SYS-CON
THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES

SOA WORLD MAGAZINE

www.SOA.SYS-CON.com
APRIL 2008 / VOLUME: 8 ISSUE 4

Long-Tail SOA and the Mythology of Reuse

$6.99US $7.99CAN
04>
0 71486 03420 9

Sound Architecture Requires Proper Planning

WEB AGE SOLUTIONS - YOUR TRAINING PARTNER FOR SOA

In all phases of SOA migration, Web Age Solutions provides training and customized services from awareness to implementation. We support vendor specific or generic SOA training tailored to your organization's needs.

Custom training for complementary SOA technologies

XML • WEB SERVICES • WSDL • SOAP • WEBSPHERE • WEBLOGIC • JBOSS • J2EE • SPRING/HIBERNATE/STRUTS • WID/WBI/WMB

Custom training plans for virtually every job role

BUSINESS ANALYST • ADMINISTRATOR • DEVELOPER • ARCHITECT • QA/TESTER • MANAGER • EXECUTIVE • SECURITY

Consulting services for all phases of SOA migration

IBM Premier Business Partner

Add Web Age Solutions to your plan & stay ahead of the competition

www.webagesolutions.com - 877.517.6540 - soa@webagesolutions.com

WEB AGE solutions inc.

SOA WORLD MAGAZINE
www.SOA.SYS-CON.com

EDITORIAL OFFICES
SYS-CON MEDIA
577 CHESTNUT RIDGE ROAD, WOODCLIFF LAKE, NJ 07677
TELEPHONE: 201 802-3000 FAX: 201 782-9637
SOA World Magazine Digital Edition (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending
Woodcliff Lake, NJ 07677 and additional mailing offices
POSTMASTER: Send address changes to:
SOA World Magazine, SYS-CON Publications, Inc.
577 Chestnut Ridge Road, Woodcliff Lake, NJ 07677

Defining Terms

WRITTEN BY **SEAN RHODY**

It seems like not a day goes by lately in which some new story of malfeasance in office doesn't come out – whether it's lying under oath, using the services of a call girl, or spying on other officials in the government in order to further a personal agenda. Clearly, our elected officials don't have a clue about the meaning of governance.

Given that, it's not really surprising that there seems to be some confusion in the SOA market place about what we mean when we say "SOA Governance." To some vendors, governance means a product. When I say governance, I mean a process. While I think it's simple to explain what we mean, I don't think the confusion about terms is beneficial to the industry.

When an enterprise IT organization adopts SOA, it begins a journey of transformation within the infrastructure as well as within the fundamental business processes of the corporation. This has a number of results that are fundamentally disruptive to the steady state of the organization (change always is). The small pilots that are created as the first steps toward SOA have very little impact on the organization. They create a few services, help define product selection and everyone goes on the way they have been working. But after a certain point, maybe 20 or 30 services or so, a variety of issues crop up.

One is ownership of a service. Ownership implies control, but it also implies cost and funding. When a department funds a small server to do LDAP, and it becomes overwhelmed because 16 other departments now want to use the same shared service (identity management), finger pointing can quickly occur. Rarely do funding models and growth predictions get rolled into the first services, and yet many of them are the support services that are necessary for every project.

Another issue is that of version control and consolidation. When a company decides that there will be one defining service for some particular purpose, it needs to be able to consolidate various versions of the truth that may already exist. This poses several problems. The most immediate problem is getting multiple groups to agree on what the single version of the truth is. Sometimes this results in huge battles or a large number of "optional" parameters. Groups that have defined basic communications differently (like marketing and manufacturing having their own views of what is a customer) are not likely to easily change their perspectives, but it's usually in the organization's best interests not to have multiple versions of the truth. In the case of a tie, who gets to be the tie breaker?

These and other questions point to the organizational and managerial aspects of service-oriented architecture. While there are multiple ways to organize around these basic issues, there is no getting around the fact that there needs to be some organization that makes decisions and helps determine the transformation of the IT landscape from silo applications to shared (or at least shareable) services.

That's really what I mean when I say governance – organization and processes for controlling the use of services.

Now what some vendors seem to call governance is something I think of as SOA Management – namely the observation of, enforcement of, and reporting of the quality of service or service level agreements. SLAs and QoS are both important facets of the IT world – but I rarely regard them as governance. In most cases you can't truly decide on the SLA or QoS, rather it's derived from best performance characteristics or limiting factors like network latency. So rather than governance, I usually think of this as Management or Reporting.

Most of the time, confusion over terms isn't so bad – but you can see that calling one thing something else can lead to bad behavior and mistaken purchase orders.

Time to run; I think I've just been nominated for governor. ■

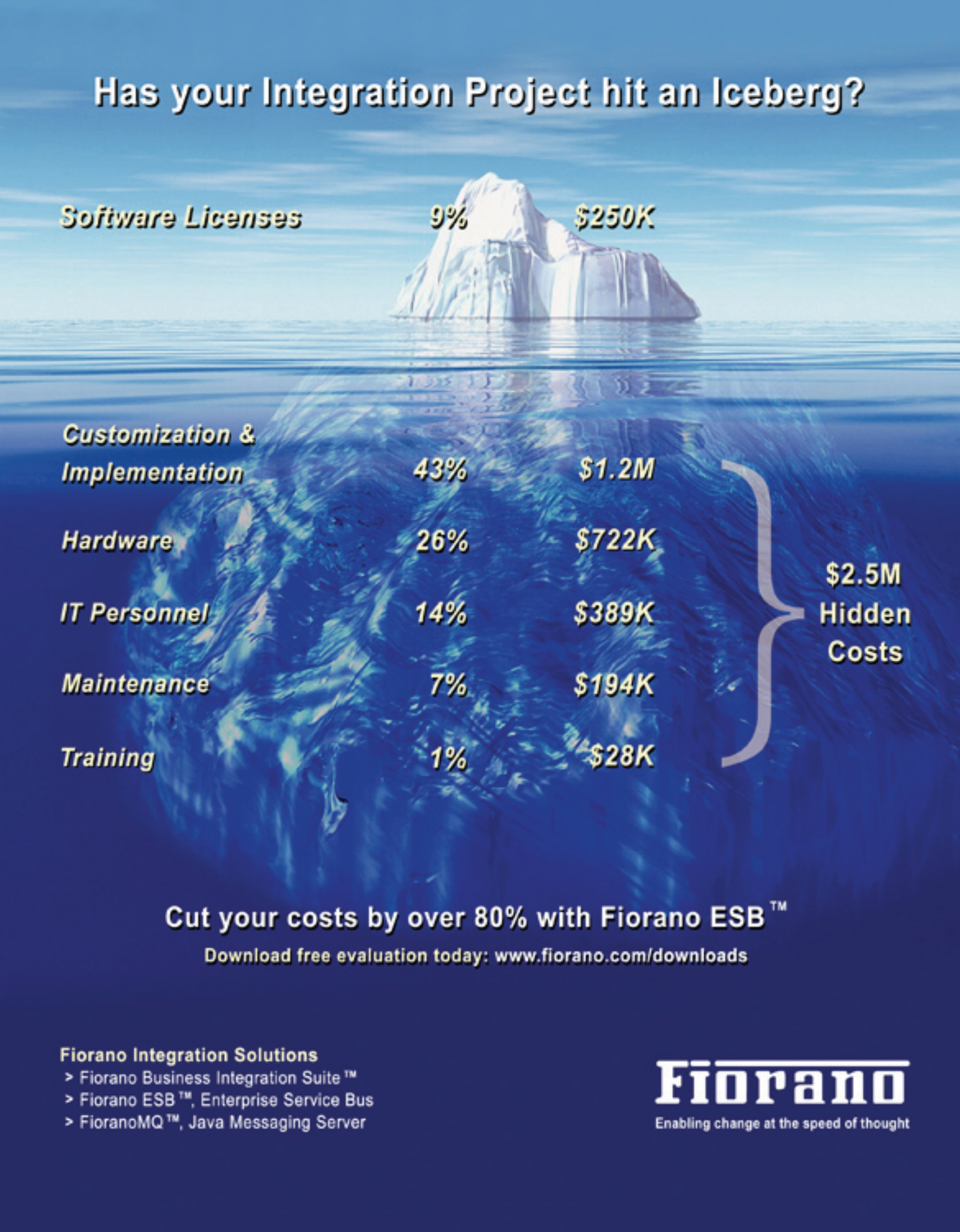
About the Author

Sean Rhody is the editor-in-chief of **SOA World Magazine**. *He is a respected industry expert and a consultant with a leading consulting services company.* *sean@sys-con.com*

INSIDE SOAWORLD MAGAZINE

DOWNLOAD PRINT VERSION

Has your Integration Project hit an Iceberg?



Cut your costs by over 80% with Fiorano ESB™

Download free evaluation today: www.fiorano.com/downloads

Fiorano Integration Solutions

> Fiorano Business Integration Suite™
> Fiorano ESB™, Enterprise Service Bus
> FioranoMQ™, Java Messaging Server

Fiorano

Enabling change at the speed of thought

SOA WORLD MAGAZINE
www.SOA.SYS-CON.com

Testing Process Orchestrations Based on the BPEL Standard

BPEL testing got a bad rap

BY MIKE PELLEGRINI

Composite applications are made up of discreet services that have been tried and proven reliable, but building an orchestration that incorporates services that come from several sources, some of them outside of the company, could introduce testing hazards beyond just bad output. For example, let's say that your business has a process that includes activities to run a credit check with an external credit agency or to schedule a package delivery with an external shipping service. When doing either of those tasks, we are introducing into the process two elements that can't be easily tested live, simply because we don't own the services internally.

Many processes also include delays or timeouts that are built into the process – this can further complicate matters, so testing each run-through of the application could take days. Besides these external factors, there are also internal processes that are in heavy use for other applications that are subject to changes and downtime. What you're left with is few, if any, services that are easy to test.

Meanwhile you have out-of-band events, correlation, and other factors that present challenges to testing composite applications. The need to pay extra attention to these risk factors introduced by building applications from loosely coupled services doesn't diminish the value of building standards-based composite applications, but a methodical and visual system to address these factors can extend the value even further.

In fact, unit testing for BPEL-based applications doesn't have to be harder than testing any other kind of program. If you take a few reasonable steps you'll have the information you need to confidently revise and deploy your standards-based composite application using a test-first methodology.

A clear first step is to take the process offline. For example, if your application calls for a credit check, it may be impractical to send 10, 100, or 1,000 requests to the credit agency while you're testing. This can be done by collecting the actual responses created by live use of the service or generating sample data on your own to represent both the expected responses and all likely alternate responses, like responses that indicate a failure or responses that contain unexpected data. By using sample data in place of actually calling live services you can safely run process tests and be guaranteed of the expected outcome.

At this point you're ready to run your suite of process scenario tests. The best way to understand the flow of activity and how problems develop is via a visual representation of where scenarios have passed, failed, or reached an unknown state. For example, displaying a report showing that out of 100 scenarios, 67 passed, 30 failed, and three reached an unknown state. This will help you find the problem areas and avoid putting a band-aid on certain symptoms when the real problem is somewhere else.

Finally, you need to annotate the results to pinpoint the problems further. So, of the 30 failures detected, you'd want to be able to visually depict where exactly in the process orchestration the failure occurred. From there, you're armed to fix the issues and retest the application until all of the results are as you expect and any failures are dealt with to your satisfaction.

Many architects believe that testing composite applications is extremely difficult, or even impossible. However, if the correct steps are taken, that stigma can be erased, and testing composite applications can be just as easy as testing other any other program, and testing is crucial to gaining maximum benefit from your SOA and composite applications. ■

About the Author

Mike Pellegrini is a principal architect in the Office of the CTO at Active Endpoints. For the past 15 years, he has worked for both leading software vendors and IT organizations in moving technology solutions from early adoption to mainstream acceptance.

Are your SOA Projects in Line?

It seems there are SOA projects everywhere these days, helping businesses accelerate application development and deployment. But managing SOA performance can be difficult, since traditional tools cannot relate end user experiences to web services. B-hive is changing that with award-winning solutions that align IT Service performance policies with actual business goals.

Visit www.bhive.net today to find out more and download the free test drive.

b-hive

Tel: (650) 358-1300

CITRIX ready

vmware certified virtual appliance

Long-Tail SOA and the Mythology of Reuse

In praise of the niche market, the other 80%

BY MARC RIX

Not all services are created equal. It would be great if implementing SOA were simply a matter of applying a standard design pattern to all services. Once IT had identified and codified an optimal design standard, services could be stamped out in assembly-line fashion until the IT landscape had been transformed. Unfortunately, we don't live in a cookie-cutter service Utopia.

Different Strokes for Different Folks

Services exist in all shapes, sizes, colors, and flavors. What's more, they're not all conducive to the same design approaches. Services are the modern digital incarnation of useful business functions: Get Employee, Create Purchase Order, Calculate Credit Score, Find Recommended Products. Any amount of work done by a system on behalf of another system (or user) qualifies as a business function. IT's raison d'etre is to provide these functions to the enterprise, and the types of functions offered are as varied as the customers who use them, the complexity of the tasks done, and how each one relates to other functions. Some functions are big, some are small, some are used throughout the enterprise, and some are used only in obscure business niches. No two services are exactly alike and each service delivers a unique amount of value to the enterprise. To understand a service's value we need to understand how it's used.

Like books, songs, movies, news, and coffee, different customers prefer different IT services. There are popular "mainstream" services

that appeal to large groups and unpopular "niche" services that appeal to only a few people. What's more, as the world is discovering that there's tremendous value in offering niche products to obscure "Long-Tail" consumer markets, IT has a similar opportunity in providing big value to the enterprise by offering Long Tail services. This flies in the face of traditional IT strategy (and popular SOA strategy for that matter), but it's an essential ingredient in the recipe for agility.

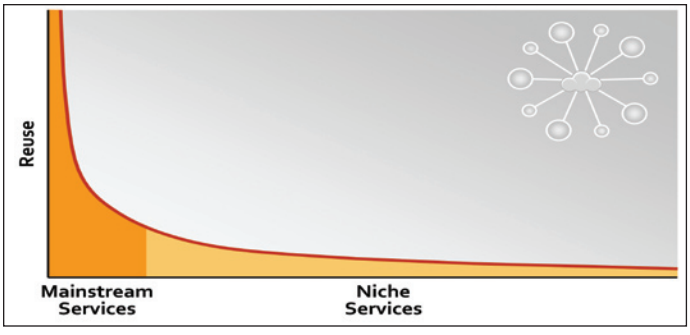
Creatures of Habit

The big events get all the press. The A-List actors make all the money. The biggest hits get all the radio play. The coolest gadgets get all the sales. We latch onto mainstream products because they make a big impact. These phenomena follow Long Tail distributions, where 20% of the causes lead to 80% of the effects. The Long Tail is hard at work in our IT departments too, governing the popularity of our technology functions. (A technology function is any unit of software that could be exposed as a service through SOA.) Each function carries a certain level of demand. Plotting the demand profiles of all IT functions and ranking them from most used to least used would produce a Long Tail graph similar to Figure 1, where the most used 20% account for roughly 80% of the total market for IT functions. These are IT's star performers and are being tracked down and converted into services by SOA practitioners everywhere. The other 80% lurk in the shadows of IT's portfolio and are generally overlooked as candidates for SOA. When it comes to deciding which business needs to focus on for SOA, we instinctively draw a clear line in the sand. We grab onto the most recognizable 20% and forget the rest. Why such divided interest?

We see opportunity in catering to the needs of the masses. We see risk in catering to the needs of the few. As SOA practitioners, our goal is to maximize the value of IT by instituting sweeping architectural change. Our job is to unwind decades' worth of accidental architecture and convert IT from a necessary liability into a strategic asset, and do it all before our competitors do. No pressure. Naturally, when the pressure is on we focus on what we know, what we are most comfortable with. We are most comfortable building for maximum reuse.

The Mythology of Reuse

IT historically caters to the business masses. We own the ERP systems, the intranets, the corporate Web sites, the enterprise portals, the business intelligence platforms, the data warehouses, the e-mail infrastructure, the LDAP forests, the B2B feeds. We build million-dollar solutions and roll them out to hundreds, even thousands, of users. We go big because we have to. IT is the technology utility provider for the enterprise. Because our IT departments have limited resources, we must be selective about the projects we undertake. Traditionally, we select the ones with the biggest impact, the broadest reach, the highest reuse to satisfy as large a contingent of customers as possible. We can't afford to occupy ourselves with providing obscure products in low demand any more than the corner convenience store can afford to stock every brand of toothpaste, cereal, and beer. IT has been in the business of providing mainstream products to a mainstream audience for decades. When it comes to SOA, we instinctively follow the same strategy. We strive for SOA solutions that offer the greatest value. Unfortunately, the way we typically valuate SOA solutions is flawed.



GRAPHIC BY BRIAN TEIFER

Figure 1: The Long Tail of IT

The most popular method of determining SOA value involves examining reuse. "Do that which maximizes service reuse" seems to be the mantra of the SOA industry today. Indeed, in my own experience, reusability has been the primary, if not exclusive, yardstick with which SOA success is measured. SOA success seems to be synonymous with high service reuse. This philosophy implies that reusable services are more valuable to the business than non-reusable ones, and bases ROI on the number of consumers a service has. More consumers means higher reuse means greater value. In reality, the link between service reuse and business value is a myth.

We tend to believe that services will be reused if designed well, but we have little direct control over how reusable a service is. Sure, we can take care to design our services in ways that let them be leveraged by more than one consumer, but this just ensures that the service is capable of being reused, not that it will be reused. Reuse is really a function of demand. Naturally, the higher the demand the higher a service's reuse potential. The point is that a service is not reusable because it was designed a certain way. Rather, a service is reusable because it exposes a business function that is already in high demand.

We also have a tendency to believe that reuse leads to business alignment and agility. Service-enabling IT's most widely used capabilities should bring IT and business closer together and change-proof most of the enterprise, right? Actually, mainstream services benefit IT more than they benefit the business. IT is likely to respond more quickly in times of change, but the benefits of that agility seldom extend out into the lines of business. Suppose, for example, that an IT shop anticipates replacing an antiquated ERP system. It prepares by abstracting its widely used reporting functions through service interfaces and retooling existing business intelligence systems to use those services. Then at some point in the future executive management sends down the order to swap in a new system. If everything goes well, users of the BI tools (and the tools themselves) will have no knowledge of the swap. The transition will be seamless and transparent to consumers of ERP reporting functions. The success would showcase IT's agility and alignment with corporate directives, but wouldn't benefit lines of business directly. There's a difference between providing services that improve business practices and providing services that prevent disruption to business practices. Service-enabling mainstream functions protect lines of business from change (an IT virtue) but don't make lines of business more competitive (a business virtue).

Mainstream service opportunities are also relatively rare. They are extremely important, to be sure, but they are far outnumbered by niche opportunities. To provide only highly reusable services is to turn a cold shoulder to 80% of the company's business needs. Who will be there when those needs change? What's more, niche services benefit business users directly since it's usually line-of-business personnel who request them. Niche services solve very focused business needs and, by definition, aren't very reusable;

however, niche needs exist en masse and come directly from the trenches – and that's where business battles are won and lost.

As long as the goal of SOA is to improve business agility, the needs of our companies' niche markets shouldn't be ignored. Focusing on popular solutions and maximum service reuse is a natural IT response to the challenge of building enterprise agility. But an SOA built on the premise of maximizing service reuse focuses too much energy on internal, corporate needs. To deliver on the true promise of SOA, our strategies must also push specialized services out to the very edges of our businesses where they can be leveraged when needed. Services do not have to be reusable to be valuable. They just need to be available. This is Long Tail SOA.

Mainstream versus Long Tail SOA

Reuse is only one factor in the SOA value equation. Now that the myths about service reuse have been revealed we can open our minds to other considerations — namely demand for services and the nature of change. These forces influence SOA implementation and can act for or against SOA goals depending on the strategy. What may weaken a mainstream approach may strengthen a Long Tail approach, and vice versa. Remembering that mainstream SOA focuses on providing popular services for a few large business "markets" and Long Tail SOA on providing specialized services for many small markets; let's examine how each approach is affected by service demand and change.

As said, different business customers need different IT services. When a service is needed by more than one customer reuse happens. Reuse is a measure of the demand for an individual service and is generally good. But it's also important to measure the demand for entire collections of services, or service "markets." The aggregate demand for mainstream services is huge, encompassing roughly 80% of all service consumers. By contrast, the aggregate demand for niche services is relatively small at 20%. At first blush this might suggest that building niche services makes poor business sense. But due to the Long Tail effect, four niche services are in demand for every one mainstream service. In other words, mainstream SOA leaves four markets underserved for every one it satisfies. These four markets may be small, but they still represent a missed opportunity for IT. How much missed opportunity is a function of the protection those services provide during times of change.

The value of SOA is revealed during turbulent times. This is where the SOA rubber meets the road. Like the distribution of service demand, the distribution of change across the enterprise isn't uniform. Some areas of business are more susceptible to change than others, and some periods of change are more disruptive than others. SOA strategies should be in tune with these differences, otherwise they risk committing the cardinal sin of IT — being unprepared to facilitate fluctuating business needs.

IT must be prepared to deal with the magnitude of change. We have a good understanding of this already by virtue of our mainstream IT mentality. We know first hand that when change affects core enterprise systems — through government regulation, acquisitions, legacy system retirement, and so on — the shockwave can rip through the entire company. Being unprepared for major infrastructural change leaves a big crater and can be the death knell for a company. Mainstream SOA protects against this kind of threat. By contrast, a change to a niche service leaves a small crater and only affects a small pocket of the business. Companies usually heal quickly from isolated niche threats.

IT must also be prepared to deal with the frequency of change, which is often overlooked. Every time change, big or small, is allowed to affect the business, the business's defenses are weakened. Mainstream SOA guards against the biggest threats but lets the smaller ones through, which would be fine if there weren't so many small threats to worry about. In reality, though, big change is rare and small change is constant. Mainstream services change infrequently because they are mainstream services. They are owned by IT, they are the lifeblood of the business, and carry core functions and data to all facets of the enterprise. They uphold the fundamental policies, business practices, and values of the company throughout the business ecosystem. They must remain as stable as possible because changing them risks disrupting the business on a global scale. As such, change in mainstream IT typically occurs on time scales measured in fiscal years or quarters.

Change in niche IT, on the other hand, is measured in days. Niche services are owned by the business and change with external market conditions. They deliver data and functions that provide relevance and differentiation to business units in local markets. People at the edge of business — sales reps, regional managers, product managers, R&D engineers — need more than mainstream one-size-fits-all services because their customers need more than generic one-size-fits-all products. These areas of the business make demands of IT that are a direct reflection of their customers' needs, which are always changing. For IT not to respond to small changes in niche service markets is to allow the systematic erosion of LOB revenues. Real IT-business alignment and agility comes from Long Tail SOA because when change happens, it's most likely to be felt at the front lines rather than the back office.

So which is the best SOA strategy? In the end, is it better to hedge against infrequent big change or frequent small change? Is it better to provide services to a few large audiences or many small ones? Is it better to strive for reuse or agility? Like most situations, posturing at one extreme or the other is seldom optimal. The best strategy is the one that best addresses the strategic goals of the company, but probably involves some amount of blending of mainstream SOA and Long Tail SOA.

Hybrid SOA

Mainstream SOA maximizes service reuse and prepares IT for the most crippling effects of change, but it's complex, expensive, and takes years to show significant returns on investment. It "feels" more natural but aligns IT more with the corporate hub than with the business at large. Long Tail SOA caters directly to line-of-business needs, dampens the effects of everyday change, is inexpensive, and yields a quick ROI, but it's less intuitive and addresses only one-fifth of the total demand for IT services. Either approach by itself won't lead to SOA nirvana, but combining them through intelligent service design can.

Make Long Tail SOA an extension of mainstream SOA. Most niche services are refinements or customizations of mainstream services. A mainstream "Get Property For Sale" service could evolve into "Get Houses By Zip Code In Price Range" as it travels down the Long Tail. In this example, the original service became less valuable (generic and reusable) at the mainstream level but more valuable (customer-oriented) at the niche level. Although the second service is a different "product" than the first with a different market, it doesn't have to be implemented separately. In fact, it should borrow as much as it can from the generic service and just add the unique parts. For example, it could call the generic service and refine the

results based on its specialty. In other words, Long Tail services can be derived from mainstream services simply through effective service composition. Easily stated, but this requires a bit of discipline.

Hybrid SOA requires a strong Mainstream SOA foundation. Because Long Tail services are dependent on mainstream services, mainstream services need to be stable. If a mainstream service changes it could cause entire families of related Long Tail services to change. Solid design, high availability, and loose coupling are essential at the foundation service level.

All services must be discoverable. In Hybrid SOA, single-service solutions are rare. Instead, IT solutions are value chains of services of varying granularity and complexity strung together to meet specific business needs. Developers (or mashers!) of Long Tail services need to know where the raw materials are to draw from. Otherwise, the services either can't be built or are built by duplicating existing services.

Finally, strong standards and governance are required to enable hybrid SOA. This point may seem obvious, but the importance of standards and governance should be stressed since hybrid SOA brings two distinctly different service economies together. Standards and governance practices that are effective in either mainstream SOA or Long Tail SOA may not be relevant when the two are combined. For instance, mainstream SOA may operate well without strict interoperability standards, especially if the major IT systems are more or less homogenous, such as in predominantly Microsoft or Oracle shops. But interoperability standards would become important when introducing Long Tail SOA if technology used by lines of business were more of a mixed bag. IT may also not have the resources or bandwidth to supply all Long Tail services, in which case niche service development (or mashing!) would need to be decentralized. Standards and governance would need to be rigid enough to maintain quality of output yet flexible enough to accommodate a diversity of composition tools and methodologies.

The Bottom Line

Like many products, the demand for IT functions follows a power law distribution, where there are a few extremely popular mainstream functions and many unpopular niche functions. IT tends to aim its SOA guns at popular solutions because of the perceived benefits of high service reuse. But mainstream SOA leaves vulnerabilities in the business to be exploited by constant change. IT can manage change risk more comprehensively and align itself with the business more closely by offering a broad assortment of "one-off" services in the Long Tail market.

Neither mainstream SOA nor Long Tail SOA is effective alone. They are complementary strategies that should be blended together to maximize the benefits of Service Oriented Architecture. In hybrid SOA, mainstream services provide the foundation for enterprise-wide networks of service-oriented value chains. These value chains flow down into the Long Tail and deliver specialized just-in-time services to small teams at the edges of business — where agility matters most. ■

About the Author

Marc Rix is a lead SOA solutions architect at SAIC, focused on accelerating key business activities through SOA and BPM. He has been building enterprise-scale integration solutions for the past 11 years.

marc@marcrix.com

Synergize WSRP & AJAX with ESB for Customizability & Security

Exploiting complementarities of Web 2.0 and SOA

BY KRISHNENDU KUNTI, UJVAL MYSORE AND NAGARANI BADVEETI

SOA has come a long way from a concept to wide-scale adoption by the enterprise at multiple layers of IT. SOA implementation at the UI layer is the latest in SOA adoption trends. SOA has manifested itself in a number of flavors like creation of a rich user experience by using technology like AJAX (e.g., Google Maps), provisioning value-added services by mashing up data from multiple sources (e.g., chicagocrime.org), community-based peer-to-peer interactions (e.g., Facebook and Flickr), creating collective intelligence (e.g., Digg and del.icio.us), creating collaborative platforms often catering to a trusted community, and creating modular content-based sites (e.g., Marumushi.com).

All of these adoption models use an architecture best suited to their purpose that includes techniques such as of Java scripting at browser end to asynchronously fetch data (e.g., Google Maps), creating content mashups at server end from multiple data sources like RSS or screen-scraping techniques, and creating mashups purely at client end, which might be difficult due to browser-based restrictions (e.g., Mozilla restricts direct data access from multiple machines from a single browser session). The core idea is there's no single technique to address different kinds of scenarios for SOA adoption at the UI layer.

Scenarios that require aggregation and customization of reusable UI components along with value-added features of rich user experience and security can greatly benefit from using technologies like AJAX and WSRP. Picking up on that point, we have proposed an architecture involving WSRP and AJAX that not only takes care to present a customized UI to the user but can also handle non-functional requirements like security, which assumes significance in creating trusted business communities.

In the proposed architecture (as shown in Figure 1) an Enterprise Service Bus manages UI customization based on a user profile by transforming a markup fragment generated from the WSRP producer. The ESB also handles features like security and caching and acts like a single-point proxy for WSRP producers and the Web Services there by addressing any browser-side restriction as we said. In this architecture AJAX controls are used to do partial updates of remote portlets by making a direct call to the ESB instead of routing the call from the browser to the WSRP producer via the WSRP consumer thereby achieving significant performance benefits and a rich user experience.

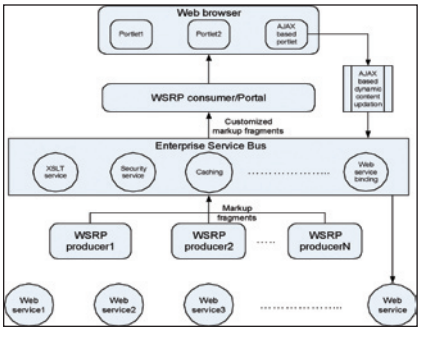


Figure 1:

The proposed architecture also handles overall security concerns like authentication, single sign-on (SSO), authorization, message confidentiality, and privacy. Consider a business scenario where a user needs to access data from more than one portlet from different WSRP producers either simultaneously or one after the other. In either of these cases the user credentials should only be validated once; this feature is provided using WSRP Security. Features like selective access of portlets to certain users, message encryption, digital signatures, and SSO/secure access to enterprise services from browser-based AJAX controls is also provisioned using the enterprise service bus. Using WSRP and AJAX along with an ESB is an interesting proposition in a number of real-world scenarios that need to share a dynamically customized UI based on consumer profiles and provisioning of rich user experience, while catering to security requirements conducive to a trusted business community.

References

- OASIS Web Services for Remote Portlets (WSRP). OASIS. http://www.oasis-open.org/committees/wsrp/ (accessed Dec 11, 2007).
- RSS 2.0 Specifications. http://blogs.law.harvard.edu/tech/rss (accessed Dec 11, 2007).
- Padmanabhuni Srinivas; Kunti Krishnendu; Sahoo Lipika; Bharti Shaurabh. "Coupling RDF/RSS, WSRP and AJAX for Dynamic Reusable Portlets: An Approach and a Use Case." http://ieeexplore.ieee.org/application/enterprise/entconfirmation.jsp?arnumber=4278734 (accessed Dec 11, 2007).
- WSRP Security Technical Note, OASIS. http://docs.oasis-open.org/wsrp/TechNotes/wsrp-security.doc

About the Authors

Krishnendu Kunti works as a technical architect at SETLabs SOA Center of Excellence, Infosys Technologies, Ltd.
krishnendu_kunti@infosys.com.

Ujval Mysore works as a technical specialist at the SOA/Web Services Center of Excellence at SETLabs, the research arm of Infosys Technologies, Ltd.
ujval_mysore@infosys.com.

Nagarani Badveeti is a software engineer working at the Web Services Center of Excellence at SETLabs, Infosys Technologies, Ltd.
nagarani_badveeti@infosys.com.

EVERYTHING
ABOUT JAVA™ TECHNOLOGY. AND SO MUCH MORE.

You won't want to miss the JavaOne conference, the premier technology conference for the developer community. This year's Conference presents the latest and most important topics and innovations today to help developers access even more richness and functionality for creating powerful new applications and services.

LEARN MORE ABOUT

- Web 2.0
- Rich Internet applications
- Compatibility and interoperability
- Open source
- E-commerce collaboration
- Scripting languages
- Java Platform, Standard Edition (Java SE)
- Java Platform, Enterprise Edition (Java EE)
- Java Platform, Micro Edition (Java ME)

MORE
OF WHAT YOU NEED

200+ technical sessions

More than **100** Birds-of-a-Feather sessions

15 Hands-on Labs

Save $200
on Conference registration!

Register by April 7 at
java.sun.com/javaone

Please use priority code: **J8DA1SC**

Java™

JavaOne℠ Conference | May 6–9, 2008
JavaOne℠ Pavilion: May 6–8, 2008, The Moscone Center, San Francisco, CA

Sun microsystems

Platinum Cosponsors

AMD intel ORACLE

Cosponsors

InterSystems Microsoft

BLU AGE PARASOFT

EDI to XML: A Practical Approach

Accessing or creating EDI messages as XML

BY CARLO INNOCENTI

While EDI transactions account for most worldwide commercial activity, XML-based alternatives are beginning to gain traction. According to Forrester Research stateful XML, stateless XML, and even flat file exchanges are all projected to grow at a faster rate than EDI over the next few years. The firm predicts stateful XML transactions will be required for a growing number of B2B process-oriented transactions and are projected to exceed the growth of EDI transactions over the next five years.

This article will discuss how today's B2B support infrastructures must be designed to take advantage of both new and old technology to meet the full range of an organization's external integration needs and support process improvement efforts across the entire value chain. This article will illustrate how to use XQuery and XSLT on non-XML data – bridging the legacy gap using technologies currently deployed and showing how EDI and other legacy formats can co-exist with XML for maximum gain.

How Things Were

One of the challenges that B2B infrastructures have to face is making multiple business partners communicate with each other "understanding" what the information they're trying to exchange is and what it means. It's not a new problem; human beings after all started facing this issue long before software platforms came into being.

In the mid-90s, when B2B infrastructure started becoming relevant, vendors cleverly thought that agreeing on standards describing the messages being exchanged through the B2B platforms would be useful; that's why EDI became popular; EDI provided a well-defined, compact way to exchange messages, providing self-contained information about what the message is about (an invoice, purchase order, healthcare enrollment request), and where inside the message the relevant information is (here's the list of ordered items, here's the address where to ship the merchandise, here's the name of the member).

That's why large EDI standards like X12, EDIFACT, IATA, HL7, Odette, and SWIFT came out, each of them controlled by a specific standard body, each of them with their little syntax peculiarities, each of them interpreted slightly differently by each vendor; but still, standards!

At the end of the 90s it was starting to become clear that EDI, the text formats (comma-separated value files, for example) and binary formats that were widely used in all industry verticals lacked a number of important characteristics: they weren't (easily) human readable; they required dedicated parsers for each version or sub-version or dialect or interpretation of the format; and, probably most importantly, they weren't easily extensible. Those are some of the reasons that led to the creation of XML (eXtensible Markup Language).

XML is human-readable; no matter what the XML structure is a single standard parser can process it; and XML is meant to be easy to extend (after all, that's what the "X" stands for!). Yes, XML is much more verbose than other ways of exchanging information, but hardware and bandwidth made a lot of progress by the late 90s, and IT professionals were willing to trade the cost of dealing with larger messages for the benefits that doing so brings to the table.

How Things Are

Fast forward several years. One might expect that all B2B infrastructures has switched to XML; all information among companies exchanged as XML under the control of standard XML Schemas that enforced the validity of every XML message moved on the wire. Developers only need to focus on dealing with the XML data model in their applications and integrating new business partners in the B2B infrastructure is a breeze.

Well, reality is very different. It's true that XML has become the major standard for exchanging information in B2B infrastructures in most industry verticals; and it's true that XML has replaced the prominent role of EDI formats or proprietary text or binary formats in doing so. But it's surely not true that the use of non-XML formats in B2B contexts has disappeared: in many contexts EDI is still the most adopted way to exchange information (think about SWIFT); in some other cases standard bodies provide both EDI and XML formats for their own specifications (think about IATA or HL7), and your organization may end up being in the situation where it's dealing both with vendors that are still using EDI formats and with others that have switched to XML. And to make things even more complicated, services on B2B infrastructures are usually developed using languages like Java or C#, which create yet-another level of indirection with the data model that has to be used for information to be exchanged (XML, in the lucky cases) or even stored (tables).

There's Hope

In the past few years the XML community has tried to find a remedy for this situation. So-called native XML languages have been in-

Register Today for the 2008 CommunityOne Conference

May 5, 2008 | The Moscone Center | San Francisco, CA

Advance your skills in 70+ sessions led by experts and contributors from more than 30 projects spanning the opensource stack and powering the Internet:

- Scripting and RIAs: PHP, AJAX, Python, Ruby, Javascript, JavaFX, and jMaki
- Tools: NetBeans, Eclipse, Subversion, and Others
- Web 2.0 Applications: Social Networks, Mashups, and Web Services
- Chip Design: OpenSPARC
- Web Scale Computing: HPC, SaaS, Utility/Cloud, Open SSO, and Project Darkstar
- Operating Systems: OpenSolaris and Linux Distributions
- Web and Application Servers: GlassFish, Apache, and JBoss
- Databases: MySQL, PostrgeSQL, and Java DB
- And more: Music, Robotics, ODF, OpenDS, OpenJDK, OpenOffice.org, Mobile & Embedded, and Strategy

PLAN TO STICK AROUND

You'll have complimentary access to the 2008 **JavaOne**SM conference general sessions and Pavilion on Tuesday, May 6.

ATTENDANCE IS FREE, BUT SPACE IS LIMITED.

developers.sun.com/events/communityone

Please use referral code: **CARS**

"CommunityOne is a must attend event. It's not just dialed into the future, but also knows how we want to get our information and tools."

- 2007 CommunityOne Attendee

COME EARLY. STAY LATE.

GENERAL SESSION
9:30 am - 10:45 am

TECHNICAL SESSIONS
11:00 am - 6:00 pm

COMMUNITY RECEPTION
6:00 pm - 7:30 pm

CommunityOne, 999 Skyway Road, Suite 300, San Carlos, CA 94070 866-405-2517

troduced at least to make it easier for IT organizations to deal with XML structures, eliminating the need for bridging between the XML and the Java or C# data models whenever one had to deal with or create XML documents. First XSLT and then XQuery have provided a big help in that direction. Even the more recent Linq-for-XML approach introduced by Microsoft moves in the same direction. Both XQuery and XSLT are languages that are designed to work against the XML Data Model (XDM); when developers write XQuery or XSLT the only data model they're dealing with (to access/query it or create it) is XML. The most obvious reaction that IT organizations may have is that neither XQuery nor XSLT are well suited to dealing with a set of different data sources, like XML messages, EDI messages, or text or binary formats.

But if you think carefully about it, what XQuery and XSLT really require is for the data they're working to look like XML; the data must be available as a XML Data Model; it doesn't really matter what the underlying physical representation of such data is as long as XQuery or XSLT can see it as XML. Considering that XML provides a highly flexible way to represent information, it's definitely possible to think about a variety of ways in which the data contained in an EDI message, for example, can be interpreted as XML. That's an intriguing concept: provide a way to access or create EDI messages as XML, and my B2B infrastructure can be designed to just deal with XML: there's no need to have special cases meant to support business partners that are dealing with EDI or even proprietary text or binary-based formats.

A Practical Approach

That all sounds very interesting; but in practical terms how can that work? Are there ways to make XSLT or XQuery processors deal with EDI as if it was XML?

One simple yet powerful way to make most existing XQuery and XSLT engines able to process EDI as XML is to make the processor able to access non-XML data through custom URI resolvers. What is a URI resolver? When an application tries to access a resource referenced as, for example, file:///c:/myFolder/myDocument.xml, or http://myServer/myFolder/myDocument.xml, what happens under the cover is that the standard URI resolver understands that "file:" identifies a resource on your local file system and it retrieves the file myDocument.xml in the myFolder directory; or that "http:" identifies a request to access a resource through the HTTP protocol, and it connects to myServer to GET myfolder/myDocument.xml.

And that mechanism is extensible: you can, for example, create and use a custom URI resolver that knows how to deal with a specific URI scheme. For example, if your application references a URI like converter:EDI?file:///c:/myMessage.edi, the custom URI resolver will know that what the application is requesting is to retrieve an XML interpretation of the EDI message myMessage.edi.

That's quite a powerful mechanism; applications can leverage it to do what we described above: give XQuery or XSLT processors the ability to handle both XML and non-XML data transparently as if everything was in fact XML.

For example, an XQuery could be in charge of retrieving all the book orders received as part of an incoming EDI message from a business partner, and return them in an XML format required by your application. Assume the EDI incoming message looks Listing 1 and that the XQuery processing the EDI message looks like Listing 2. The result of such XQuery would be something like Listing 3. As you can see, the XQuery doesn't know about the fact that you're querying an EDI message to generate XML: from the XQuery point of view, you're dealing strictly with XML.

A similar extension can apply on the output side. Suppose I want to answer my business partner using an EDI message (since that's all that partner can process); of course I'd like to be able just to deal with the generation of the proper response in XML terms, and then assume that what the business partner gets is actually EDI. You can do that, for example, by extending the concept of output method. Both XQuery and XSLT already support the ability to instruct the processor to serialize the result of their processing not only as XML, but also as HTML, XHTML, or event text. But what if you were able to tell your XQuery processor "serialize the result as EDI?" Your application would be able to focus on the fact that it's dealing with querying and creating XML, but its result would in fact be EDI.

For example, the XQuery in Listing 4 (where only a portion is displayed) would generate an EDI message as the result as shown in Listing 5.

Conclusion

Even if XML is becoming more and more widespread, modern B2B applications still need to deal with a variety of formats for incoming and outgoing messages. EDI is still widely used, especially in some specific market verticals, and it seems unlikely that its use will vanish any time soon. Being able to have infrastructures that can deal with heterogeneous data formats without forcing IT organizations to worry about what such data formats are and how to deal with them is crucial. Languages like XQuery and XSLT that are meant to work against the XML Data Model provide a great way for abstracting the physical details of the data that has to be manipulated or created and focus only on handling XML data. That approach allows EDI, text, or binary formats to be queried or even generated as if they were plain XML. There are already XQuery implementations on the market that expose such functionality, and that even extend that approach to the relational world, making it possible to deal with relational data, XML, and non-XML documents as if they were all expressed as an XML Data Model.

About the Author

Dr. Carlo Innocenti is senior XML program manager at DataDirect Technologies (www.datadirect.com), responsible for the overall strategy and direction of the XML products group including DataDirect XQuery (www.xquery.com), DataDirect XML Converters (www.xmlconverters.com), and Stylus Studio

carlo.innocenti@datadirect.com.

Listing 1

```
UNA:+.? '
UNB+UNOC:4+SSTUDIO:1+DDIRECT:1+20051107:1159+6002'
UNH+SSDD1+ORDERS:D:03B:UN:EAN008'
BGM+220+BKOD99+9'
DTM+137:20051107:102'
NAD+BY+5412345000176::9'
NAD+SU+4012345000094::9'
LIN+1+1+0764569104:IB'
QTY+1:25'
FTX+AFM+1++XPath 2.0 Programmer?'s Reference'
```

The Number One SOA Event
Comes to New York Colocated with the Virtualization Conference & Expo
June 23-24, 2008 at The Roosevelt Hotel

Service-oriented architectures (SOAs) have evolved over the past few years out of the original vision of loosely coupled web services replacing constrained, stovepiped applications throughout enterprise IT. Every major enterprise technology vendor today has developed its own SOA strategy, supported by innumerable mid-size companies and start-ups offering specific SOA aspects or entire solutions. This explosive growth in SOA technology is in response to a global demand--IDC estimates that spending on SOA services alone will grow from $8.6 billion to more than $33 billion by 2010.

SOA World Conference & Expo 2008 East brings together the best minds in the business to New York for a two-day conference that offers comprehensive coverage of SOA and what it means to enterprise IT today. As Zapthink analyst Jason Bloomberg has noted, "SOA involves rethinking how the business leverages IT in many various ways." Attend SOA World Conference & Expo 2008 East and learn from more than 100 speakers about how SOA is transforming business--and the way IT and business managers think about their businesses, processes, and technology.

The colocation of SOA World Conference & Expo 2008 East and Virtualization Conference & Expo delivers the most relevant content to IT and business. Conference attendees will be able to choose from four great tracks at this year's event. Mix and match all you want, or slot into your favorite topic for the duration! The tracks include:

- Web 2.0/AJAX and SOA
- Interop Standards and Services
- Real-World SOA
- SOA Technology
- Virtualization
- Specially Selected Hot Topics

Who Should Attend?

CEOs and CTOs, senior architects, project managers, Web programmers, Web designers, technology evangelists, user interface architects, consultants, and anyone looking to stay in front of the latest Web technology.

REGISTER TODAY AND SAVE!

www.soaworld2008.com

```
LIN+2+1+0764569090:IB'
QTY+1:25'
FTX+AFM+1++XSLT 2.0 Programmer?'s Reference'
LIN+3+1+1861004656:IB'
QTY+1:16'
FTX+AFM+1++Java Server Programming'
LIN+4+1+0596006756:IB'
QTY+1:10'
FTX+AFM+1++Enterprise Service Bus'
UNS+S'
CNT+2:4'
UNT+22+SSDD1'
UNZ+1+6002'
```

Listing 2

```
<order>
    {
    for $GROUP_28 in doc('converter:EDI?file:///c:/ order.edi')/EDI-
FACT/ORDERS/GROUP_28
    return
    <book>
        <quantity>
            {$GROUP_28/QTY/QTY01-QuantityDetails/QTY0102-Quantity/
text()}
        </quantity>
        <ISBN>
            {$GROUP_28/LIN/LIN03-ItemNumberIdentification/LIN0301-
ItemIdentifier/text()}
        </ISBN>
    </book>
    }
</order>
```

Listing 3

```
<order>
    <book>
        <quantity>25</quantity>
        <ISBN>0764569104</ISBN>
    </book>
    <book>
        <quantity>25</quantity>
        <ISBN>0764569090</ISBN>
    </book>
    <book>
        <quantity>16</quantity>
        <ISBN>1861004656</ISBN>
    </book>
    <book>
        <quantity>10</quantity>
        <ISBN>0596006756</ISBN>
    </book>
</order>
```

Listing 4

```
declare option ddtek:serialize "method=EDI";
<X12>
    <ISA>
        ...
    </ISA>
    <GS>
        <GS01-FunctionalIdentifierCode>CT</GS01-FunctionalIdentifier-
Code>
        <GS02-ApplicationSendersCode>{/metadata/sendersCode/text()}</
GS02-ApplicationSendersCode>
        <GS03-ApplicationReceiversCode>{/metadata/receiversCode/
text()}</GS03-ApplicationReceiversCode>
        <GS04-Date>{/metadata/date/text()}</GS04-Date>
        ...
    </GS>
    <TS_831>
     ...
    </TS_831>
    <GE>
     ...
    </GE>
    <IEA>
     ...
    </IEA>
</X12>
```

Listing 5

```
ISA+00+ +00+ +01+1515151515 +01+5151515151 +041201+1217+^+00403+00003
2123+0+P+*'
GS+CT+9988776655+1122334455+20041201+1217+128+X+004030'
ST+831+00128001'
BGN+00+88200001+20041201'
N9+BT+88200001'
TRN+1+88200001'
AMT+2+100000'
QTY+46+1'
SE+7+00128001'
GE+1+128'
IEA+1+000032123'
```

2008 VIRTUALIZATION CONFERENCE + EXPO

www.virtualizationconference.com

The First Major Virtualization Event in the World!

June 23-24, 2008 New York City

Reaching Beyond Physical IT: Explore the Power of Virtualization in Enterprise Computing

Virtualization, the fundamental technological innovation that allows skilled IT managers to deploy creative solutions to such business challenges, is rapidly evolving from taking one computer and making it appear to be many computers, to virtualizing almost any information technology resource.

Virtualization of distributed computing, of IT resources such as storage, bandwidth, and CPU cycles are all increasing significantly in importance within enterprise IT. Virtualization can also apply to a range of system layers, including hardware-level virtualization, operating system level virtualization, and high-level language virtual machines.

In short, Virtualization is being adopted by IT organizations around the world so rapidly right now that, according to research firm IDC, spending on virtualization software and services is expected to exceed $15 billion worldwide by 2011 - up from $6.5 billion in 2006.

The Virtualization Conference & Expo 2008 East, taking place on June 23-24, 2008 in New York City, is the leading event covering the booming market of Virtualization for the enterprise.

TOPICS WILL INCLUDE:

- I/O Virtualization
- Application Virtualization
- Desktop Virtualization
- Network Virtualization
- Physical to Virtual (P2V) Migration
- Server Virtualization
- Storage Virtualization
- Virtual Machine Automation
- File Virtualization
- Management Applications, Tools and Utilities
- Virtualization Scripts and Procedures
- Paravirtualization
- Real-time Virtualization
- Device Virtualization
- Thin Client Solutions
- Virtual Desktop Infrastructure
- Greening of IT
- Vista & Virtualization
- Microsoft's Remote Desktop Protocol
- Virtual Machine Desktops
- Security & Performance Issues
- Virtualization Security Audits
- Utilization and Performance
- Virtual Backup
- Systems Integration
- Cloud Computing
- High Availability (HA)
- KVM (Kernel-based Virtual Machine)
- Virtual Appliances
- and more...

REGISTER TODAY AND SAVE!

www.virtualizationconference.com

Growing an SOA Garden

Turn flawed perspectives into something positive

BY STUART SMITH

Adopting SOA is a lot like gardening. It takes time, skill, a lot of hard work, and the process can be messy and even a bit frustrating at times. I know you've probably heard tons of different analogies that attempt to put SOA and governance into everyday terms and I'm sure that growing the SOA "garden" through governance won't be the last. I began thinking about this when faced with a client who seemed to be coming at things from all the wrong ways and expected me to grow the whole garden in the week we had together. At the end of the engagement all parties walked away with something that was very successful, not because I tried to convince them to approach things differently, but because I focused on how the positive elements of what they were saying could be turned into actionable items going forward.

In these situations the client already knows what "garden" they want to grow. It's our job to work with that vision, not convince them to grow a different garden, and simply help them avoid the "weeds" that will destroy what they're trying to do. We are trying to teach them how to grow their own "SOA garden" so they'll have the skills they need after we leave.

In my opinion consultants in SOA engagements may see an increasing number of these situations in the near future from clients that know they want to do SOA and have been hearing the drumbeat from the industry saying "Governance, governance, governance..." In these situations I think it will be successful to focus more on offering constructive guidance to turn whatever these companies, who want to do SOA and know they need governance, say to us into positive action.

Let's take a look at some of the flawed perspectives that may be offered and how to turn them into something positive...

1. We have SOA because we bought the products but now we need governance.
2. We need all the details of governance *today!*
3. Can you give us the governance artifacts from another company?
4. How do we bring everyone to the same level of maturity?
5. We think this will be part of our SOA governance, how do we make it happen?

We have SOA because we bought the products but now we need governance. Obviously the biggest problem with this is a view that SOA is something you buy. Thankfully the vendors that have been preaching this to these clients weren't able to convince them that SOA governance is also something you simply buy. It's critical in these situations to reinforce the idea that SOA governance, even more than SOA in general, isn't something you buy but something you do. Growing a garden isn't simply buying all of the right materials, but is also the effort you put into using those materials to create a beautiful garden. I think in these situations it will be important to evaluate what products and tools the client has, what capabilities those products have in relation to governance, and how those tools might be leveraged, even if only in the short-term. The reality is that if the products are already bought, the organization won't have to go through the process of lobbying for money. Hopefully they have some product that can become a valuable part of their governance development process. As long as we can get the commitment they may need to reassess what products they may need in support of governance in the not-so-distant future, they can start doing governance today.

We need all the details of governance *today!* The driving force behind this motivation is that those tasked with addressing the need for SOA governance want to be able to report to their superiors "we are gardeners because we finished the garden." The attitude that an outside consultant will have to change is that all details have to be in place before they can claim this. An important aspect to focus on in this situation is that an element of governance is understanding how the governance mechanisms themselves will grow over time. We can also focus on how they might be able to use the governance structures they already have in place and simply make small adaptations to adjust to SOA. If we can get these clients to focus on developing a plan for how they will develop SOA governance mechanisms, they can understand how to grow the maturity of the governance itself. Then they can say "we have SOA governance and we have a plan for making it better!"

Can you give us the governance artifacts from another company? Besides the obvious non-disclosure issues, this is the "can you give us someone else's garden?" mentality. The thought process is that if other organizations have already gone through the process and suffered the growing pains, perhaps we can avoid the same thing by dropping in place what they did. Although templates for service contracts, policies, and methodology might provide some guidance, it's important to focus on the fact that these will have no value until they've been customized for and validated on the client's environment. As long as we can get the client to commit to asking, "would this template work for us?" we can offer some generic artifacts to start the process, like using bulbs or young plants rather than seeds.

How do we bring everyone to the same level of maturity? The issue with this question is that the client thinks they should focus on

those elements that lag behind while ignoring or even holding back those elements or groups that may be more mature. The positive part here is that they recognize they'll need to pay special attention to those groups that lag behind the broader level of maturity. Rather than trying to get the entire garden to grow at the same rate, you'll have to get them to become experts in the dual role of growing the garden from scratch and tending the established garden. In fact, in the early stages of SOA adoption it's probably more important to identify and leverage those groups or areas that are more mature as opportunities to be more effective with the limited resources initially available. Establishing a "selective SOA" process can help identify those areas that can provide more immediate returns, even if that only includes abstract (and more difficult to measure) benefits like "better connectivity" and "more robust architecture." Another aspect of this approach is not to try to force SOA on those that may not see the value. If we see potential for value but those involved with the projects don't, we should move on to the next potential valuable project rather than try to "force a fit" to SOA that carries more risk of failure, something we want to avoid early in the adoption process. Later on, when we can show off the beautiful gardens we can grow, it will be easier to turn these people into gardeners.

We think this will be part of our SOA governance, how do we make it happen? In this situation the client is presenting the SOA consultant with a specific project they would like to do in support of what they currently believe to be SOA governance. It's important not to shoot this down unless they're completely off base, after all, we don't want to throw out a plant they think is beautiful simply because we don't happen to like it. Hopefully they've heard enough about SOA governance that this project is something that can be an element, just not the entirety, of their SOA governance mechanisms. If we take the approach of rejecting their ideas and trying to force them down a different path, we'll be doing the same thing we're trying to tell them *not* to do for their early projects. It's more important to focus on how to turn their project into a successful part of their early governance, hopefully also using the products they already have. It will be more important to assess the lifespan of that project as an element of their SOA governance, whether the project needs to be enhanced, phased out, or adjusted in their future governance plan, and what other projects might be good to focus on to evolve the governance structure towards further maturity.

Although these examples certainly don't cover every situation an SOA consultant might find himself in, I think it provides several common situations that may occur in, most engagements.

Too often clients see the "garden" as being an SOA-enabled enterprise but don't know how to get there. The vendors are going to concentrate on "don't we have great plants!" and not be as concerned with how beautiful everything looks when taken as a whole. As we help get clients started with how they're going to become an SOA-enabled enterprise, I think it's important to reinforce the good things that they're already telling you and turn that into something they can "grow" now. ■

About the Author

Stuart Smith is an application server and portal administration lead at Web Age Solutions.

stuart.smith@authors.sys-con.com

SOA SDLC: On-Demand

Sensors & on-demand visibility

BY RAMI JAAMOUR

Spending time with my parents over the holidays got me to thinking about the differences between this generation and the previous one.

My parents expect to spend a certain amount of time and effort managing certain aspects of their lives. For example, when they drive to an unfamiliar vacation spot, they inquire about directions and even write or plot the route before they head out. Whereas for me, it is a matter of popping out an iPhone or a GPS device, saving time, improving accuracy, and avoiding the mistakes of manually drafting the directions.

The same on-demand principles apply to SOA lifecycle evolution. When an infrastructure that allows for changes on-demand is established, you need the agility to execute on them quickly and safely while preserving the quality of the overall system. This is a challenge that organizations face — attempting to change their underlying SOA systems without compromising existing business processes. However, this challenge can be solved with ubiquitous on-demand information.

Safe SOA Evolution

The fact of the matter is that you almost never start anything from scratch. Instead, you're always evolving what you already have. So the key to an agile SOA is to look into the various change activities that combine in a service lifecycle. These change activities include change inception, change elaboration and impact analyis, construction, and finally, transition into production.

There are three important points here: first, you can't afford to spend time and resources repeating the same manual activities associated with each of these phases every time a change is needed, such as changes to the environment or testing and impact analysis. Second, the change processes are increasingly iterative; the days of the waterfall model, starting with requirements and design and ending with testing, are gone. Third, the phases I listed – which are mostly based on RUP (Rational Unified Process) – don't include testing as an isolated phase. In fact, you should never have an explicit, sequential testing or validation phase if you want to have an agile, quality process that produces quality results.

The more time you spend doing tests on changes made in your underlying systems, the more you are likely to compromise the agility of your SOA, and risk quality and continuity.

Quality needs to be baked in. You don't test it out of an application. I'm by no means suggesting that you shouldn't test and validate, but the process of testing and validating against established policies needs to be continuously applied throughout the SDLC process, and information pertaining to these policies must also ubiquitously exist, on-demand, in the underlying environment. Just like when you're driving, you wouldn't pull out a piece of paper to draw up a new map every time the route or destination altered, you simply know where you are and have full situational-awareness then you set the new goal to get there from where you are. The same principles apply to the SDLC of software in general, and SOA in particular, because SOA's main goal is to increase agility.

So, efforts and resources have to be spent on establishing the infrastructure that provides on-demand visibility into the quality of the underlying artifacts. When you decide on a change in a system, its impact has to be analyzed immediately in terms of what existing services are going to be affected and what business processes rely on these services. What this means is that while the change is being worked on, developers have to be able to execute the relevant tests and validations immediately on the components that are impacted directly by that change.

This is achieved by instituting process sensors at different points in the lifecycle to ensure quality at each of these points. As soon as the change activities commence, modifications to code or any service metadata artifacts (such as WSDLs and BPEL models) are validated, regression tests are executed at the code, service, and business process layers to detect undesirable side effects or deviations from the existing functionality. Such multi-layer process sensors are similar to a manufacturing assembly line. They ensure that whatever is produced consistently adheres to established quality standards. These sensors also provide the on-demand visibility needed to achieve full situational awareness and quality measures anytime during the process. So let's talk a bit more about these sensors and how they're applied.

Design and Development Policies

Quality is best defined with a consistent policy. Just like runtime security requirements, or how service use and availability are described with policies, quality can also be asserted with a set of declarative policies. Such quality policies, which I'll refer to as design and development policies, define how the various artifacts are changed and produced, and what standards and rules they need to adhere to. For example, a policy may specify that developers with a certain role review code changes before they're committed

The World's Leading Java Resource Is Just a >Click< Away!

JDJ is the world's premier independent, vendor-neutral print resource for the ever-expanding international community of Internet technology professionals who use Java.

SYS-CON MEDIA

Twelve Monthly Digital Editions!

Four Special Show Print Editions: March, June, September, December

www.JDJ.SYS-CON.com

or 1-888-303-5282

Advertiser Index

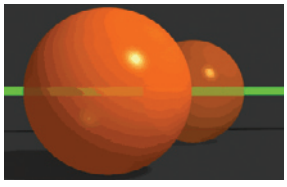
ADVERTISER	URL	PHONE	PG
Java Book	www.theriabook.com	201-802-3020	21
B-hive	www.bhive.net	650-358-1300	7
CommunityOne	http://developers.sun.com/events/communityone/	866-405-2517	15
DataServices World	www.dataservicesworld.sys-con.com		36
Fiorano	www.fiorano.com/downloads		5
JDJ	www.jdj.sys-con.com	888-303-5282	23
JavaOne 2008	http://java.sun.com/javaone/sf/index.jsp		13
Managed Methods	www.managedmethods.com		35
AJAX Book	www.realworldajaxbook.com	201-802-3020	11
SOA Conf & Expo	http://soaworld2008.com/		16
Sterling Commerce	www.sterlingcommerce.com/B2BCommunications		25
SYS-CON Media	www.sys-con.com	201-802-3020	33
VirtConf & Expo	www.virtualizationconference.com	201-802-3020	19
Web Age Solutions	www.webagesolutions.com	877-517-6540	2

General Conditions: The Publisher reserves the right to refuse any advertising not meeting the standards that are set to protect the high editorial quality of ***.Net Developer's Journal.*** All advertising is subject to approval by the Publisher. The Publisher assumes no liability for any costs or damages incurred if for any reason the Publisher fails to publish an advertisement. In no event shall the Publisher be liable for any costs or damages in excess of the cost of the advertisement as a result of a mistake in the advertisement or for any other reason. The Advertiser is fully responsible for all financial liability and terms of the contract executed by the agents or agencies who are acting on behalf of the Advertiser. Conditions set in this document (except the rates) are subject to change by the Publisher without notice. No conditions other than those set forth in this "General Conditions Document" shall be binding upon the Publisher. Advertisers (and their agencies) are fully responsible for the content of their advertisements printed in ***.Net Developer's Journal.*** Advertisements are to be printed at the discretion of the Publisher. This discretion includes the positioning of the advertisement, except for "preferred positions" described in the rate table. Cancellations and changes to advertisements must be made in writing before the closing date. "Publisher" in this "General Conditions Document" refers to SYS-CON Publications, Inc. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

to a certain source control system; and according to the policy, the code changes need to follow certain maintainability rules, WSDLs need to pass WS-I BP compliance, messages must use a certain version of SOAP and adopted WS-* specifications, and schemas must adhere to a set of best practices around design and type definition. These policies essentially define the structural requirements of the artifacts and as they're enforced and measured with proper sensors by the underlying infrastructure, they eliminate certain defects or interoperability issues from getting into the system.

Functional Regression Tests

Regression tests are what ensure that what worked yesterday is still working today. To capture system functionality best, they need to be applied at several layers: the code/class layer with JUnit or NUnit tests, the service layer with functional operation tests, then the process layer with tests that exercise the end-to-end transactions as defined by the various business process use cases. Once these various test assets are created, they're maintained and evolved along with the application artifacts. As they execute continuously and automatically in the background, they ensure that when a change takes place it didn't break any existing functionality. However, the key here is that these tests are created and maintained to be automatically runnable. To the largest extent possible, regression tests shouldn't consume time or resources to be executed, only to be created and maintained. This is a critical principle that, once applied successfully, provides a huge improvement in both process agility (in terms rapid delivery), and test coverage. I've seen Fortune 1000 organizations achieve high double-digit gains on these two fronts.



System Performance

Like quality, best practices, and functionality tests, performance validation can be executed in a continuous fashion to provide up-to-date results. This is an alternative to leaving it as an activity at a later integration cycle, which often leads to surprises, delays, and unpredictability. When load tests execute against isolated parts of the system, even in the development environment when changes are being made on the system day to day, the quality-of-service metrics can be monitored for any abnormal deviations. The absolute metrics around the desired performance of the system in a development/test environment is certainly different from the live system, and they're hard to determine. However, that's not important. What's important is to capture a baseline of these performance metrics (such as execution times, memory utilization, etc.) and have the subsequent load test run results compared to that original baseline. Once this is put in place, performance side effects or configuration changes in the application would be detected immediately and addressed. The lack of such immediate detection and on-demand information results in errors being caught much later when analyzing and resolving becomes harder and more time consuming, and when uncertainty is introduced.

SDLC Process Automation

The various activities of the development process, such as build and deployment, change management and issue tracking, code reviews, and the collaboration activities among the various stakeholders such as audits, approvals, and release procedures, need to be defined and automated as much as possible. In the end, the activities handled by people are the most time-consuming activities — especially when several stakeholders are involved. Therefore, the extent to which these activities can be defined and driven via an automated set of processes to facilitate the collaboration among people will significantly impact agility and the accuracy with which these activities are carried out.

Quality Visibility

With these practices in place, measurement can provide the on-demand SDLC process visibility. At any point in time, an architect or a manager should be able to know the current quality status of an artifact, at the code level or the service contract level. This visibility is best achieved in SOA by feeding these metrics directly into the service registry so it's associated with the related artifacts. For example, not only would the registry show the metadata around a service such as dependencies, access controls, and quality of service metrics, but also quality metadata such as test coverage, adherence to interoperability and organizationally established standards. This is particularly important since service registries today have evolved to take a role in the lifecycle process of a service. This helps the various stakeholders manage the lifecycle so that quality visibility – serving as a health check for the service – is what provides the triggers to promote the service from one state in the process to another, or instills confidence in the stakeholders that it has met the quality criteria needed to go live.

The lack of visibility and an objectively measurable process can lead to the organization and its partners simply not reusing the business components. This lack of trust, which is reasonable when dealing with mission-critical business process, will deteriorate the efficiencies that the SOA model promises.

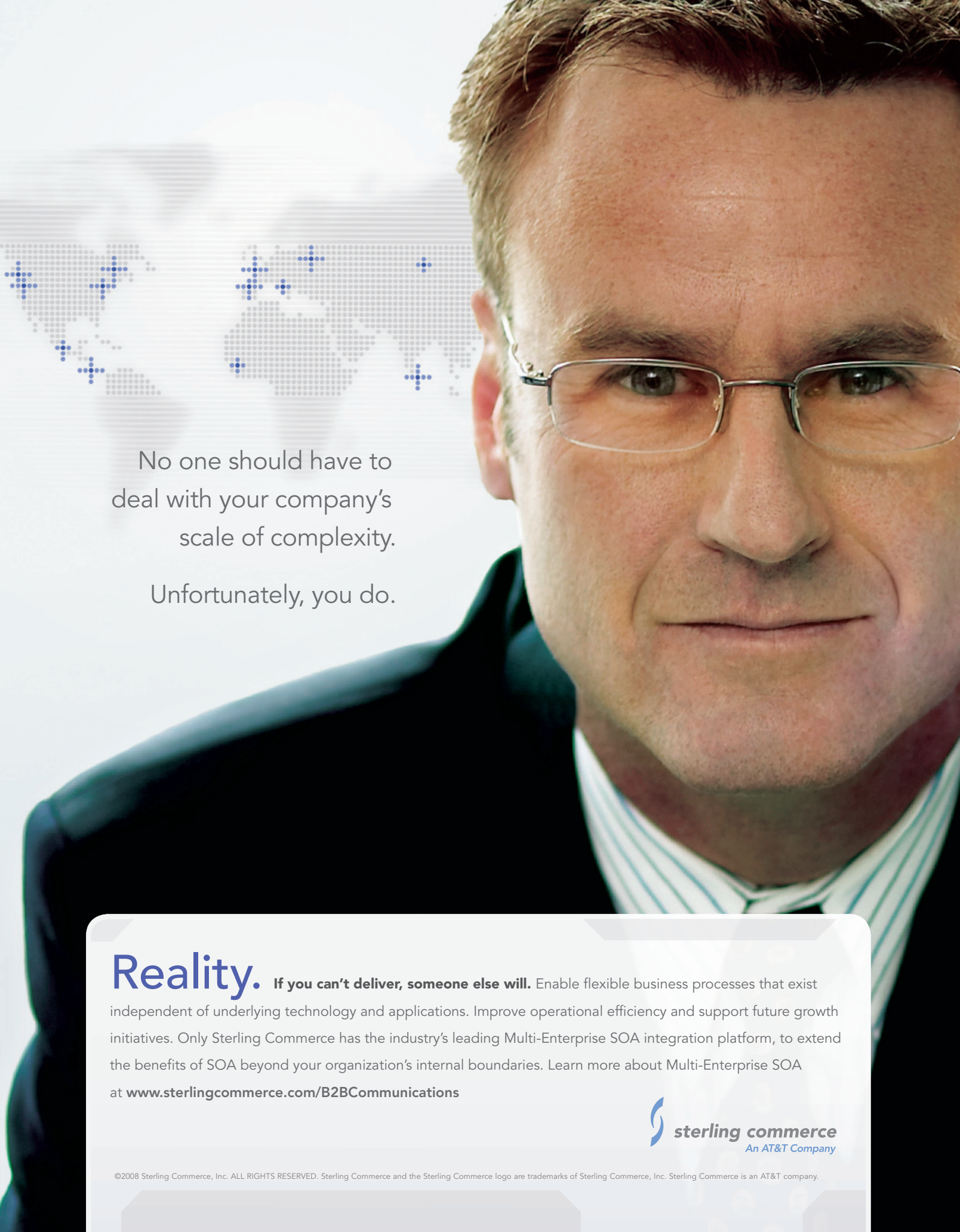
Conclusion

When information is always available, instantly and ubiquitously whenever needed, tasks can be done quickly, easily, and with a high level of accuracy. On-demand information lets you quickly adjust to changing needs and provides alternative choices with little overhead — as long as you have the necessary information readily available and visible.

For an agile Service Oriented Architecture, ubiquitous and on-demand information can be readily available via various process sensors that can be applied throughout the SDLC. Design and development policies, functional regression tests, system performance metrics, DSLC process automation, and quality visibility all provide the necessary information to ensure that your SOA remains agile. ■

About the Author

Rami Jaamour is the product manager for our SOA Solutions at Parasoft. He has contributed to the WS-I Testing Tools Work Group and the Apache Software Foundation, where he contributed to the WSS4J project, an open source WS-Security implementation for Java. Rami has published articles related to Web services security, and spoke at several events related to SOA and Web services. His experience with SOA and Web services includes the development of effective Web services automated testing methodologies and working with many of Parasoft's customers to ensure secure, reliable and compliant Web services.



Reality.

If you can't deliver, someone else will. Enable flexible business processes that exist independent of underlying technology and applications. Improve operational efficiency and support future growth initiatives. Only Sterling Commerce has the industry's leading Multi-Enterprise SOA integration platform, to extend the benefits of SOA beyond your organization's internal boundaries. Learn more about Multi-Enterprise SOA at **www.sterlingcommerce.com/B2BCommunications**

sterling commerce
An AT&T Company

First Step Toward Manufacturing Semantic Interoperability for SOA Adaption Strategy

Without semantics, data has no meaning and is just a binary

BY RAMY ABAAS

Interoperability is the ability of two or more systems to work with each other. In the loosely coupled environment of a Service Oriented Architecture (SOA), separate resources don't have to know how each of them work, but they do need to interoperate with each other by having enough common ground to exchange messages without error or misunderstanding.

On the other hand, semantics is the intended meaning. In computing, semantics is the assumed or explicit set of understandings used in a system to give data meaning. Semantic interoperability is the foundation of a SOA implementation. It ensures that service consumers and providers exchange data consistently and flexibly.

Interoperability wasn't a concern years ago when manufacturing started developing IT systems. Manufacturing has invested heavily in creating large, complex, heterogeneous IT systems to support critical business processes. This investment must be maintained. But to remain competitive, manufacturing must find ways to seamlessly integrate the disparate manufacturing applications within its enterprise business systems and with its suppliers. To simplify the integration of disparate manufacturing systems, semantic manufacturing interoperability should be viewed as a necessity.

Manufacturing interoperability means sharing information seamlessly among the different manufacturing and business applications in the enterprise and with the extended enterprise. This information sharing is now a necessity as well as a challenge. Since interoperability is achieved by implementing standards and semantic is achieved by implementing ontology, this article will explore the opportunities and challenges of manufacturing semantic interoperability and focus on interoperability as a foundation for an enterprise-wide SOA adaption strategy. Ontology deserves another article. We'll also identify the interoperability standards that apply when sharing and exchanging information with partners, and between the plant systems and the business systems (P2B).

Semantic Interoperability Drivers

There's growing interest in semantic interoperability in manufacturing because it:

- **Improves collaboration:** Manufacturing organizations are looking for innovative ways to improve their collaboration with their suppliers and business partners.
- **Reduces integration costs:** Manufacturing organizations are actively seeking new ways to reduce integration costs.
- **Increases business agility:** Today, manufacturers are under tremendous stress and have never seen the shifts are happening. As a result, business agility continues to be on the minds of global executives as a way to stay ahead of the curve or simply survive. SOA hold the key to this business agility.

Semantic Interoperability Challenges

To achieve manufacturing interoperability and information sharing, manufacturers have to overcome the following challenges:

- **Semantics**: Semantics is related to the understanding and integrity of the information and requires an agreed business language. Semantics also requires collaboration across organizations (in the enterprise) and across enterprises (with the suppliers). Achieving consensus on meaning is the most difficult challenge. Agreement on semantics and syntax is difficult to achieve because of:
  - Perceived loss of control (resistance to change): reluctance to give up one's view of the world (process and data)
  - Lack of incentives to cooperate and collaborate: what's in it for me?
  - Cost – lack of a program budget for activities outside the program or the project.

- **Standards:** There are too many overlapping standards supporting manufacturing interoperability such as OAGi, ISA95, MIMOSA, OPC, WBF, ACES, PIES, and ISO 10303 (STEP). These standards cover the interfaces, messages, and documents, but don't cover the business processes. Many of these data interchange standards are adopting XML as the basis for specifying their data content standards, which can be used to tag collections of data with labels. As part of the standardization activity, communities can agree on the names for these labels. An interoperability problem remains however if different people have a different understanding of the meaning of these labels. In other words, XML standardizes the syntax of data exchange, but wasn't designed to capture the semantics of the data. This limitation isn't an issue if used in a common

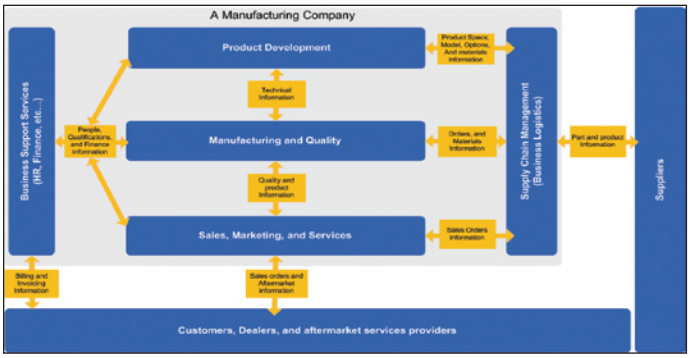


Figure 1: Manufacturing organization interactions

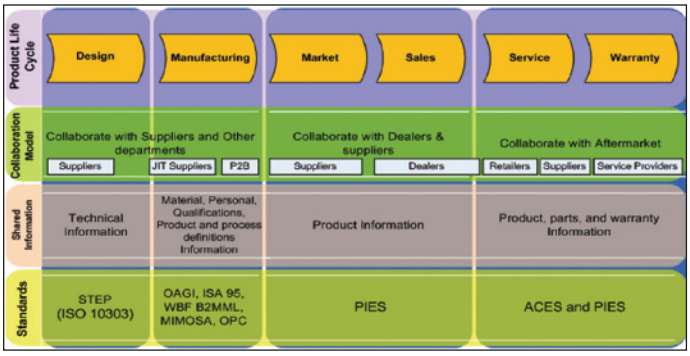


Figure 2: Product lifecycle collaboration

context, but it becomes a problem when moving data from one context to another, for example, sending data from a manufacturing context to a financial context. Without an explicit and rigorous definition of terms, misunderstanding is an inevitable.

- **Globalization:** Globalization is the major trend in manufacturing today — globalization of markets and globalization of partners. The globalization of markets means that companies want to design anywhere, manufacture anywhere, and sell their products anywhere. The globalization of partners means that supply chain members are located anywhere and do business anywhere with the manufacturing. Both have led to an explosion in the amount of information sharing that must take place. It's absolutely critical to the success of companies and their suppliers that this sharing is done correctly, efficiently, and inexpensively.
- **People:** Changes in technology are impacting the way in which information sharing takes place. Nevertheless, people still provide the bulk of the understanding needed to determine what the information means and most of the tacit knowledge needed to make decisions based on that understanding.

Manufacturing Semantic Interoperability Framework

To overcome some of the semantic interoperability issues, manufacturing should create a semantic interoperability framework. This framework is a strategy for achieving a common view of information. It should cover what can be shared for cost and security reasons. Every enterprise has many areas that can potentially benefit from semantic interoperable systems and more cost-effective integration. Interoperability is achieved by implementing standards and semantics is achieved by implementing ontology.

The process of creating this framework consists of:

- Identifying the areas in manufacturing that can benefit from applying the framework.
- Identifying the interoperability standards that can be applied to these different areas.

To identify the different areas in manufacturing that can benefit from applying an interoperability framework, we should start off with the touch points between the different manufacturing organizations, their partners, and the kind of information they share such as product development, manufacturing, quality, and supply chain management. They also share and exchange information with suppliers, customers, and aftermarket services providers.

Figure 1 illustrates manufacturing entities, the type of information exchanged by them, and touch points with external entities.

Interoperability Standards

Manufacturing collaborates and interfaces with a number of external organizations to produce its products. It collaborates with suppliers, dealers, and aftermarket service providers and retailers. It shares different information with these entities such as technical (engineering) information, product and parts information, and aftermarket information such as warranty information. Usually, in some large supply chains, a dominant OEM will mandate that supply chain partners conform to a particular proprietary solution. This has been the practice, for example, in the automotive sector. The problem with this approach is that the interoperability problems are simply pushed lower down the supply chain – they're not eliminated. First sub-tier suppliers are forced to buy and maintain multiple, redundant systems if they want to do business with several major OEMs.

Published standards offer some stability in representing information and help in sharing information consistently. To identify the different interoperability standards, we need to take close look at the product lifecycle, the collaboration model, and the type of information shared at different phases of production with different entities based on the collaboration model. For example, a product goes through a design phase first. At this point, the manufacturer collaborates with its suppliers by exchanging technical information about the design and the different technical specs of the product and its parts. The standard available here for exchanging technical information is ISO 10303 (STEP).

Figure 2 illustrates a product lifecycle, the collaboration entities, the information shared, and the available interoperability standards.

In general, manufacturing standards can be divided into two kinds:

1. Business-to-Business interoperability standards (B2B)
2. Plant-to-Business interoperability standards (P2B)

B2B Interoperability Standards

Below is an overview of the manufacturing standards available:

- **STEP:** The Standard for the Exchange of Product Model Data is a comprehensive ISO standard (ISO 10303) that describes how to represent and exchange digital product information. Nearly every major CAD/CAM system now includes a module to read and write data defined by one of the STEP Application Protocols (APs). STEP APs support roughly 40 kinds of information exchange in an effort to support the Model Data during its entire lifecycle, from concept design to final disposal. In the U.S.A. the most commonly

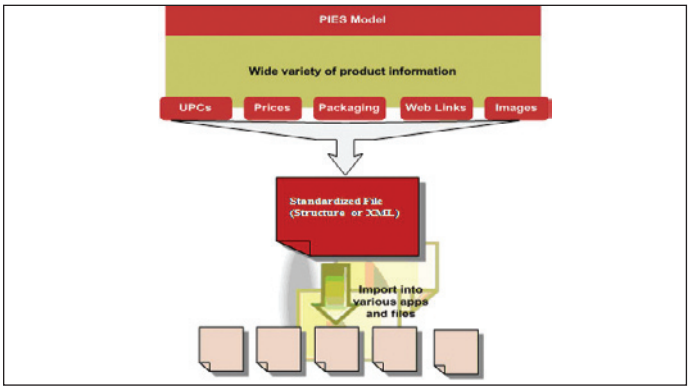


Figure 3: PIES

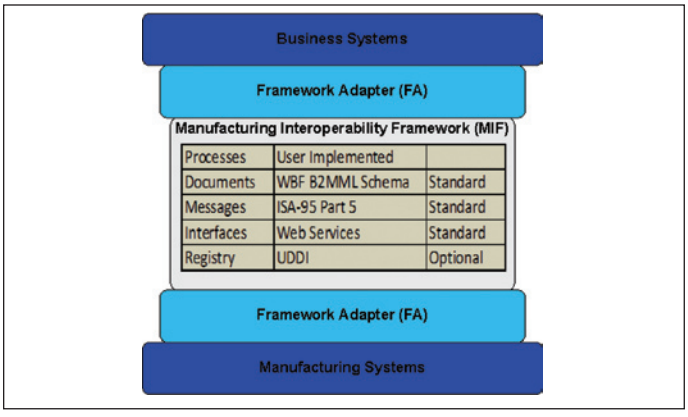


Figure 4: MIF components

implemented protocol is called AP-203 (Configuration Controlled Design). It's used to exchange data describing designs represented as solid models and assemblies of solid models. In Europe a very similar protocol called AP-214 (Core Data for Automotive Mechanical Design Processes) does the same thing.

- **ACES:** ACES is the industry standard for managing and exchanging automotive catalog and vehicle data. With ACES, suppliers can publish automotive data with standard coded designations for vehicle attributes, parts classifications, and qualifier statements. ACES provide a machine-readable format (XML) for trading partners to use in exchanging vast amounts of information. ACES is created and maintained by the Automotive Aftermarket Industry Association (AAIA).
- **PIES:** The Product Information Exchange Standard is a product of the AAIA's Electronic Commerce Committee. The purpose of this effort was to develop a standard for exchanging product information between all the members in a supply chain from the manufacturer to the retailer/wholesaler and to the service retailer. Representatives of the automotive aftermarket industry including manufacturers, retailers, wholesalers, and electronic catalog providers had a hand in developing this standard. It was designed to provide a facility to exchange data regarding products that's more comprehensive than just price sheet data. Besides price and price sheet information, the standard provides the facility to exchange information regarding such diverse elements as:
  - Packaging like dimensions and weight
  - Barcodes/product identification
  - Extended product information;
  - Warranty information
  - Shipping information
  - Links to product data sheets and safety information
  - Links to product images

Figure 3: Illustrates the type of information PIES supports.

P2B Interoperability Standards

The second most important integration effort is Plant-to-Business (P2B) interactions. These interactions are complex. To keep manufacturing producing products, there are site-autonomous requirements that we should consider. To factor these requirements into interoperability and integration, one needs to understand the different points-of-view between the business systems and the manufacturing systems. Table 1 highlights the different points-of-view between the two:

There are a number of standards that contribute to P2B interoperability such as ISA-95, WBF, and OAGi. The deliverables of these standards overlap. So manufacturers are faced with the complex decision of choosing a standard that will define their business progress, the standard's support community, and the integration elements they'll use and maintain for years. These standards are described below:

- **ISA95:** Known internationally as IEC/ISO 62264, ISA95 provides a formal model for exchanging data between business and manufacturing systems. The models define Manufacturing Operations Management, the activities on the shop floor that need production schedules and do the actual work of making products, and provide visibility into production. It has five parts:
  - *Part 1: Models and Terminology."* Also Draft International Standard ISO/IEC 62264-1. The models can be used to define the exact boundary of enterprise systems and control systems.
  - *Part 2: Object Attributes."* Part 2 determines the attributes of all the objects defined in Part 1. You can use these objects and attributes to exchange information, but they're also an excellent basis for developing relational databases.
  - *Part 3: Activity Models of Manufacturing Operations Management."* Part 3 defines production activities and information flows. Within production areas several activities are executed and a lot of information is exchanged. ISA-95 part 3 provides reference models for production activities, quality activities, maintenance activities and inventory activities
  - *Part 4: "Object Models and Attributes of Manufacturing Operations Management."* This specification defines object models that determine which information is exchanged between Manufacturing Execution System (MES) activities (which ISA-95 defines in Part 3). The models and attributes in Part 4 are the basis for the design and implementation of interface standards and ensure a flexible lapse of the cooperation and information exchange between the different MES activities.
  - *Part 5: "Business to manufacturing transactions"* is a technical specification that defines the operation between business and production automation systems that can be used with the object models in Parts 1 and 2. The operations connect and organize the production objects and activities defined in earlier parts of

the standard. Such operations take place on all levels in a business, but the focus of this spec lies on the interface between the enterprise and control systems.

- **WBF:** WBF is the Forum for Automation and Manufacturing Professionals, a non-profit that promotes the exchange of information related to the management, operation, and automation of batch process manufacturing. ISA and WBF work together to specify everything from models and terminologies to XML-based implementation elements, such as B2MML schemas, a set of XML schemas written using the World Wide Web Consortium's XML Schema language (XSD), which implements the data models in the ISA-95 standard.
- **OAGi:** The Open Applications Group is the non-profit organization that develops and maintains OAGIS, which includes both B2B and P2B elements. OAGIS standardizes the information exchanged between business and production systems. OAGi defines implementation elements without explicitly standardizing models and terminology.

Different Points of Views		
Categories	Business Systems	Manufacturing Systems
Time Horizons	Long-term view	Real-time view
Model detail	Linear route structures	Complex routes with rework paths
Control emphasis	Product cost and overall profitability	Physical movement & accountability
Modeling criteria	Accounting reference points	Material movement reference points
	Has inventory value changed significantly? If not, don't model separately	Does product stop moving? If not, don't model separately
View from	The boardroom	The work center

Table 1 Business & Manufacturing Points-of-View

The question is which standards should you use for the P2B interoperability. The answer always depends on your objectives. If you need standard models, ISA- 95 is your only choice and then you should use B2MML too. If you only want to standardize implementation elements then you should evaluate both OAGi and WBF deliverables.

Current Manufacturing Interoperability Efforts

We've seen some collaboration effort in the standards communities to resolve overlaps and consolidate the differences between the different standards in this space. For example, a Manufacturing Interoperability Framework (MIF) has been defined to organize the standards into complementary sets and identify gaps. Multiple frameworks may be organized using a scaled maturity model that can also be used to assess the interoperability of existing systems in planning systems improvements.

MIF's first task is to highlight P2B interoperability requirements. The framework consists of:

- Processes
- Documents
- Messages
- Interfaces

A high level of interoperability can only be achieved when all the elements of the framework are standardized. Figure 4 shows one possible high-level framework. In this example, there are no standard processes for P2B, leaving this for suppliers and end users to define; ISA and World Batch Forum specifications are identified for two MIF elements; and Web Services will be used to exchange the messages. This MIF indicates that a services registry is optional. Ideally, each element in the framework references a specification from a standards body. Otherwise, end users must develop specifications as part of an internal standardization processes to fill in the gaps.

Conclusion

Each approach to interoperability requires that a set of implementation standards be selected, implemented by multiple software suppliers and then supported by the manufacturing infrastructure. A single standard seldom covers all needs – from terminology to physical transport – forcing businesses, industry groups, and suppliers to choose complementary standards and creatively fill in the gaps not covered by any standard. Standards selection is the first step in creating a semantic interoperability framework. Semantic interoperability is the foundation for the SOA adaption. Without semantics, data has no meaning and is just a binary. Without semantic interoperability, service consumers and providers could easily misinterpret and corrupt data, ultimately bringing undesirable effects to a SOA and the business.

Sometimes the value of interoperability standards can be overlooked. Although it's never a single factor for manufacturing to adapt SOA to create flexible production capabilities and remain competitive globally, having interoperability standards is vital to its overall success. It's not an option, it's a necessity.

References

- Bianca Scholten. The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing.
- Guy Doumeingts, Jörg Müller, Gérard Morel, and Bruno Vallespir. Enterprise Interoperability: New Challenges and Approaches.

About the Author

Ramy Abaas is a global strategy architect focused on SOA and EII, who works for Hewlett-Packard and helps GM. He has more than 22 years of experience designing and developing IT systems for the Big Five and Big Three organizations.

ramy.abaas@hp.com

The SOA Governance Imperative

Are you ready for SOA?

BY IAN BRUCE

It's a question we're asked a lot: How do I get started with SOA?

There is uniform agreement that SOA holds great promise as a strategy for improving business agility, better aligning IT and the business, and increasing overall IT efficiency. And developing an SOA strategy has become a key issue for most large enterprises: CTOs in a 2007 McKinsey survey ranked SOA as their top strategic item.

Yet despite this, organizations struggle with how to start the transformation to SOA, especially in assessing their business readiness. For every organization that successfully adopts SOA and can testify to its benefits, there are others whose SOA initiatives have failed to deliver sustained business value, or have stalled. What does it really take to be successful with SOA? What are the technical, organizational and business challenges? What do successful SOA adopters do differently?

SOA Assessment & Domains

In the summer of 2007 we teamed with IDC to research how well-prepared organizations are for SOA and assess the critical areas that would help drive overall success. Based on the analysis of many different SOA implementations from a variety of industries, HP identified eight primary domains that together provide a framework for measuring SOA maturity and readiness. These domains are:

- **The Business Domain** – Ensuring business and IT commitment and involvement in SOA.
- **The People Domain** – Ensuring that the right mix of skilled staff understand and are committed to SOA. Fundamentally, SOA requires a change in the way IT people work together and the way they work with the business.
- **The Program Management Domain** – SOA requires an iterative approach, with each step providing a complete business solution. The key is ensuring this is program management that involves

the right organizational span across relevant teams, departments, business units, and partners.

- **The Governance Domain** – This concerns the models, systems, and processes that manage services across the lifecycle. Typically SOA governance is much more important in an SOA that in a traditional IT environment.

- **The Architecture Domain** – The "A" in SOA is a reminder that enterprise architecture, solutions architecture, information architecture, and technology architecture are all critical.

- **The Enabling Technologies Domain** – Of all the SOA domains, enabling technologies has gotten the greatest attention and so is probably best described. This area covers the tools and technologies needed to support the goals and processes of enterprise SOA.

- **The Operations and Management Domain** – This covers all aspects of SOA operations and management, and the application of the processes and policies defined in the governance domain.

- **The Supply and Demand Domain** – As SOA implementations grow and mature, they open up a variety of new opportunities for sourcing. By having resources and assets described as modular services, it becomes possible to be more dynamic and granular in how these services are realized.

Being a master of all these domains ensures that you have the capabilities and assets in place to effectively adopt and operate SOA. Conversely, understanding deficits in an organization's domain model offers a good indication of where to devote resources that will drive successful SOA transformations.

The domain model was the basis for constructing a detailed online questionnaire designed to assess a respondent's overall readiness for SOA in each of the eight areas. The high-level assessment identified the capabilities and assets that have to be developed or acquired, and the actions that need to be taken. The assessment tool can suggest where an organizations needs to concentrate efforts to start their SOA transformation, or point to deficits in an existing SOA strategy. The assessment tool acknowledges that SOA doesn't happen overnight: SOA adoption is and should be incremental, and ideally should provide measurable and cumulative benefits at each stage of adoption.

The Governance Imperative

In the first six months after the online assessment tool was made available it attracted over 200 responses from different companies worldwide. Respondents varied widely in size, SOA maturity, and business. About 40% were based in North America, 25% in Europe, 20% in Asia/Japan, and 15% South America, Africa, or the Middle East. No industry sector accounted for more that 10% of respondents.

Despite the wide variety of respondents, one issue became clear: Governance is the single, most consistent area that needs to be significantly enhanced in most organizations. It was also cited by companies that have mature SOA implementations as the area that provided a foundation for SOA success, and the domain that had surprised them by being so fundamental and so underrepresented in their organization.

As mentioned earlier, SOA governance is typically much more important than governance in traditional IT environments because when organizations adopt SOA they are typically moving away from an architecture consisting of a relatively small number of large, static, and well-bounded applications and systems into an environment of myriad smaller, easily accessible business services. The very benefits of SOA, such as service reuse and enhanced business flexibility, means that the relationships between these smaller business services and their impact on the business are more complicated to grasp and control. Done wrong, SOA can quickly become chaotic.

SOA introduces many independent and self-contained moving parts — components that are widely reused across the enterprise and are a vital part of mission-critical business processes. What happens when a service is changed? How can you be sure the service you're consuming is of high quality? What happens if a subcomponent of a composite service is retired? How can you be sure a new service is compliant with IT, business, and regulatory policies? How can you ensure a service's predictable uptime? These questions illustrate the need for SOA governance. It's about managing the quality, consistency, predictability, change, and interdependencies of services. It's about blending the flexibility of service orientation with the control of traditional IT architectures.

A significant challenge to widespread SOA adoption is that while the management of service quality is paramount, information about the quality must also be effectively communicated and measured. For the first time, quality must be proven and demonstrable to consumers to gain their trust and create an effective shared-service environment.

A useful way to think about the importance of trust in SOA is to consider the example of a consumer marketplace such as eBay, where anonymous buyers and sellers are expected to come together and quickly establish some degree of trust, despite their total anonymity. According to basic economics, a market requires information to function efficiently. Information is the lifeblood of any market, largely because it enables buyers and sellers to make informed decisions, and provides the basis for establishing trust. Buyers and sellers on eBay trade on the basis of information. Buyers won't do business unless they understand what's being offered, the terms and conditions of the sale, and the reputation of the seller; likewise, sellers want some assurance of the buyer's ability and willingness to pay in a timely fashion. In this respect, SOA is no different. SOA can't be successful without trust — consumers will simply fail to reuse services if they can't be assured of the quality, predictability, and transparency of the terms and conditions. In the same fashion, organizations shouldn't encourage the use of services without understanding and controlling access, provisioning, and the overall fitness of reusable services.

Tightly coupled systems define governance and control in the context of the application. SOA is different in the sense that the application context is varied and ever changing. This means that governance must be managed at a different level of abstraction — on the services themselves. Policies need to be taken out of the code and externalized as metadata associated with them. Complicating matters is the fact that, in a loosely coupled world, change is a constant and increasing. Loosely coupled architectures potentially involve hun-

dreds of services that evolve and change based on their own unique lifecycles. With all of this change happening at once, how can an IT organization identify and manage the potential impact and interdependencies of change? This is a key domain of SOA governance.

The challenge facing enterprise architects tasked with developing SOA governance processes is creating an open marketplace for services that's regulated enough to protect the business, but not so much that over-regulation causes adoption failure. Services need to be easily accessible and demonstrably trustworthy, but how and when services are reused or changed needs to be carefully controlled.

The Criteria for Comprehensive SOA Governance

It's a mistake for organizations to discount governance as something that's optional, nice to have, or a later-phase aspect of SOA. Governance must begin with the initial SOA deployment, providing the framework, processes, and practices for scaling out a healthy and efficient SOA. An organization can't simply back into governance down the road once an SOA implementation has reached a new level of maturity. In the context of SOA, governance doesn't follow success; governance begets success.

SOA governance must focus on establishing a framework for assuring service quality and engendering trust between service providers and consumers as both individual services — and the service network as a whole — progress through their lifecycles. Without strategies or infrastructure for governance in place organizations will hit roadblocks as they try to advance their SOA initiatives.

We've identified three broad criteria for comprehensive SOA governance:

Engaging the Organization

Governance needs to have teeth, and needs organizational buy-in. Creating a SOA Center of Excellence is practical way to integrate a program for SOA governance, quality, and management with lifecycle services to plan, assess, implement, and manage your SOA initiative. It unites key business and IT stakeholders in a decision-making body and provides a mechanism for instilling consistency and control. It provides methodologies, expertise, and a common set of tools to support a standardized SOA infrastructure, encourage and enforce new development methods and operational procedures, and monitor SOA adoption, service utilization, and overall business outcomes.

Once established, the SOA Center of Excellence assumes full responsibility for supporting SOA initiatives — from strategic planning to SOA infrastructure and operations — as you:

- Standardize the SOA infrastructure, development methods and operational procedures
- Develop a reference architecture, service characteristics, patterns, and SOA blueprint
- Leverage new and existing SOA expertise across the organization
- Deliver an SOA roadmap for management and technical tracks
- Stay current with the latest technologies and techniques
- Establish an SOA program and create metrics, dashboards, and checklists to measure efficiency
- Promote high-quality, cost-effective timely services by applying consistent SOA methodologies and techniques
- Apply enabling technologies to enable SOA governance, quality and management

Best Practices Automation

SOA is about embracing heterogeneity, leveraging resources, and exploiting existing IT assets. It's not about rip-and-replace. It doesn't require that everything be done one way using one set of technologies. However, it does require a robust foundation to ensure visibility, trust, and control for SOA governance and to automate and manage governance processes.

At a minimum, SOA governance requires creating a system-of-record that provides a canonical source for all information about services. It requires a way to manage the full lifecycle of services, and engage service providers and consumers so they can contract and agree on how services will be reused. It requires a means of managing business and IT policies, associating them with the right services, and enforcing them at runtime. Governance technologies will necessarily be used across the organization by developers creating services, enterprise architects managing processes, and by business consumers of services: they must provide meaningful information and insight to each of these different stakeholders.

Support for the Extended Enterprise

One of the promised benefits of SOA is its ability to bridge technological and business process divides that separate business units, partners, the supply chain, and customers – the "extended enterprise." For this to be realized, SOA governance must be middleware and platform-independent to support heterogeneous IT. This requires an open, standards-based approach to the way governance solutions share information and interoperate with other SOA technologies. SOA governance can't be tied to a single vendor's "stack." Standards and specifications that support an ecosystem of products and supported technologies make SOA governance easier to implement, integrate, and extend as adoption accelerates.

As organizations assess how to begin their SOA transformations, a compelling place to concentrate is the governance domain. By establishing a governance foundation that engages with the organization, provides best practices automation, and offers support for the extended enterprise, gaining measurable business outcomes from SOA initiatives is much more likely. ■

"Classical SOA and ESB technologies don't support large collaborative business networks"

About the Author

Ian Bruce is director of SOA marketing at HP Software, responsible for the worldwide marketing for HP's SOA products. Prior to HP, he was director of marketing for Systinet, a pioneer in the SOA governance and Web Services that was acquired by Mercury Interactive. Ian was head of marketing for the financial services software company CWB (acquired by Thales), and head of communications for CSC in Europe. He has a BS in engineering from Coventry University and a PhD in communication from the Newhouse School at Syracuse University.

Visit the *New* www.SYS-CON.com Website Today!

The World's Leading *i*-Technology News and Information Source 24/7

FREE NEWSLETTERS
Stay ahead of the i-Technology curve with
E-mail updates on what's happening in your industry

SYS-CON.TV
Watch video of breaking news, interviews with industry leaders, and how-to tutorials

BLOG-N-PLAY!
Read web logs from the movers and shakers or create your own blog to be read by millions

WEBCAST
Streaming video on today's i-Technology news, events, and webinars

EDUCATION
The world's leading online i-Technology university

RESEARCH
i-Technology data "and" analysis for business decision-makers

MAGAZINES
View the current issue and past archives of your favorite i-Technology journal

INTERNATIONAL SITES
Get all the news and information happening in other countries worldwide

JUMP TO THE LEADING i-TECHNOLOGY WEBSITES:

IT Solutions Guide
Information Storage+Security Journal
JDJ
Web Services Journal
.NET Developer's Journal
LinuxWorld Magazine
Linux Business News
Eclipse Developer's Journal
MX Developer's Journal
ColdFusion Developer's Journal
XML Journal
Wireless Business & Technology
Symbian Developer's Journal
WebSphere Journal
WLDJ
PowerBuilder Developer's Journal

Why Enterprise Architects Continue to Fall Short with SOA...and Enterprise Architecture

Why your old data center infrastructure won't scale in the SOA age

BY DAVID LINTHICUM

If you read this column and listen to my podcasts, you know that I call SOA what SOA is... an architectural pattern. In many instances, SOA is a vital component of healthy enterprise architecture. Indeed, I've provided some keynote talks around this very topic at about half-a-dozen enterprise architecture conferences to date. However, generally speaking, the enterprise architects out there still don't "get" SOA, and they continue to do a poor-to-average job of creating enterprise architectures that...well...support their enterprise.

By the way, those of you who will respond to this column and tell me that you're doing well with your architecture, that's fine. Unfortunately, dozens of you can be an exception to my observations, and it still doesn't mean I'm wrong here. This is a systemic problem; however, there are clearly islands of success out there.

What's core to the issue is that many enterprise architects don't have the political will, or the authority, to solve many of the core problems. It's very difficult to make changes in enterprises today. There is a certain amount of job risk that comes with those types of actions, risks that many enterprise architects are unwilling, or unable, to take.

Here are the issues, the way I see it:

EAs don't understand SOA. The largest issue is that the majority of enterprise architects (EAs) do not understand what SOA is and what SOA is not. Either they just don't bother, or they want the definition of SOA to fit some preconceived and incorrect notion in their minds. Are you listening...you guys who use the terms "SOA" and "ESB" interchangeably?

EAs don't understand their own issues. The other problem is that many enterprise architects can't tell you the cost of inefficiencies within their existing IT infrastructure and enterprise architecture, any value they would get from reuse, and any metrics around the value of agility within the enterprise. In some instances there is no central record/artifacts around data semantics, APIs, processes, workflow, etc. If you don't have a clear understanding of what the current issues are, you cannot know how best to correct them over time.

EAs fear change. If things are bad, then change is typically good. Unfortunately, change also means risk, and risk is something people typically don't like. The fact is that people are rewarded for maintaining more than they are rewarded for improving. This answers the question about why so many of the enterprise architectures out there are now layers upon layers of tactical one-off solutions designed to "keep things going a few more years." Somebody needs to have the political will to figure out a long-term solution using sound enterprise architectural approaches, including SOA.

Those enterprises that have clear architectural issues typically don't understand how to approach fixing the problems. Indeed, it's often overwhelming to most architects, and thus – like anything else that's overwhelming – it's easier to ignore the core issues and go about your daily routine.

If you have a problem, you know the symptoms. It takes two months or more to change any major business process. You have no holistic understanding of your data, your services, or your business processes. You have redundant and dysfunctional data, without any consistent integration strategy, nor common interfaces into the systems. You get the idea.

Unfortunately, I'm not sure that guys like me shining a light on these shortcomings will have much impact. I think it's going to take some well-published disasters that almost kill a company or two before the powers that be understand the real problem here. Hopefully, a few of you will be more proactive.

About the Author

Dave Linthicum is the CEO of StrikeIron (www.strikeiron.com), which offers Web services on-demand. In additional Dave is the author or co-author of 10 books, a thought leader in the Web 2.0 and SOA space, frequent keynote presenter, and has served as the CTO for 3 technology companies.

david.linthicum@strikeiron.com

Cost Effective SOA Management

small budget

Big Questions

mission-critical services

major Pressure

short timeline

reduced staff

A flexible Web Services Management solution for the enterprises that need the following capabilities

- Runtime Governance, Security & Policy Enforcement
- Visibility, Management and monitoring of the SOA environment
- Agentless visibility into SSL and non SSL Web service traffic
- Multiple Deployment options to fit Your environment
- Cost effective

Download a free trial at www.managedmethods.com

Managed Methods

www.managedmethods.com

DataServices WORLD

Welcome to DataServices World at SOAWorld 2008!

June 24, 2008 New York City

Ken North
Chair, DataServices World at SOA World '08

DataServices World is about the confluence of databases, data warehousing, business intelligence, enterprise computing and Internet computing. Its focus is architectures and technologies for accessing data from heterogenous data sources and providing that data to consumers such as components, services and applications.

Distributed processing, high-speed networks, powerful servers, components and collaboration define the landscape of 21st century computing. For building new systems and exploiting legacy applications and data, developers are looking to collaborative applications assembled from distributed components. The emergence of XML and web services spurred an increase in services-oriented architecture (SOA) adoption. SOA today can include web services, grid services, integration services, semantic web services, components and messaging systems.

Developers who've enjoyed success with component-based development are looking to new architectures with services as the new components. But even as the Service Component Architecture and other new technology gain tractions, some system characteristics remain consistent. Today's systems, like their predecessors, typically have a requirement for persistent information and databases.

Many organizations have a variety of persistent data stores, including SQL databases, geo-coded data files, spreadsheets, content management systems and XML. Services, applications, and mashups can consume and integrate data from disparate data sources. In an n-tier enterprise architecture and a service-oriented architecture, the logic for providing data from databases and other data sources resides in data access layers and data services layer.

Today's data services layers encapsulate logic for accessing data stores, typically using standards-based technology such as ODBC, JDBC, ADO.NET and Service Data Objects. These specifications define solutions for uniformly accessing and manipulating data from heterogeneous data sources.

At DataServices World, we'll uncover architecture and technology solutions for accessing, integrating and processing data from multiple sources while guaranteeing security and scalability. These solutions include robust, high-performance data access middleware, optimized databases, efficient protocol handling, tuned queries and state of the art data services. We'll be looking at technology of interest to CTOs, enterprise architects, system architects, information architects, developers, database gurus, consultants and analysts.

DataDirect® TECHNOLOGIES

2008 Diamond Sponsor

REGISTER TODAY AND SAVE

www.dataservicesworld.sys-con.com