Where Do You Want to Go in Your SOA Adoption Journey?

Shrikant Mulik, Sushil Ajgaonkar, and Kavindra Sharma, *L&T Infotech*

Not every company has the same needs in adopting a service-oriented architecture (SOA). The authors compare four options for increasing system flexibility.

These days, almost every CIO or IT manager is looking into service-oriented architecture (SOA). A year ago, *The McKinsey Quarterly* surveyed 72 senior IT executives about their investment priorities and found that 64 percent were planning to implement SOA in 2007.[1] Forrester reported in February 2007 that 62 percent of its surveyed companies from the North American and European regions are either using SOA or planning to do so within the next 12 months.[2]

Interestingly, adopting SOA isn't the same as deploying a software application, which can be a one-time activity. Rather, it's a journey for an organization over a long period of time—an important detail for everyone involved to understand.

Although a few vendors claim that SOA can be quickly and easily implemented with their products, the overall understanding in the industry is that SOA adoption is an evolutionary, rather than revolutionary, process.

Beginning the Journey

Technically speaking, SOA is an architectural pattern that says that computational units (such as system modules) should be loosely coupled through their service interfaces for delivering the desired functionality. This pattern can be applied to the architecture of a single system (such as a quality management information or insurance claims management system) or the overall architecture of all applications in an en-

 Published by the IEEE Computer Society 1520-9209/08/$25.00 © 2008 IEEE

terprise. It's important to note that the services don't have to be Web services. They can also be Corba services or Jini services, though Web services currently represent the de facto technology for realizing SOA.

Certain SOA principles, such as loose coupling, ensure that systems can be highly maintainable and adaptable. However, two challenges arise. First, loosely coupled modules yield lower performance than tightly coupled modules. Another way to look at it is that loosely coupled modules would incur more hardware costs than with tightly coupled. The science of designing systems with SOA principles is also still evolving and so the service design remains largely an art.

Nonetheless, the benefits that SOA brings to the table can be quite appealing. In a world of increasing competition and constant transformation, SOA makes it easier to implement enterprise-wide changes by exploiting the inherent flexibility it offers. That means easily modifiable information systems, which is a top priority for any CIO. With this in mind, it's not surprising to see SOA rising to the top of many CIOs' agendas.

Does that mean that organizations should immediately start rearchitecting all of their information systems? In a word, no! Even before drawing plans for adopting SOA, it's important to decide the actual destination you're aiming for. Simply turning applications into services one after another might bring the benefit of flexibility, but planning it with a predefined destination will bring you the same benefit with less cost.

Considering what SOA can do for an organization, you should choose from among four destinations for your SOA adoption journey:

- reusable business services,
- service-oriented integration (SOI),
- composite applications, and
- a foundation for business process management (BPM).

The process can be multiphased. An organization can start off seeking any of these outcomes and later decide to aim for another, more advanced goal. At another extreme, a few organizations might directly opt for building a foundation for their BPMs using SOA. Whatever direction you choose, it's important to fully understand what each option entails.

Reusable Business Services

Organizations typically have some systems that supply core data—such as customer or product data—to the rest of the systems. Typically, development of any new system requires interaction with such systems. Given that such interaction tends to be tightly coupled, any change in core systems causes changes in many systems that the core system feeds. This change could ripple further if the systems became increasingly tightly coupled.

Simply turning applications into services might bring the benefit of flexibility, but planning it with a predefined destination will bring you the same benefit with less cost.

In such cases, exposing services from systems that provide core data becomes a good solution. For example, an employee information service with operations such as `getContactDetails`, `getPersonalDetails`, or `searchEmployeeByLastName` can act as a single source for employee-related data. Designing the interface of such services isn't an easy task because it needs to take into account what multiple users currently need for different tasks and what they might need for future integrations.

Many organizations need to connect portions of their internal systems to their business partners across the Internet or proprietary networks. Rather than working out different mechanisms for connecting with each partner, the organizations can design generic service interfaces that follow industry standards (such as the Accord system in insurance or Onix in the book-selling business). An example could be a catalog service that provides operations such as `searchItem`, `lookupItem`, and so on. Again, in the absence of industry standards, designing a service interface like this is a difficult task—but it can save a lot

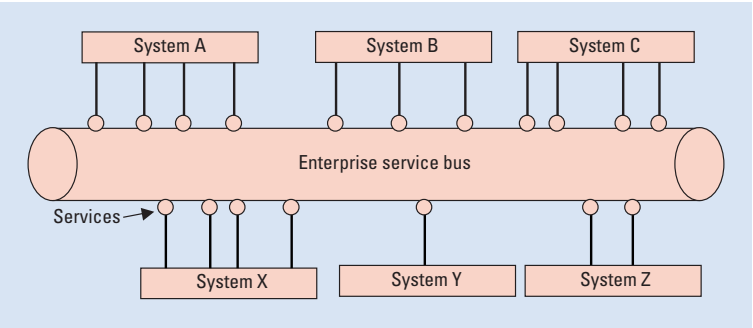

Figure 1. How you can use an enterprise service bus to connect or integrate your applications.

of money in the long run and provide agility to organizations.

Reusable business services are also useful when an organization has some of its application functionality developed using legacy technology. Rather than redeveloping such applications from scratch, we can wrap them as services for consumption by new-age technologies such as portals, smart clients, and mobile devices.

Service-Oriented Integration

Integrating internal applications has historically been challenging for IT managers because of heterogeneous platforms among applications. For many organizations, FTP remains a dominant mechanism for integration. Since the advent of enterprise application integration tools in early 2000, however, companies have addressed this challenge extremely well. Vendors such as webMethods, Tibco, and SeeBeyond (now part of Sun Microsystems) provide enterprise application integration (EAI) tools that can connect packaged applications and custom applications across the enterprise using either a single bus or a hub for all kinds of integration needs.

Perhaps the only shortfall of this approach is that these EAI tools have been proprietary. Once you deploy a tool from one vendor, it's difficult to switch to another. The answer to this shortfall came with the *enterprise service bus*. Simply put, an ESB is a software infrastructure tool that provides messaging, content-based routing, and XML-based data transformation for services to integrate. Consider it a lightweight EAI tool. Figure 1 shows a simplistic view of how you can use an ESB.

Many organizations have started using service-orchestration engines (SOEs) along with ESBs for integration. Such tools help us visualize business process execution as the orchestration of services. This helps us respond to business changes quicker because any business change can be translated into a change in business processes.

The Business Process Execution Language is the de facto standard language for this approach. BPEL is supported by almost all service-orchestration engines. This standard helps you switch one SOE with another, though with some effort. Many prefer an integration that combines ESB and BPEL engines, thereby using process-based rather than service-oriented integration.

Composite Applications

It's common to see duplicate application functionalities across many information systems. This occurs largely because application components are difficult to reuse if they're not properly structured. If you take care to review existing systems and consider how they can be developed into reusable business services, however, you can avoid duplication, excess, and having to start from scratch. Rather than developing isolated business applications, it's worth considering building new applications by reusing existing services and developing the rest of the functionality.

We can classify composite applications as either *static* or *dynamic*.[3] Static composite applications are built programmatically. That means programmers are required to write new code, which can connect to existing services. On the other hand, dynamic composite applications can be built using BPEL engines, which provide a GUI for orchestrating services.

A business analyst can also compose a new application by orchestrating existing services along with newly developed services. Such orchestration can be exposed as a service, thus making the service composition multilevel.

We should note that loose coupling is desirable for the user interface in such composite applications. That means that you could access a single composite application through multiple channels such as smart clients, portals, or mobile

devices. This offers some much-needed flexibility in adapting to users' ever-changing needs for accessing applications.

Foundation for BPM

BPM involves modeling, monitoring, measuring, and optimizing business processes' performance.[4] Because most business processes are digitized, it's now easy to enable BPM via software tools. Such tools have been available from vendors such as Savvion for a long time. However, because these tools tend to connect to existing applications in a proprietary manner, you might end up with a tight-coupling problem. To avoid this, we recommend fully adopting SOA and then building the BPM infrastructure on top of it. This also eases the implementation of BPM tools, which can now leverage your service foundation. Currently, some of the available tools for BPM either bundle or offer easy integration with SOA tools, thus making it easier to combine BPM adoption with SOA.

Comparison

Figure 2 shows a comparison of all four options, taking into account factors such as the up-front investment required versus system flexibility. As you can see, the reusable business services approach provides the least flexibility among the four options, but it also requires the least up-front investment. It's unlikely that an organization would stop at this destination, but by defining it as the first stop, we can show the business benefits of investing in and adopting SOA. On the other extreme, it's unlikely that most organizations would define the foundation for BPM as a destination, largely because of the substantial up-front investment needed.

Knowing where you want to go before you start your journey is definitely important when an organization decides to move toward adopting SOA. Although it can bring much-needed flexibility to an organization's information systems, adopting SOA with only a vaguely defined goal of achieving system flexibility could be an invitation to failure. Instead, choose a clearly defined destination to help your organization build a strong business case and clearly define the success criteria. IT

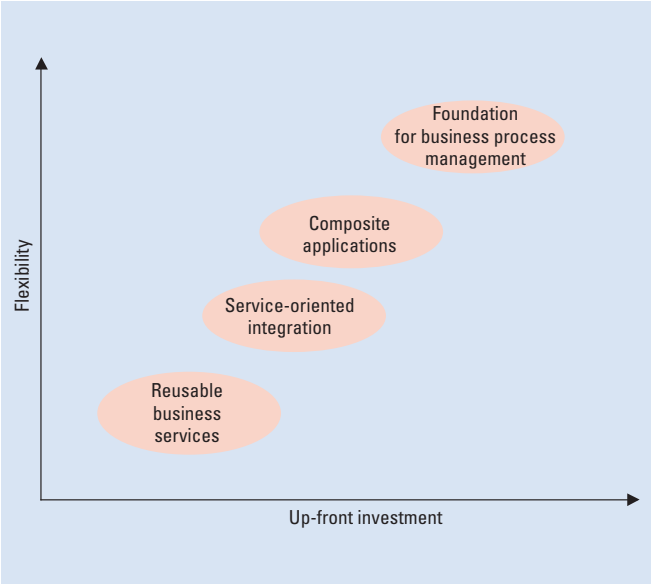

Figure 2. Comparison of options for your SOA adoption journey.

References

1. J. Akella, K. Kanakamedala, and R.P. Roberts, "What's on CIO Agendas in 2007: A McKinsey Survey," *The McKinsey Quarterly*, Jan. 2007; www.mckinseyquarterly.com.
2. R. Heffner et al., "Planned SOA Usage Grows Faster than Actual SOA Usage," *Forrester Research*, 28 Feb. 2007.
3. L. Wilkes, "Composite Applications," *CBDi J.*, no. 9, 2004; www.cbdiforum.com/secure/interact/2004-09/.
4. M.J. Melenovsky, "Business Process Management as a Discipline," report, Gartner Research, 1 Aug. 2006.

Shrikant Mulik *is the senior technology consultant at L&T Infotech, Mumbai. Contact him at shrikant@iitbombay.org.*

Sushil Ajgaonkar *is the SOA practice lead at L&T Infotech. Contact him at sushil.ajgaonkar@lntinfotech.com.*

Kavindra Sharma *is the chief of technology at L&T Infotech. Contact him at kavi@lntinfotech.com.*