Why should I care about SOA?” We try to answer this question first.

The promise of service oriented architecture is to liberate business from the constraints of technology and unshackle technologists 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 that supports the business.

From our perspective, one of the most important aspects of SOA is that it is 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.”



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

## Business Lib

One of the myths that plagues business today is that senior management is in charge. Yes, we know who holds the title, but a management title is a lot like the title to a car. The title is one thing, and the keys are another. And, although no one ever saw it coming, the keys to the business have been slipping, little by little, into the hands of IT. This is not good for business, and what is not good for business is ultimately not good for IT because without the business, IT ceases to exist.

Now, we are not advocating that business should (or can) wrest the keys from the hands of IT. Our 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. We are advocating that business and IT work together to create this new world order. *Together*, business leaders and IT determine how the business should operate and work together to make it a reality by using SOA. *Together*, IT and business leaders determine a strategy that both liberates business from IT and allows IT to create maintainable, extensible, compliant systems.

## Tech Lib

Just because business has become constrained by technology, don’t think 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, corporate geeks finally get to be part of the business adventure again, developing new ways to use technology to grow the firm, helping to spot new trends and opportunities, and seeing new ideas to fruition. But before you go marching off to save the world, we have some more explaining to do. A story will help.

## Once Upon a Time

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. 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 a hundred years.

But then, things got complicated. Other companies started stealing their 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 and 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 make money that didn't cost too much. Its leaders thought up exciting new solutions based on the knowledge of their business and their customers and through new, cool *technology*.

In addition, Management thought ABC could better compete by acquiring some 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?” cried Management.

“It is this,” said IT. “We can no longer simply buy or build more programs to implement our new moneymaking, cost-saving ideas. 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 have built and acquired over years working together smoothly — the programs to tally the money coming in, the programs to administer the claims processing going out, to do risk analysis, premium billing, payroll, invoicing, and sales commission calculation. When you come right down to it, our company is the aggregation of all our programs. Everything we need in order to carry out our day-to-day business functions — all our policies and information, including all the information about our customers — is locked inside these programs.”

“Well,” said Management, “You can just write new programs to tie everything together. We’ll *integrate* and we will all be very happy.”

And IT said, “Yes, it is possible to *integrate*, but integrating will take a *very, very long time*. Integrating will take at least 18 months, maybe 2 years, and by then you may want more changes that will take another 18 months or 2 years, and by then it may be too late. And,” IT continued, “*it will cost lots and lots of money*.”

Management and IT were very sad. They knew that ABC would not survive if they couldn’t find a *new way of thinking*. So they began asking everyone they knew if there was any way to save ABC. They searched and they studied and they prayed until one day a package arrived from Amazon.com. In that package were several copies of a yellow-and-black book. On the cover of the yellow-and-black books, they read *Service Oriented Architecture For Dummies*.

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

1. The IT organization will partner with the line of business managers to create a high-level map of what the business will look like.
2. The IT organization will create a flexible structure that will turn key IT software assets into reusable services that can be used no matter how the business changes. These services will include everything from business processes and best practices to consistent data definitions to code that performs specific business functions.
3. The IT organization will use only accepted industry standards to link these software assets together.

4. 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 the way the business operates.

Together, Management and IT began a journey, and, as far as we know, they are living happily ever after . . . In Part V, we give you many real-life case studies from real-life companies you may know 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 idea that you don't throw things out. 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 to many large companies that have been around for a while is that they have lots of similar programs. Every time a department wants something slightly different, the department builds its own version of that something so that, across a particular company, you can find umpteen versions of more or less the same program — 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 coordinate with other divisions. This sort of duplication also happens a lot when one company acquires another and finds that they have similar (but not identical) programs purporting to do the same thing.

These slight variations are precisely what make systems very complicated and expensive to maintain — if you make a change in business policy that affects the sundry applications, for example, you have to find and change each and every instance in every application that is affected. And even the slightest difference in implementation can result in inconsistencies — not a nice thing to find when those compliance auditors come snooping.

With SOA, these important programs become *business services*. (We talk more about this in Chapter 2.) 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 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 are not going to create a new Accounting department, new Human Resources department, new Legal department, new Cleaning department, new Training department, and new Travel department to go along with it. We trust that you will use your existing Accounting department (you may have to add staff), your existing HR, and your existing Cleaning, Training, and Travel departments to — note the expression — *service* this new department.

The problem is that, over time, IT — not those nice folks in the IT department today, but IT *over time* — ends up embedding redundant function in individual 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 will ultimately eliminate — giving 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*, impervious to the whims and fancies of errant engineers, audacious autocrats, tyrannous technologists, business bigots, and other such unsavory suspects. No more random acts of software. No more self-designated despots. Hail the new world order!

## *Dancing with Strangers*

If you dance any kind of formal dance, from the cha-cha to the waltz, you know that 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 later, we promise) allows the business vastly more flexibility than it enjoys today to change its business model, to reorchestrate itself, and to partner dynamically.

You feel confident that the appliances that you plug in at home today will plug in equally well at the office or if you move across town. You may also be aware that if you travel abroad, you will likely need adapters. You can plug in anywhere that the standard interfaces agree. Where they are different, you must adapt. Likewise, working with industry standards set forth by standards bodies enables autonomous entities (partners, customers, suppliers, hint, hint) to dance at the ball.

## Redundant reiteration again

For any IT old-timers out there who have labored long and hard in the IT trenches, the concept of reuse is not new. You're familiar with 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 2 and 3.) 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.

## Hiding the Unsightly

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 next few "conceptual" chapters to make sure you understand what we mean by the terms we've decided to use. Then dive headlong into Part II, which we promise will put meat on the bones and give you a lot to chew on — metaphorically of course.

One big reason we think business managers are going to like SOA is that, with SOA, business gets to focus more on business and less on technology. Like the plumbing in a well-designed home, SOA technology just works — it's there, but it is mostly invisible at the business layer. We show and tell you all about this in the next chapter, but right here in Chapter 1, we want you to consider what your life would be like if technology was not an obstacle but an aide in making your business act the way you want it to act.

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 to describe the process of creating, for example, 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. Really. Trust us.

## *Why Is This Story Different from Every Other Story?*

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*. SOA is about taking what you have and structuring it in a way that allows you not only to continue to use it, but to use it secure in the knowledge 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.