e-Newsletter Exclusive

Print

SOA—Reality or Myth?

August 2007 | by Jaqui Lynch

Over the past 30 years I've seen many technologies come and go and can't even keep track of the number of acronyms. Every time a new acronym comes out someone declares it to be a paradigm shift and so it begins. So is service-oriented architecture (SOA) yet another short-lived hot-button acronym found in an airline magazine or does it truly offer value in the IT enterprise? I think SOA is useful and here to stay, and hopefully you'll agree after reading this article.

What is SOA?

SOA is a framework or architecture that allows IT assets to be organized into discrete components (called services) that can then be made available to applications to use. At the basic level, SOA is a collection of services that communicate with each other. This can be as basic as two services passing data, or it can be more complex and involve more coordination.

A service is any repeatable business task. In IT terms this means a well-defined, self-contained function or entity that's independent. The features that allow the integration and linkage of these services are enabled by middleware software that's commonly referred to as an enterprise service bus (ESB). The ESB isn't necessarily Web services-based, but it's standards-based and needs to support many middleware or transports. The ESB's purpose is to provide the necessary messaging technologies and routing such that disparate systems or applications can talk to each other.

What SOA Brings to the Table

Where SOA differs from previous technologies is that it's more an architecture that enables the integration and reuse of services through applications based on many technologies including Web services. The key requirement is the services must be independent, and that they're written to predefined standard interfaces that make it easy for the services to communicate with each other. The services should also communicate in a manner that's independent from the underlying hardware, software or programming language. The interface definition hides most of this platform- and language-specific information such that an application should be able to call the service using the interface with no regard to whether the service is Java, .NET, C++ or another and with no regard to what hardware or operating system the service is running on. This is critical to the enablement of reuse of services.

The intent of SOA is to provide a framework that combines tools and technologies in such a way that it's possible to implement new services and applications. This allows IT to be more responsive to changing business needs while retaining control over costs. SOA is showing great promise in this area.

Flexibility of SOA

Several characteristics are common to the general IT approach to solving business problems, including having a focus on technology or function, being reactive, working in silos in a very ad-hoc manner and being heavily IT-focused. IT constantly struggles with how to adapt quickly to changing business needs while still maintaining system stability and security. Over the years many new regulations have been introduced (e.g., HIPAA and Sarbanes-Oxley Act) that have exacerbated this problem even further. IT now has to deal with Web-based systems that go across international borders and that run 24-7. Companies deal with a world where competition is fast and furious and they must be able to bring up a new application quickly in order to stay competitive. Clearly the current IT focus on technology and functions isn't conducive to rapid reliable change.

SOA, on the other hand, is more focused on the business and the coordination of resources. It's built to be flexible, designed to be integrated and its services are repeatable. The orientation is service-based, not operations-based, making it a prime contender in the architecture battle to renovate IT focus. This shift in focus to a more service-oriented approach allows SOA to provide a more flexible environment because everything is coded to APIs or well-known interfaces. This provides maximum code reuse and a simpler integrated environment.

A Useful Tool

As with all frameworks and architectures, a cultural change is necessary in order to have a successful migration into the new environment. The key to success here is that all of the processes and services that enable SOA need to be well documented to provide the reuse necessary to truly take advantage of SOA. Management buy-in is also necessary at the highest levels to map this out rather than just waking up one day and saying, "today we will start implementing SOA."

SOA can reenergize a business and an IT department if proper planning is put in place ahead of time, if both the business and IT understand that this is a priority within the company, and if it's supported from the highest levels down. Education is another key component of reorganizing into an SOA world. As we all know, technology is rarely the reason that new technologies or architectures fail--in general it's due to a lack of organizational commitment, unreasonable timeframes or expectations. SOA is no panacea--it's another architecture where proper planning and education can make a huge difference to the business' success.

While SOA shows great potential in helping speed up application development and integration and can be used to provide a great foundation for collaboration, it's important to remember that SOA is only part of the answer and needs to be viewed as part of an overall framework for the business. SOA isn't going away any time soon and should be reviewed for its potential as one of the tools in the IT and business toolkits.