

SOA: A Closer Look

by *aimpublications LLC*
(www.aimpublications.com)

Implemented strategically, SOA really can deliver business transformation. Here's a look at what's involved and why it's worthwhile for the competitive-minded.

Two out of three enterprises want to grow faster than the market in 2007, notes a recent survey by Gartner Inc., a Stamford, Conn.-based research and consulting firm. For these companies, competitiveness has, by definition, become a key goal.

If current market research is to be believed, there's little question that insurance carriers regard service-oriented architecture (SOA) as one of the technologies that will help them compete. The numbers vary, but several studies show an overwhelming majority of carriers now embrace SOA.

And for good reason: SOA enables software developers to mask the technical complexity inherent in business operations. Implementing SOA can reduce carriers' operational and application integration costs while improving use of assets, the ability to adapt existing systems and the delivery of services to agents. SOA can boost both IT and business agility and enable better alignment of IT and business objectives.

But how a carrier embraces SOA makes all the difference in whether SOA expectations are actually achieved.

SOA strategy vs. Web services tactics

That's because SOA is far more than the latest, greatest IT tactic. An entire information technology strategy, service-oriented architecture is based on the concept of creating and delivering not applications but services. Built using a

How a carrier embraces SOA makes all the difference in whether SOA expectations are actually achieved.

methodology called service-oriented programming, these services are comprised of meaningful chunks of software—components—that can easily be linked with other chunks for use in multiple business operations.

To be functional, the code bundled into a component needs an interface—a wrapper—describing what it does and the ways it can be connected to other components. How this is accomplished involves the tactics of SOA.

Over the last several years these tactics have included customized programming links and integration software from a variety of vendors. More recently, much of this connective tissue has become standardized in the form of Web services, a set of communication and linking mechanisms based on the World Wide Web.

In a perfect world, investment in SOA strategy—that is, creating a new enterprise technology architecture, a centralized development methodology, a rationally cleaned-up data set, a registry/repository to track available services, and appropriate governance practices—would always parallel efforts in mastering SOA tactics. But too often this is not the case.

Thus insurance carriers tend to deploy Web services to enable access to back-end legacy systems and support Web-based applications without equivalent concern for an SOA strategy.

Committing to transformation: It's a business decision

When carriers limit themselves to Web services tactics, however, they also limit their ability to achieve the benefits of SOA, which can consolidate policy and claims systems, lower operating costs, and simplify maintenance. To achieve these more competitive results, carriers must commit to an SOA strategy. Of course, such a commitment transforms the way an organization functions.

The impacts of SOA strategy on the wider IT environment cannot be overestimated. They include:

“The impacts of SOA strategy on the wider IT environment cannot be overestimated.”

- Designing an architectural blueprint appropriate to the entire enterprise
- Adapting the existing infrastructure to support a layered services-oriented approach
- Preparing for changes to performance demands,
- Reshaping IT culture to deliver services—not applications—that meet business requirements with minimal operational and technological barriers
- Addressing sometimes formidable IT problems, including low-quality IT processes, ineffective project management, inflexibly hard-coded business rules and data that needs cleaning or conversion or both.

In addition to these architectural and cultural challenges, key SOA strategy issues involve governance, service repositories and registries and data integration and data architecture.

Carriers go through all this for business rather than technical reasons. Application integration, a common use of SOA, takes less work when it's done using traditional methods. One analyst estimates the extra work involved in service-oriented development ranges from 30 percent to 100 percent in the design phase.

But SOA work pays off across the entire enterprise, not just for a single business unit or application, nor just once: When service-creation efforts get reused, they make a foundational contribution to the enterprise's continuing ability to do business.

Governance and SOA

Such reuse of services across a company is not possible without:

- Preventing creation of multiple incompatible versions of the same service and/or incompatible linkages between services
- Having a central place—a repository or warehouse—where all services reside; services must be well-documented so developers understand what each service does, know where to find them, can see where they're being put to work and are aware of the rules governing their use.

SOA work pays off across the entire enterprise, not just for a single business unit or application.

Hence the ability to track services is essential in SOA. SOA governance begins with a centralized software development methodology and a services repository and registry. Solid SOA governance policies permit insurance carriers to provide automated change capabilities, self-administration of policies and agent access to information.

Compared to carriers' traditional IT environments—which can require up to a year to introduce new products and almost that long to generate rate changes—well-governed SOA delivers speed-to-market that enables carriers to respond faster to market changes while cutting operating costs.

Repositories and registries: Both are needed

Good SOA governance often requires both SOA repositories and registries, which perform distinctly different and complementary functions.

An SOA repository stores all the information that is actually useful to develop, deploy and run services, including service artifacts—such as WSDL, XSLT (eXtensible Stylesheet Language Transformation) and other XML schema—and service dependencies. Among the data elements an SOA repository can store are policies, processes, transformations, documentation, XML assets, source code, requirements, architectural documents, reports and run-time data.

Often the services stored in a repository are referenced by a service registry, which is a catalogue or index that contains descriptions of services based on UDDI (Universal Description, Discovery and Integration), an XML-based OASIS standard for describing, publishing and finding Web services. Typically, a registry is itself a component containing references to elements stored elsewhere (in a repository) and does not actually store what it references.

An SOA registry provides a central point of control for run-time discovery, usage and measurement of deployed services and helps in implementing the governance model that supports real reuse.

“In many traditional IT environments, the state of data is often a chaotic mess.”

The need for a repository becomes clear as a company's SOA efforts move past prototypes and pilots to involve the entire enterprise and its many functions and units. At this point, a repository is necessary for coordinating not just basic service usage but also other sorts of metadata, including information about business requirements. The repository becomes a critically important central clearing house that tracks interactions between all parts of the organization that influence a service.

Why data architecture matters

All this effort is useless, though, if the data is dirty.

In many traditional IT environments, the state of data is often a chaotic mess: assorted siloed systems maintain the same information in incompatible formats using incompatible semantics and individual records often harbor plenty of duplications and inaccuracies. Even for organizations with no intention of adopting an SOA strategy, problems with data loom large and painful, and data-generating technologies like email and RFID will continue to exacerbate the situation.

In larger enterprises where custom-developed interfaces have been used to integrate disparate applications, a complex legacy has been spawned that defies easy integration. Meanwhile, both competitive and compliance requirements have been pushing many firms toward more sophisticated business intelligence capabilities—but here again data integration is hindered by poor data quality and lack of data semantics standards.

For those implementing SOA to stay competitive, clean, decent-quality data is a necessary condition for success. Furthermore, multiple instances of data and databases must be reconciled and rationalized. The fact is that service-oriented architecture needs to be supported by a data architecture based on:

- An unequivocal understanding of the location, structure and quality of corporate data

Success depends on showing executive management that an SOA strategy—not just Web services tactics—is a wise investment.

- A standard set of tools—ranging from ETL (extraction, transformation and loading) tools and data federation to database gateways, replication and file transfer—for implementing data integration tasks.

Presenting the business case

Because SOA is a transformational force impacting the business rather than just IT, success depends on showing executive management that an SOA strategy—not just Web services tactics—is a wise investment.

This is best accomplished by presenting a business case for an SOA strategy. According to Gartner's recently released EXP CIO report *Creating Enterprise Leverage: The 2007 CIO Agenda*, the CEO's demand for better ways to compete, which involves leveraging both short-term performance and long-term competitiveness, can be met by IT leaders in four ways:

- Using technology that reduces enterprise cost structures, improves operational scale or raises process performance through automation, integration and standardization
- With agility, so the speed, scale, cost and risk of change is handled using the disciplines of change management
- Using information to achieve the business insight and understanding required to act in a changing environment
- With innovation, which extends current capabilities as well as implementing new ones to successfully introduce new ideas and products to the market.

Stages of SOA

Some technology watchers describe a three-step SOA process:

- 1 Software abstraction:** Increasingly capabilities—such as data management, identity management—are no longer coded separately into each application and instead become part of an overarching software infrastructure.
- 2 Infrastructure consolidation:** The number of siloed applications and solutions is reduced.
- 3 Functional componentization:** Software is modularized into linkable components that mask technical arcanities effectively enough to allow end-users to construct their own business-function composites.

The SOA Advantage

Web services tactics provides some limited, short-term leverage of technology, agility, information and innovation. But SOA strategy does much more:

Reusing code. The chunks of software that become services can be hard to get right. Achieving the best size, or granularity, is as much art as science, but it's worth the effort because the services can be reused the next time that function is needed. Even a single reuse can pay off, despite more initial design work, because these initial design costs are eliminated thereafter.

Experience indicates the technology and innovation leverage achieved with such reusability is likelier if the code is developed in the context of a broad SOA strategy—one that includes elements that boost the odds of reusability. This includes governance mechanisms—notably a single enterprise-wide development methodology, centralized development teams and a service repository and registry. It also includes uniform development methodologies, a centralized enterprise architecture planning staff and business analysts with an enterprise-wide reach.

Boosting productivity. When services are reused, development cycles tighten and integration becomes a lot cheaper (at least 30 percent cheaper, according to estimates by Gartner).

Becoming more agile. Services make systems easier to modify even when reuse isn't an option and this, in turn, makes the business more agile.

Aligning IT with the business. SOA provides a high-level view of all of an enterprise's business processes and flows, so non-technical people can grasp how their businesses are constructed in terms of technology. Such understanding engenders their support.

Longer term, once SOA fully service-enables major business processes, non-technical people will be able to

**When services are reused,
development cycles tighten
and integration becomes a
lot cheaper.**

engage in innovative use of technology and information to build their own composites of available services, creating new processes and new levels of business agility. ■

About aimpublications

aimpublications LLC (www.aimpublications.com) provides commentary and insight about the complex business and technical issues shaping information technology's transformational impact on business value and competitiveness in key industry sectors, including financial services.