

BUILDING SOA YOUR WAY

A fault line runs beneath the groundswell that began a few years ago with XML Web services and continues today as SOA (service-oriented architecture). True, nearly everyone agrees that XML messaging is the right way to implement low-level, platform-agnostic services that can be composed into higher-level services that support enterprises business functions. Yet, here's also a sense that the standards process has run amok.

IBM, Microsoft, and others have proposed so many Web services standards that a new collective noun had to be invented: WS-* (pronounced "WS star" or sometimes "WS splat"). The asterisk is a wild card that can stand for Addressing, Eventing, Policy, Routing, Reliability, ReliableMessaging, SecureConversation, Security, Transactions, Trust, and a frighteningly long list of other terms. Surveying this landscape, XML co-creator Tim Bray pro-

Every enterprise needs to find its own balance between a complete, scalable architecture and simply building an SOA that works

nounced the WS-* stack "bloated, opaque, and insanely complex."

It wasn't always so. Simple forms of XML messaging were succeeding in the field long before any of these standards emerged. At *InfoWorld's* SOA Executive Forum in May, Metratech CTO Jim Culbert described how his company's service-oriented billing system worked back in the late 1990s. The messages exchanged among partners were modeled in XML and transported using HTTP with SSL encryption — the method still used for most secure Web services communication today. Seybold analyst Brenda Michelson, who was then chief architect at L.L. Bean, tells a similar story about that company's early experience with Web services.

Two factors were prominent at the time. First, the Web offered a simple, pervasive integration framework, one later promoted to the status of architecture and assigned the label REST (Representational State Transfer). Second, XML provided a universal way to define services in terms of the data they produced or consumed, rather than in terms of the code that produced or consumed the data. In combination, these factors were — and still are — powerful enablers.

Cranking Up Complexity

How, then, did we arrive at WS-*, which Culbert and others say is a cart that's gotten way ahead of its horse? One theory holds that the heavy-hitting vendors, working closely with key customers and partners, have ratcheted complexity up to a level that only they will be able to sustain. Because those specs are so far ahead of what most users need today, their development hasn't been an organic process driven by well-known requirements.

Patrick Gannon, president and CEO of OASIS, the

BY JON UDELL | ILLUSTRATION BY MICHAEL MORGENSTERN

“If you need to describe your interfaces precisely, why wouldn’t you use WSDL?”

— Frank Crossman, Mindreef

standards body now coordinating a number of the WS-* specifications, reluctantly agrees that users should have been more engaged from the beginning. “I wasn’t involved in creating those specs without formal user requirements on the table,” he says. “But I’m a pragmatist; the specs are there.”

Another view holds that industry heavyweights, who have paid their dues when it comes to security, transactions, and reliable messaging, are indeed qualified to translate their experience in these matters into the language of XML. TN Subramaniam, director of technology at RouteOne, which makes software that streamlines credit management applications on behalf of car dealers, learned that lesson the hard way. At one point he began drafting his own spec for single sign-on, only to abandon it when he discovered SAML, which his joint-venture partners enthusiastically adopted because all their identity management vendors — including Netegrity and Oblix — were supporting it.

“What are the chances,” Subramaniam asks, “that five architects meeting every other day will iron out all the possibilities, versus having a committee thinking it all through in great detail with all the vendors on board?”

It’s tempting to interpret the tension between these two perspectives as a replay of the cathedral and the bazaar — or perhaps instead, WS-Heavy and WS-Lite. In that dichotomy, WS-Heavy would refer to the security, reliability, and scalability that WS-* claims to deliver, whereas WS-Lite would mean the speed, simplicity, and agility that attract labels such as REST, AJAX, and RSS. None of the enterprise architects we interviewed for this story has pledged allegiance to either of these camps, though. They’re intensely pragmatic people who will do whatever it takes to

get the job done, and it’s instructive to learn how they are — and are not — making use of Web services standards.

RouteOne: Securing Credit Checks

Although end-to-end SSL is often sufficient, RouteOne’s Subramaniam has two reasons to prefer the more granular approach enabled by WS-Security. First, it’s necessary to digitally sign the credit applications his application transmits, and to do so according to rules understood by service partners. WS-Security defines such rules, although admittedly, and unfortunately, too many of them. One method is to put the signed application into the body of the SOAP message; another is to use SOAP with attachments. In the end, there was no agreement among the service partners, so RouteOne uses both. That’s frustrating, but Subramaniam would rather have two rules than none.

The second reason touches on one of the deep principles that motivates the design of the WS-* stack: pervasive intermediation. RouteOne is required to maintain meticulous audit logs and would prefer not to have to encrypt all of them. So it’s using DataPower’s XML router/accelerator to selectively encrypt only sensitive items such as gross pay and Social Security number. Because it’s a standards-based intermediary, the DataPower box can straightforwardly modify RouteOne’s XML message traffic in this way, and it could be swapped out for another appliance that did the same thing.

When services communicate directly, as many if not most still do, there’s no need to define the rules of engagement that enable service intermediation. Today’s most visible exemplars of WS-Lite — Amazon and eBay — use Web services in a point-to-point way. In that mode there’s not much difference

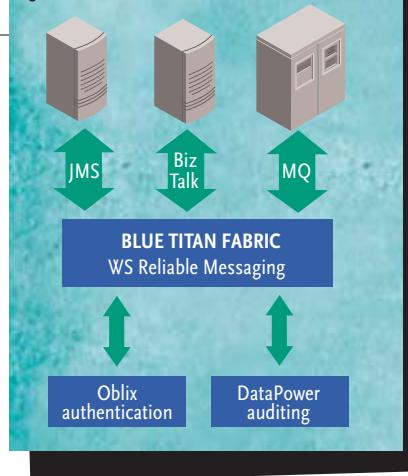
between SOAP/WSDL APIs and REST APIs, so it’s not surprising that developers who work with these platforms overwhelmingly prefer the REST flavor. But when you do need to flow your XML traffic through intermediaries, SOAP and WSDL suddenly make a lot more sense.

Subramaniam is a pragmatist, however. Plain XML over HTTP, sans WSDL, also plays a role in RouteOne’s internal and external affairs. Because it’s a no-brainer to put a servlet interface onto an internal legacy system and pull XML data through it, that strategy is used where appropriate. Some of RouteOne’s external partners use the same approach, and because “they’re making money hand over fist” doing so, Subramaniam can’t mandate otherwise. Instead, RouteOne normalizes inbound traffic to SOAP and WSDL in order to enable its expected future use of BPEL (Business Process Execution Language) for service orchestration. Today, partners who don’t present SOAP and WSDL interfaces are not competitively disadvantaged. But the tipping point may not be far off.

RouteOne depends on both SAML and WS-Security, and Subramaniam wishes he could use a standard form of reliable messaging, too. “If I don’t send a message, we are losing money,” he says. Drawing inspiration from ebXML (e-business XML) and JMS (Java Message Service), he specified — and is now using with partners — a scheme that guarantees orderly and reliable delivery of messages. But he’d rather it were otherwise and hopes that OASIS will succeed in merging the two proposals it is now hosting: WS-Reliability and WS-ReliableMessaging. This duplication is “really, really bad,” Subramaniam says. “I wish we had a common spec so I could dump my stuff and just use it.”

The Pfizer Fabric

Identity-based security and reliable messaging form the warp and woof of this pharma giant's SOA.



Corillian: Point-to-Point Simplicity

Many service-oriented systems don't require reliable messaging and, according to Scott Hanselman, chief architect at Corillian, his company's banking middleware falls into that category. Corillian's product, called Voyager, handles services touched indirectly by 25 percent of all users of online banking, Hanselman says. "But the only transaction they care about is the one at the host." So he's not worried about the merger of WS-Reliability and WS-ReliableMessaging. Although he does make use of WS-Security, he regards SSL as equally effective in most cases. That approach precludes routers and intermediaries, he admits, "but rarely do I use them, because nine times out of 10 we're doing point-to-point messaging."

He's also dismissive of UDDI, the much-maligned standard for publishing directories of Web services. What about the argument that services not found in the yellow pages won't be reused? Hanselman doesn't buy it. Finding services isn't really a problem for developers, he says. Using them easily and effectively is. Imagining a fictional average developer named Mort, Hanselman opines that SOA will be a nonstarter until we can shield Mort from XML angle brackets and X.509 certificates. To that end, he thinks the most important standard is WSDL because it's a tool-enabler.

Of course WSDL has earned its fair share of criticism, too. RouteOne's Subramaniam thinks that the "goofy" complexity of WSDL 1.1 made it a ball and chain that SOA has had to drag around, and he hopes that the "much cleaner" WSDL 2.0 will lighten the load. Perhaps, Hanselman says, but "you can't unring the bell." Millions of Web services transactions ride on WSDL 1.1 and will for a long time to come. Using WSDL

1.1, Corillian was able to describe the objects, messages, and services at the core of Voyager and to bind those descriptions to internal machinery that doesn't speak XML. As the need arose, the company created alternate bindings that enable customers to see the engine through a Web services lens. If WSDL 1.1 was an 80 percent solution, Hanselman thinks, then WSDL 2.0 might be a 90 percent solution, but either can deliver crucial leverage.

Ohio State: Securing Vital Signs

The most widely adopted of the advanced Web services standards is clearly WS-Security. Beyond that it's hard to find practitioners who have worked with the more exotic beasts in the WS menagerie, but Furrukh Khan — who holds joint appointments in the colleges of engineering and medicine at the Ohio State University Medical Center and is broadly responsible for its medical IT — tells a fascinating story about his transition from basic to advanced Web services.

In this scenario, vital signs flowing from monitors are recorded in databases and are simultaneously delivered to smart clients that observe, replay, analyze, and annotate the streams of

data. The streams must be delivered to a lot of clients reliably, securely, and in near real time.

A first implementation, based on Microsoft's WSE (Web Services Extensions), made use of WS-Policy, which hasn't yet found a home in a standards body but likely will soon. WS-Policy was used to declare the means of authentication to back-end databases — for example, to require X.509 certificates signed by a specified key — as well as the required payload signature and encryption.

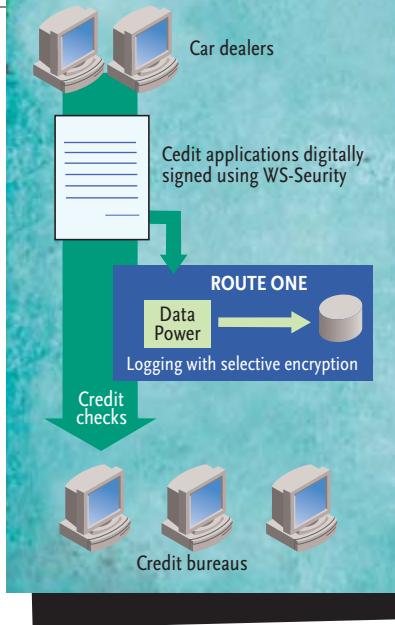
The current implementation — based on the beta version of Microsoft's Indigo, a Windows implementation of a stack of advanced Web service protocols — uses WS-ReliableMessaging to ensure orderly and reliable delivery of messages. And it uses WS-SecureConversation to optimize that secure, reliable channel for high-volume traffic.

Khan explains that WS-Security alone, in concert with WS-Policy, could not sustain near-real-time traffic. The protocol, which required frequent exchanges of credentials with the identity management system, was too chatty. WS-SecureConversation, which enables caching of credentials, streamlines the protocol. That, coupled with a feature of Indigo's implementation of WS-ReliableMessaging that enables a router to broker a connection between two end points and then get out of the way, resulted in a massive scale-up.

"Before, with WSE, each router limited us to 300 clients," Khan says. Indigo can support 638 clients per router, he adds, and with optimization, that many clients for each service running behind the router. "So if you keep on adding services, it scales linearly," he says. The system currently supports more than 1,000 clients, all observing vital signs simultaneously every 30 seconds.

RouteOne's Connections

This clearinghouse for car buyer credit checks uses WS-Security and SAML to secure consumer data.



Reflecting on the transition from WSE to Indigo, Khan echoes Scott Hanselman's point about shielding developers from XML. WSE handled the basic scenarios, he says, but beyond those, "we had to go into the schema and do all the angle brackets." Thanks to Indigo's higher level of abstraction, that problem vanished.

More broadly, Indigo made a harder problem — the appropriate use of Web services in concert with platform-native services and transports — tractable. "Behind each Web service there's an MSMQ [Microsoft Message Queue] and an enterprise service," Khan says. "In the Microsoft domain, enterprise services are completely different from Web services, MSMQ lives in its own world, and XML has its own toolset." Different team members had to be experts in different disciplines; no one person could master them all. From Khan's perspective, Indigo gives "Mort" the leverage he needs.

Providence: Enforcing Contracts

Providence Health Systems deploys what's becoming a typical two-tiered SOA to support its clinical and business applications and its physician and patient portals. A set of coarse-grained services, which map closely to business processes, are woven from another set of more elemental services. Although some advanced standards are in use, such as WS-Security, Providence doesn't deal with them directly. "We rely on our vendor's implementation of the security stuff," says Mike Reagin, vice president of development at Providence. The vendor in this case is Infravio, whose Web services management system provides the framework within which Providence deploys and manages its services.

Infravio implements UDDI, but Reagin says that, with relatively few services in play, directory lookup isn't a big deal. Declaring and enforcing policies that control the use of those services, however, is a very big deal, as is monitoring service activity.

In Infravio's model, services are provisioned as producer/consumer pairs, each of which is governed by a contract. The master patient index, for example, is a common service used by both the physician and patient portals but in slightly different ways. The patient's health-plan member number, which appears in the patient portal, must be stripped from the physician portal. By creating separate WSDL interfaces for separate consumers, Infravio enables the common service to be reused rather than duplicated. This variation is achieved in a declarative way, rather than by writing code.

Providence's SOA deployment is, for now, largely internal. Services feed its outward-facing portals but are not yet directly exposed to partners. That day

will come, Reagin feels sure, and when it does, he expects that his use of the core standards, SOAP and WSDL, will enable more advanced scenarios: orchestration, reliable messaging, policy-governed security, and auditing. Which pieces of the WS-* stack will enable those scenarios? Reagin doesn't lose sleep over the question. When the time comes, he'll buy — rather than build — the needed infrastructure.

Pfizer: Trusting the Fabric

Security and reliable messaging are key requirements for the Pfizer Global Pharmaceuticals (PGP) group. The pharma giant's SOA deployment meets those requirements with the help of Blue Titan's Network Director, which manages PGP's Web services traffic across the enterprise.

On the security front, Blue Titan's "fabric" enforces a policy that routes requests through a DataPower intermediary for compliance auditing and through an Oblix system for authentication. Martin Brodbeck, PGP's application architecture director, sees WS-Security as the integration framework for these activities. Although he doesn't deal directly with related standards, such as WS-Policy or WS-Trust, Blue Titan does in fact support them.

It's worth noting that a number of standards said to be "vendor-driven" are primarily of interest to vendors. For example, another architect interviewed for this story was hands-on with WS-Security but unaware that WS-Trust plays a role in his implementation. Why? The WS-Trust protocol is spoken only between his security broker, VordelDirector, and his identity provider, Entrust. The messages exchanged between his company and its Web services partner have nothing to do with WS-Trust, says Mark O'Neill,

“If I don’t send a message, we are losing money.”

— TN Subramaniam, RouteOne

CTO of Vordel. “We and Entrust chose to use it because it’s a spec that we don’t have to work out ourselves,” he says. The WS-Security protocol used by the service end points and the WS-Trust protocol used by infrastructure components are “solving completely different problems — it just so happens that both involve specs that begin with WS.”

Along with security, reliable messaging is a key PGP concern. With various flavors of message-oriented middleware in play, along with multiple versions of some of these (such as JMS), the company values the Network Director RM’s capability of hiding the differences. Although that product’s support for WS-ReliableMessaging is not immediately relevant, PGP is evaluating Indigo, which natively supports the standard. “Blue Titan in concert with Indigo will make RM [reliable messaging] really, really easy to do,” Brodbeck says.

To the short list of important standards such as WS-Security and WS-ReliableMessaging, Brodbeck adds RSS, the wildly popular format for Weblog syndication. That PGP would regard this variant of WS-Lite as strategic may surprise you, but if you think about how collaboration and knowledge management drive the top line in an organization such as Pfizer, it shouldn’t. What PGP envisions, however, is not your garden-variety blogging software. “We have to recontextualize RSS for the enterprise,” says Richard Lynn, PGP’s vice president of global applications and architecture.

PGP’s requirements include virtualizing RSS feeds so that they’re independent of hard-coded addresses, aggregating them for specific business functions and securing them using the same kinds of declarative policies that govern existing Web services. According to Frank Martinez, founder and

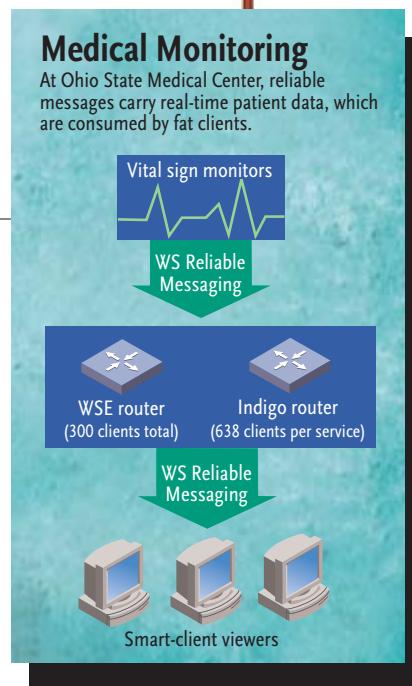
CEO of Blue Titan, a forthcoming release of Network Director will address these requirements, building on the product’s capability of wrapping WS-Heavy infrastructure around WS-Lite protocols.

Heavy, Lite, or Just Right?

When you regard the WS-* stack as a whole, you have to conclude that the critics are right: It really is a monster. Taming it will require, in part, a unifying conceptual framework. That’s a point that Gannon, Khan, and Subramaniam each make in different ways. Gannon points to a series of blueprints and reference models published by OASIS. These documents aim to help architects understand how the various WS-* specs, which are designed as modular building blocks, combine to solve specific problems. For Ohio State’s Khan, it’s not just about blueprints. He needs a toolkit that tames the complexity and thinks Indigo will be that toolkit.

RouteOne’s Subramaniam hopes that a recent initiative called JBI (Java Business Integration) will be a unifying force in the Java world. What’s hard about Web services, he says, “is that you have to see the whole picture — WSDL, and then SOAP, and relevant parts of WS-Security, and BPEL.” He’s anxious for vendors such as SeeBeyond, which was recently bought by Sun Microsystems, and webMethods to embrace JBI. “When you can see how it all fits together in the big picture of JBI, a very nice infrastructure emerges,” he says.

Of course, toolkits and frameworks are double-edged swords. Even when wire protocols are standard and open, you can get locked in to proprietary abstractions layered on top of those protocols. That’s why pragmatic architects and developers who don’t yet need advanced WS-* features tend to focus



on the basics: SOAP and WSDL. “If you need some kind of envelope, why wouldn’t you use SOAP?” Subramaniam asks. “And if you need to describe your interfaces precisely, why wouldn’t you use WSDL?” Frank Grossman, co-founder of Mindreef, says that most of the customers who use his company’s SOAPscope diagnostic suite have adopted this strategy, which he adroitly labels “WS-JustRight.”

For Grossman and others, WS-JustRight means using SOAP and WSDL to strike a balance between formal contracts and agile interoperability, while laying a foundation for future use of more advanced SOA features. PGP’s Brodbeck agrees that WSDL is the key enabler of reusable business transactions and processes. He also extends the definition of WS-JustRight, however, to include enterprise-enabled RSS as the key enabler of reusable content.

For many practitioners, WS-JustRight now includes aspects of WS-Security, too. For a few, it includes reliable messaging, transactions, routing, and policies related to these features. The definition will evolve over time, but the only one that really matters now is the one that’s just right for you. ☛

SPRINT MAKES THE SERVICES CONNECTION

The telco's component-based SOA unites customer, partner, and departmental apps

BY GALEN GRUMAN

AS FAR BACK AS FOUR YEARS AGO, Sprint's IT staff was already headed toward SOA (service-oriented architecture). They just didn't know it yet.

When developers first began exposing Sprint's back-end systems as reusable components, the concept of Web services was still largely unproven. Previously, they had connected the company's departments, customers, and partners to its mainframe systems through a series of stand-alone Web sites and b-to-b interfaces. For one of the U.S.'s largest telecom providers, with a customer base of more than 10,000 companies, such a one-off approach wouldn't be sustainable for long, however.

Four years ago, the local phone-service business unit began developing Java applications for such basic functions as log-in and password reset, as well as customer tasks such as service ordering and account updates. Seeing the wisdom in this modular approach, other business units quickly followed suit. Unfortunately,

“Often, there was no incentive for reuse of core infrastructure assets or components.”

— Edmund Vazquez, Sprint Business Services

multiple, parallel Web-services development efforts resulted. Sprint developers were creating the same modules over and over again, often without knowing it, wasting development time and dollars.

“We had lots of different platforms and technologies as each design progressed for its line of business and customer requirements,” recalls Edmund Vazquez, Web services program manager for the enterprise-focused Sprint Business Services (SBS) unit. “Often, core functionality was projectized. There was no incentive for reuse of core infrastructure assets or components.”

To rationalize its Web services deployments, SBS decided to create a common architecture to provide a framework for identifying, developing, and managing its services. A solid SOA underpinning would let SBS break services down to reusable components that would reduce overall development efforts, although it would also require a clear understanding of business needs and the services required to meet them.

To manage the SOA and the services developed on it, SBS created two distinct IT groups. One focuses on the overall architecture and strategy, while the other concerns itself with service development and integration. This ensures that the architecture is maintained and applied consistently while still allowing development teams to focus on their specific business-logic areas.

Of course, the transition didn't happen overnight. According to Vazquez, it didn't have to. “One of the nice things about SOA adoption is that adoption,

implementation, and deployments can be incremental as long as you keep your eye on the bigger picture,” he says.

Building for Manageability

Architecturally, SBS defines three kinds of services: atomic, aggregate, and composite. Atomic services might expose a single API and are usually transactional in nature. Aggregate services may involve calling sequences of atomic services, much like one Java class calling other classes. Composite services, on the other hand, require orchestration or choreography.

A composite Web service implements a process or workflow involving multiple atomic or aggregate Web ser-

down the functions into their basic components. Doing so lets developers create the most reusable components, enabling them to then build up aggregate and composite components easily when it's clear that entire sequences of services are reusable.

By comparison, Vazquez recalls some early Web services that were so specific to their customers that no one else could use them. “They're technically Web services ... but they're really just applications,” he says.

For messaging among services, SBS relies on a WSM (Web services manager) based on the X-broker platform from Infravio, which handles the atomic and aggregate services and also provides a Web services registry (see “Services Registry Aids IT, Drives Sales,” page 34).

SBS encapsulates its mainframe applications as “virtual services” within EJB wrappers that expose the functionality of the applications using SOAP, WSDL, and XML schema. This presentation meets two needs: It keeps the EJBs' internal business logic private while also offering trading partners a standards-based approach to consuming the service, Vazquez says.

The system supports several data exchange standards, since SBS service providers and customers use a wide range of technologies. Flat-file text and basic XML are the most common choices.

“We are leveraging specific XML standards for specific types of domains or transactions, such as TML, eTom, ngOSS, and others developed by telecom industry consortia,” Vazquez says. “The challenge here, as in all stan-

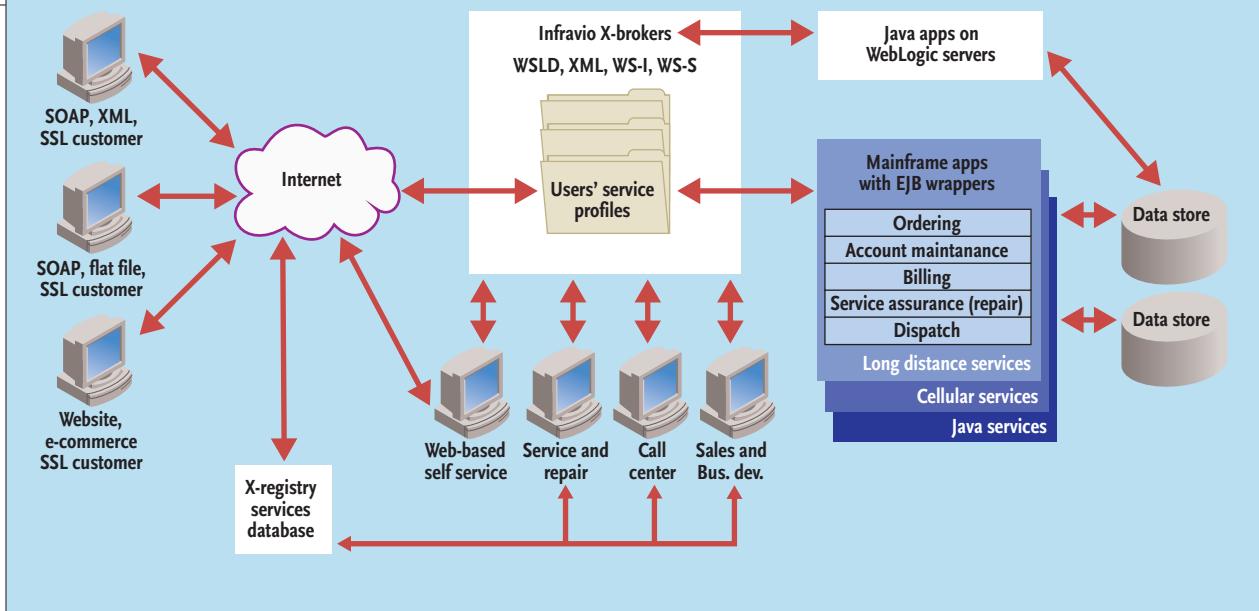


vices, including managing data flow among them. In a few cases, SBS uses the Vitria EAI platform to implement a composite service's process at the Web service level, Vazquez says. SBS also uses BPEL (Business Process Execution Language) to orchestrate services in b-to-b deployments.

In retrospect, Vazquez says going the atomic route is the best option, even though initial development takes longer because of the time it takes to break

Dialing Into Reusable Services

Sprint Business Systems' SOA uses individual profiles that determine what services are available for each user. A service broker can then assemble the needed functions from the service registry, rather than requiring a unique service for each user. The registry also helps IT avoid developing the same services for multiple users inadvertently.



dards, is the rate of adoption among trading partners.”

In the telecom industry, Vazquez explains, adoption of such specific extensions to XML is widespread because customer data and voice communications typically traverse multiple carrier networks and service providers. “But we haven’t done much in terms of defining standard enterprise XML schema,” he adds, explaining that the multitude of functional and customer differences make development of such XML schema standards too unwieldy.

Standards and Practices

Although SBS has “a world-class IT services R&D group that actively researches and monitors emerging standards,” Vazquez notes, “in some cases we simply choose to hold our integration vendor accountable for keeping us interoperable.” The reason is complexity: As more and more WS-* standards are proposed or approved, organizations are faced with increasing development

burdens, as well as greater risk of incompatibility with external users.

For example, SBS uses WSDL 1.1 because that’s what the Infravio platform supports, and Infravio supports it rather than WSDL 2.0 because “that’s what the WS-I organization has done the most compatibility testing with,” Vazquez says. “The only standards that we are really focusing on today are SOAP, WSDL, WS-Interoperability, and WS-Security,” he adds, explaining that those are widely applicable and adopted technologies.

Because vendors interpret standards differently and because over time vendors will likely diverge on which standards they support, Vazquez expects that it will become more difficult to maintain interoperability between trading partners, resulting in a more conservative approach to the adoption of standards. When necessary, he says, SBS will create multiple versions of services, each of which supports distinct standards, rather than trying to devel-

op a single service that can support multiple standards or variations. Otherwise, he says, the benefits of a component-based Web services platform will be negated by the complexity of ensuring interoperability across so many technology variations. Similarly, to keep its development efforts manageable, SBS tries to develop its Web services in Java and its wrappers to legacy systems in EJB, to maintain J2W (Java to Windows) compliance.

The company typically uses BEA WebLogic as its application server for the EJB wrappers, due to its strong interoperability, but it also uses IBM WebSphere application servers in some areas, depending on the specific application or transaction mix. “The idea is to insulate the Web services layer from the underlying technology platform, whether it’s WebLogic, WebSphere, or a mainframe,” Vazquez says.

The use of Web services has also helped to remove a labor-intensive, frustrating aspect of providing external cus-

“The icing on the cake is that internal consumption of the services can accelerate legacy migration.”

— Gayle Sweeney, Sprint

tomers access to SBS systems. Before the WSM was in place, SBS would have to reconfigure the firewall protecting each affected server to allow access to each individual customer. With a WSM, Vazquez says, direct access by the external customer is eliminated, as is the need to configure firewalls for each service. Instead, SBS limits the customer to a broker service hosted in the DMZ, which then calls up services inside the SBS network as needed.

A Platform for the Long Term

When Sprint realized it needed a common architecture for its emerging Web services, it already had a key SOA requirement in place: tight integration between business logic and application development. Sprint historically has been a process-oriented company, Vazquez says, adding that its business development teams have traditionally specified their own application requirements, often working with a development staffer sitting nearby.

“We have four process design tools,” Vazquez says, explaining that Sprint already had the process orientation required to effectively deploy an SOA. He credits that readiness for Sprint’s fairly fast ramp-up when the SOA effort began in earnest two years ago.

Today, the SOA lets Sprint use services in two ways: directly, or through an external Web-based interface for customers that don’t yet have the ability to integrate with Web services on their own. According to Jeff Lentz, who manages services architects at SBS, customers gain the most when they can consume Web services directly, because that approach allows them to encapsulate Sprint’s services in their own applications. That eases data integration, and it also frees their staffs from the need to learn new interfaces or process-

es, as they must if they access services through a Web site. Because most customers are only now starting to develop their own Web-services platforms, however, the Web interface means that Sprint gets immediate ROI.

What’s more, Sprint’s foresight in implementing SOA will help it in its next big challenge: the union of Sprint with its recently acquired subsidiary, Nextel. “The icing on the cake is that internal consumption of the services can accelerate the legacy migration we

face with the merger,” notes Gayle Sweeney, director of Web presence at the carrier.

The flexibility of the SOA approach means SBS no longer needs to reinvent the wheel every time it’s challenged with integrating a new project or initiative. And in the often-chaotic world of enterprise IT, if Vazquez can be certain of one thing, it’s that he’ll have no shortage of those. “In an enterprise of 70,000 people, we’ll always find new applications or efforts under way,” he says. ☛

Services Registry Aids IT, Drives Sales

AS THE SPRINT BUSINESS SERVICES (SBS) IT GROUP RATIONALIZED ITS existing Web services and figured out what services were still needed, it became clear that something else was needed: a registry for Web services. In a large company, having an SOA in and of itself doesn’t prevent various IT groups from duplicating others’ efforts, nor does it prevent customers from asking IT to develop services that already exist. A shared services registry, however, can do that.

Sprint now has about 25 components in its Infravio X-registry, with 25 more to be added this fall and plans for several hundred more in 2006, although it didn’t take this scale to get the ball rolling. “We knew at 15 components that we’d need a registry,” notes Edmund Vazquez, Web services program manager of the enterprise-focused Sprint Business Services (SBS) unit.

Now, internal customers and IT staff can see if a needed service already exists, or if a similar one exists that can be used as a basis to develop a new one more quickly and inexpensively, Vazquez says. Today, “we reuse assets,” he adds.

The benefits of the registry go beyond reducing development costs. Architecture manager Vijay Musuvathy says business managers quickly realized that the registry could serve their customers, including telecommunications users and service providers (for such functions as repair, resale, and installation). They urged that the catalog of Web services be published online (infoworld.com/3189), so that customers and business partners “could see what kind of capabilities they can liberate.”

Vazquez says this catalog has gained Sprint new business. One reason is that customers using Web services standards such as SOAP, WSDL, and WS-Security can integrate these Web services within their own environments, reducing the cost to both the customer and Sprint, and often speeding deployment.

Typically, it’s the sales team that identifies a customer’s need for a service. “It’s not IT calling IT,” Vazquez notes. Ideally, the sales force can promote the use of an existing service — which it knows about from the registry — to encourage more business from customers. — G.G.

Apply online at: <http://subscribe.infoworld.com>

PRIORITY CODE: WW5PDF

I wish to receive a free subscription to **InfoWorld**.
 Yes **No**

SIGNATURE _____ DATE _____

A. MAILING ADDRESS

Publisher reserves the right to limit the number of complimentary subscriptions. Free subscriptions available in the U.S. (including APO and FPO) and Canada

NAME _____

TITLE _____

COMPANY NAME _____

DIVISION / DEPT. / MAIL STOP _____

MAILING ADDRESS _____

CITY / STATE / ZIP / POSTAL CODE _____

Is the above address a home address? 1. Yes 0. No

E-MAIL ADDRESS _____

BUSINESS PHONE (INCLUDING AREA CODE) _____ BUSINESS FAX NO. (INCLUDING AREA CODE) _____

1. Over the course of one year, do you buy, specify, recommend, or approve the purchase of the following products or services worth:

Please include amounts for all locations of your organization. Consultants: please include what you recommend for your clients as well as what you buy for your own business.

- | | | |
|----------------------------------|--------------------------------|----------------------------|
| 01. \$100 million or more | 06. \$5,000,000 to \$9,999,999 | 11. \$100,000 to \$399,999 |
| 02. \$50,000,000 to \$99,999,999 | 07. \$2,500,000 to \$4,999,999 | 12. \$50,000 to \$99,999 |
| 03. \$30,000,000 to \$49,999,999 | 08. \$1,000,000 to \$2,499,999 | 13. Less than \$49,999 |
| 04. \$20,000,000 to \$29,999,999 | 09. \$600,000 to \$999,999 | 14. None |
| 05. \$10,000,000 to \$19,999,999 | 10. \$400,000 to \$599,999 | |

Product category	Write code in box
Large systems	<input type="text"/>
Client computers	<input type="text"/>
Networking / Telecom (including servers)	<input type="text"/>
Wireless	<input type="text"/>
Internet / Intranet / Extranet	<input type="text"/>
Security	<input type="text"/>
Storage	<input type="text"/>
Peripheral equipment	<input type="text"/>
Software	<input type="text"/>
Service/Support / Outsourcing	<input type="text"/>

2. What is your primary job title? (PLEASE CHECK ONE ONLY)

- | | |
|---|---|
| IT / Technology Management | <input type="checkbox"/> 10. IT Staff |
| <input type="checkbox"/> 01. CTO, CIO, CSO, Vice President | <input type="checkbox"/> 11. Other IT Professional |
| <input type="checkbox"/> 02. Director | Corporate / Business Management |
| <input type="checkbox"/> 03. Manager / Supervisor | <input type="checkbox"/> 12. CEO, COO, President, Owner, Vice President |
| <input type="checkbox"/> 04. Network Manager / Director | <input type="checkbox"/> 13. CFO, Controller, Treasurer |
| <input type="checkbox"/> 05. Engineer | <input type="checkbox"/> 14. Director |
| <input type="checkbox"/> 06. Systems Analyst / Programmer / Architect | <input type="checkbox"/> 15. Manager / Supervisor |
| <input type="checkbox"/> 07. Other IT Management | <input type="checkbox"/> 16. Other Business Management Title |
| IT / Technology Professional | |
| <input type="checkbox"/> 08. Consultant / Integrator | <input type="checkbox"/> 98. Other Title |
| <input type="checkbox"/> 09. Developer | (specify) _____ |

3. Please indicate your job function(s)? (PLEASE CHECK ALL THAT APPLY):

- | | |
|--|--|
| IT / Technology Functions | Corporate / Business Functions |
| <input type="checkbox"/> 01. Executive | <input type="checkbox"/> 09. Executive |
| <input type="checkbox"/> 02. Department Management - IT | <input type="checkbox"/> 10. Department Management - Business |
| <input type="checkbox"/> 03. Networks / Systems Management | <input type="checkbox"/> 11. Financial / Accounting Management |
| <input type="checkbox"/> 04. Applications Development | <input type="checkbox"/> 12. Research / Development Management |
| <input type="checkbox"/> 05. Management of Enterprise Applications (CRM, ERP, SCM, etc.) | <input type="checkbox"/> 13. Sales / Marketing Management |
| <input type="checkbox"/> 06. Research / Development Management | <input type="checkbox"/> 14. Other Business Functions |
| <input type="checkbox"/> 07. Consultant / Integrator | |
| <input type="checkbox"/> 08. Other IT Functions | <input type="checkbox"/> 98. Other Functions |
| | (specify) _____ |

4. Are you involved in buying, specifying, recommending or approving the following IT products / services?

(PLEASE CHECK ALL THAT APPLY):

- | | |
|---|--|
| Software / Products / Technologies | <input type="checkbox"/> 18. Web / Video Conferencing |
| <input type="checkbox"/> 01. Customer Relationship Management | <input type="checkbox"/> 19. Storage |
| <input type="checkbox"/> 02. Enterprise Resource Planning | <input type="checkbox"/> 20. Disaster Recovery |
| <input type="checkbox"/> 03. Business Process Management / Outsourcing | <input type="checkbox"/> 21. Security |
| <input type="checkbox"/> 04. Business Intelligence / Data Mining / Data Warehousing | <input type="checkbox"/> 22. Anti-Virus / Content Filtering |
| <input type="checkbox"/> 05. Portals | <input type="checkbox"/> 23. Firewall |
| <input type="checkbox"/> 06. Financials / Payroll / Billing | <input type="checkbox"/> 24. VPN |
| <input type="checkbox"/> 07. Performance / Application Management | <input type="checkbox"/> 25. Identity Management |
| <input type="checkbox"/> 08. .NET | <input type="checkbox"/> 26. Authentication / Authorization |
| <input type="checkbox"/> 09. Other Software | <input type="checkbox"/> 27. Intrusion Detection & Prevention |
| <input type="checkbox"/> 10. Networking | <input type="checkbox"/> 28. Encryption |
| <input type="checkbox"/> 11. Web Services | <input type="checkbox"/> 29. Other IT Products / Technologies |
| <input type="checkbox"/> 12. Content Delivery Networks | Hardware / Peripherals |
| <input type="checkbox"/> 13. Network and Systems Management | <input type="checkbox"/> 30. Servers |
| <input type="checkbox"/> 14. VoIP (Voice Over IP) | <input type="checkbox"/> 31. Notebooks / Laptops |
| <input type="checkbox"/> 15. Telecommunications | <input type="checkbox"/> 32. PDAs / Handhelds / Pocket PC / Wireless |
| <input type="checkbox"/> 16. Wireless | <input type="checkbox"/> 33. Printers |
| <input type="checkbox"/> 17. Remote Access | <input type="checkbox"/> 34. Other Hardware / Peripherals |

5. What is your organization's primary business activity at this location? (PLEASE CHECK ONE ONLY):

- | | |
|--|--|
| General Business Industries | Technology Providers |
| <input type="checkbox"/> 01. Defense Contractor / Aerospace | <input type="checkbox"/> 12. Service Provider (MSP, BSP, ISP, ASP, etc.) |
| <input type="checkbox"/> 02. Retail / Wholesale / Distribution (non-computer) | <input type="checkbox"/> 13. Computer / Network Consultant |
| <input type="checkbox"/> 03. Pharmaceutical / Medical / Dental / Healthcare | <input type="checkbox"/> 14. Systems / Network Integrator, VAR / VAD |
| <input type="checkbox"/> 04. Financial Services / Banking | <input type="checkbox"/> 15. Technology Manufacturer (hardware, software, peripherals, etc.) |
| <input type="checkbox"/> 05. Insurance / Real Estate / Legal | <input type="checkbox"/> 16. Technology - Related Retailer / Wholesaler / Distributor |
| <input type="checkbox"/> 06. Transportation / Utilities | Government / Education |
| <input type="checkbox"/> 07. Media (print / electronic) | <input type="checkbox"/> 17. Government: federal (including military) |
| <input type="checkbox"/> 08. Communication Carriers (telecomm, data comm., TV / cable) | <input type="checkbox"/> 18. Government: state or local |
| <input type="checkbox"/> 09. Construction / Architecture / Engineering | <input type="checkbox"/> 19. Education |
| <input type="checkbox"/> 10. Manufacturing & Process Industries (non-computer) | <input type="checkbox"/> 98. Other |
| <input type="checkbox"/> 11. Research / Development | (specify) _____ |

6. How many people are employed at this organization, including all of its branches, divisions and subsidiaries?

(PLEASE CHECK ONE ONLY):

- | | |
|---|--|
| <input type="checkbox"/> 1. 20,000 or more | <input type="checkbox"/> 5. 500 - 999 |
| <input type="checkbox"/> 2. 10,000 - 19,999 | <input type="checkbox"/> 6. 100 - 499 |
| <input type="checkbox"/> 3. 5,000 - 9,999 | <input type="checkbox"/> 7. 50 - 99 |
| <input type="checkbox"/> 4. 1,000 - 4,999 | <input type="checkbox"/> 8. Less than 49 |

7. Which of the following operating systems are in use or planned for use at this location?

(PLEASE CHECK ALL THAT APPLY):

- | | |
|--|---|
| <input type="checkbox"/> 01. Windows XP | <input type="checkbox"/> 04. Linux / Unix / Solaris |
| <input type="checkbox"/> 02. Other Windows | <input type="checkbox"/> 05. Other |
| <input type="checkbox"/> 03. Mac | (please specify) _____ |

B. CONTACT PREFERENCES

You may receive a renewal reminder via e-mail. May we send other information about InfoWorld products, services, or research via e-mail? 1. Yes 0. No

We occasionally send our subscribers email messages with news about technology solutions and special offers from qualified third parties. Would you like to receive these messages? 1. Yes