



Web Services: Paving the Way to SOA
January 2007 -- Susana Schwartz

“Web Services” comprise a body of standards and protocols that have become vital in the drive toward SOA, as they define how services interact in business transactions. Earlier fears were that Web Services might be CORBA revisited or client/server reborn. This group of interfaces and protocols, however, are opening up telecom applications to the outside world and will perhaps move telecom toward SOA.

The key to achieving SOA is loose coupling, componentization and orchestration facilitated by Web Services. This is significant to telecom’s future because the next wave of services will rely on applications from developers outside of telecom. Web Services’ collection of standards create a common denominator among execution and operations environments so that applications can be created to represent reusable business operations at all processing levels

That sets the stage for SOA, which is supposed to enable operators to abstract implementation and platform details from data sources in such a way that users and consumers can manipulate information without intimate knowledge of its origins. The specific data source won’t matter to the user, and that could foster flexibility to change services without having to change underlying business processes or architecture.

“In trying to converge telecom infrastructure and the IT world, the integration becomes more complicated due to lack of standards,” says CIO Cenk Bayrakdar of Turkcell, the leading mobile operator in Turkey, with more than 30 million subscribers. “I think of Albert Einstein’s quote: ‘Things should be made as simple as possible, but no simpler.’”

For Bayrakdar, a central layer of Web Services has already become the key to a list of capabilities, including:

- _ Establishing a standard security policy and controlled authorization and authentication facilities
- _ Decoupling business decisions from other business logic
- _ Registering Web Services in a standard, easy to understand catalogue format
- _ Executing exception-handling functions supported with standard GUIs and error logging
- _ Centralizing performance monitoring, control and operational activities reporting
- _ Creating a centralized solution for disaster recovery
- _ Providing a documented set of rules and conventions while developing new business processes.

SOA may be particularly beneficial to OSS/BSS development because it is designed to foster communication among many disparate, horizontally distributed components.

"OSS/BSS lends itself to SOA because billing and operations are entities that run horizontally across many services and applications deployed inside carriers' networks. Those components have to be shared by everyone involved in a service's delivery, which means multiple billing back ends must be brought together if services are to be integrated across multiple servers and applications," says Chris King, senior director for worldwide telecom markets at BEA Systems. The company has incorporated SOAP and XML within its SOA 360 platform, which spans Tuxedo, WebLogic and AquaLogic products.

While Web-services-based SOA promises business process interoperability over the Internet, it is in no way a panacea at this point in time. Both SOA and Web Services are immature enough that the two are used synonymously. This reflects confusion about their true definitions and about the differences between web services capabilities and the overall SOA mission. Some say the difference is in the granularity with which new services are presented, or depends on how much visibility exists into accessible services people and across applications.

The degree of success with an SOA depends on the degree of visibility into functions like provisioning and service activation, as well as systems and data. That transparency dictates how much reuse will be possible by a larger audience of technical and non-technical people. It's that reuse of components that is fostered by Web Services, as it provides an integration layer facility.

The belief is that SOA principles will drive carriers to seek out reusable components for location, presence and call control to be assembled in new applications. Then, a sort of ecology of services can be deployed in any type of network. That could enable a service provider or ISV to bring in any OSS application and quickly connect it, regardless of the underlying architecture's components.

Getting Through the Hype

Many vendors and integrators claim to be using Web Services within an SOA, but a closer look usually reveals that what they're hawking is actually enterprise application integration (EAI) based on Web Services. The fact that they are using Web Services rather than JMS or CORBA just means that they have used different "connectors" to create a common framework for applications. It doesn't necessarily mean a true SOA exists.

Service providers need to look for collaboration among competing systems that run horizontally across many services and applications. For business people it could mean the creation of a set of services that customers and partners can leverage. For engineers, it could mean finding a style or method of modeling that helps componentize environments through a collection of patterns and sequences of processes.

Regardless of the ultimate user—whether developers or business people--Web Services could help to automate the business process creation and management activities important to orchestrating relationships among interrelated, low-level services.

"The beauty of SOA is that none of the applications have to talk directly to one another when Web Services are used. You avoid hardwiring by using Web Services APIs that then talk to SOA governance systems," explains CTO John Wilmes of Ceon, a provider of service fulfillment and provisioning platforms based on J2EE and OSS/J.

He says that using Web Services allowed Ceon customer France Telecom to conceptualize integrations on paper. "They could write the service contracts ahead of time and then collaborate on the components of service creation, usage and billing later," says Wilmes. "They can create the service, productize it, get it ordered, activated and billed in a much faster time frame."

Wilmes notes that France Telecom has strong SID-based interfaces. "That means they can pick what eTOM processes they want, and then lay out the data that ties to the processes," he says. eTOM is often used as a template when breaking down processes into atomic operations so that they can be reassembled to build services. Many companies are using processes and terms in eTOM and then tying them to the SID so data can be exchanged. In cases involving legacy systems that don't use SID, semantic data integration in the enterprise service bus (ESB) translates the meaning of content that a service needs for provisioning certain data or operations.

TMF's NGOSS has evolved to utilize underlying processes from eTOM and SID, as well as APIs resulting from the merger with OSS/J. The latter offer XML Web Services profiles in addition to Java profiles. In the drive to broaden visibility and access, there are a few Web Services that have become particularly popular: SOAP, WSDL and UDDI have thus far become a key to integrating loosely coupled Web-based elements that expose standardized interfaces. WSDL interface descriptions enable the definition of static interfaces of a Web service. They are often coupled with SOAP, which supplies an XML-based messaging framework for applications that conduct simple interactions.

"BPEL also has become a strong standard, as it has helped carriers to free business processes from reliance on provider interfaces or Java code written against low-level interfaces or proprietary business process engines," says Vittorio Viarengo, Oracle's vice president of development. "Something like BPEL has become instrumental in BSS integration with the service plane, where services are executed with billing, customer service, customer care and provisioning."

In the context of SOA, Web Services like Parlay X have become important for integration with key network capabilities, such as call control through high-level APIs that help to expose the network responsibilities to third parties. Already, Parlay and Parlay X have enabled the development of interactive services between telecom and non-telecom entities.

Convergence of Network and Operations

Such abilities may help to drive convergence among operations and network organizations. However, there is still a lot of sensitivity about rushing into SOA when it comes to real-time call paths. Because IMS and SS7 signaling are critical to heavy, real-time use, Web Services are associated more with running and managing IT functions that don't involve huge numbers of real-time transactions.

Because of concerns about network impact, the Web Services' capability to invoke services, bundles and features using reusable infrastructure and SOA design and concepts concentrates right now on IT application servers, email servers, messaging and content servers. But the future will most likely mean provisioning feature sets quickly. It remains to be seen whether five-nines reliability for setup and tear-down will ultimately pull the focus toward the network. For now, the science of originating and terminating calls is not the problem; rather, what's difficult to achieve is service

diversity and expedited delivery of service offerings. Consequently, those are the current focus for Web Services standards.

Because most of that work revolves more around IT than network folks, the stewardship of SOA currently resides in the IT side of the house. However, there is momentum on the network infrastructure side of the equation as well, as mobile operators make more use of Parlay X and Web Services to expose networks to third parties that are building out services.

Who Champions SOA?

A pain point for EAI projects in the past often was operational frustration stemming from a lack of stewardship or accountability. That was attributable to the fact that, with EAI, the business logic was embedded in the systems being integrated.

Hard-coding did not lend itself to agility, so SOA brings the promise of mechanisms for combining, reconfiguring and reassembling components with simple adaptations. Those mechanisms—most likely Web Services standards—have to be collected and managed, and their dissemination enforced, by a “champion” supported by all facets of the organization.

“Rather than thinking solely about creating new functionality, all functional areas of the organization have an opportunity to think in harmony about the processes, network elements and data that can be shared by each service. That helps us to take advantage of possible overlap or synergies,” says Turkcell’s Bayrakdar. “The IT organization might be in charge of formulating requirements for something like a rating engine, but it’s important to include the C-level executives in the SOA program.”

Bayrakdar believes success can be achieved when SOA initiatives start with a defined and shared program vision, objectives, and scope. “That requires extensive communication between C-level executives and program managers,” he says, adding that consultancy teams were valuable to his organization in spreading “SOA know-how” within the organization.

Governance is supposed to be a natural byproduct of SOA initiatives, as buy-in must come from many groups within an organization. Because many groups stand to benefit from reusing components, cross-functional teams should share the costs of development. Another natural byproduct should be the elimination of any duplicate work.

According to BEA’s Benjamin Renaud, senior director for telecom products, “Most carriers have teams of architects in place who inevitably will become stewards in charge of maintaining and promoting the use of the library of Web Services.”

Since SOA forces centralized management, it already exists in some form at most carriers. “Even if it has not been called ‘SOA,’ carriers have had initiatives underway for years to foster reuse, which means they have worked with thousands of Web Services and Web Services interfaces. Now, it’s up to these architects to ensure that the Web Services are made available and published so that developer teams take advantage of them,” says Renaud.

To make “loose coupling” mandatory, there have to be rules that make transparency mandatory rather than optional. According to Kevin Twardus, manager of software

architecture for the communications industry at IBM, that means "location transparency," which ensures that no end-user service is hard-coded to any end point. "There has to be protocol transparency. The nails and glue are out, and the hinges and screws" are in, meaning things must be easy to take apart and put back together.

For that reason, more collaboration between IT and network groups will be necessary. "Both need to tap into OSS/BSS infrastructure to enable access to quotations, or presence, group lists and contact management services," says King at BEA Systems, who believes uniform access to infrastructure is possible through Web Services.

"There needs to be a much bigger thought process. Rather than just checking off functionality, you have to constantly think of the possibilities of Web Services enablement for other components that might fall outside of what you are working on at the present time," says Twardus. "That means you have to really change the way your organization functions to capitalize on SOA."

Perhaps rather than have an IT organization formulate requirements for a rating engine, a cross-functional team could correlate requirements across the life cycle of a service. A holistic view can only be assembled by business people, operations people, and engineers who truly understand the function of particular elements from the perspective of each of the other departments.

To help with that, an organization must implement metrics and tools to capture requirements in the language understood by all parties. For example, line of business executives talk of "charging," while engineering may talk of "rating." The network engineers have to understand what information generated at various collection points would be valuable to the business.

"Rather than drop new requirements on engineering, which then gets frustrated at having to start from ground zero, there has to be a repository where the charging requirements of the business people are linked to the collection requirements of engineering and to the output requirements for the operations folks," Twardus says.

If there is understanding among the many facets of the organization, there is "traceability" of the life cycle of a service. But traceability starts with the all-important steward or champion that must drive the process, whether that be engineering, IT or business lines.

Service Contracts Yield Governance

Internal and external service contracts are very important for ensuring that internal processing of applications is exposed consistently. Service contracts will have to detail specifics about where applications and governance systems interact and what data will be necessary. "The contracts really help the people putting together SOA to think in terms of services that make the most business and applications sense," says Wilmes.

He believes that "true" SOA inherently fosters governance practices that enforce service contracts at a software level. "The contracts are needed to clarify who the actors are and what their roles will be in service delivery," he says. "It's important to go to the pre- and post-conditions of interactions so that the governance systems can manage the interaction as information is exchanged." Wilmes believes the governance system will likely be a bus underneath an ESB that distributes service

requests and responses as needed.

The ESB separates application logic and integration tasks to foster message-oriented SOA. It's the ESB that handles the translation among elements that need semantic data integration, where divergent service requests and responses are defined in a common manner so all systems involved can work together.

In anticipation of the increased emphasis on tangible service contracts, AT&T, BT and several other carriers have started a working group under TMF to focus on NGOSS contracts at four levels: business, systems, implementation and deployment.

As the number of service contracts grows for trouble ticketing, product and service inventory interactions—and as product catalogs grow in number—carriers will be able to do more internal “mash-ups,” creating applications using public Web Services interfaces. That way, anything already annotated with open APIs will be reusable for developing more novel applications. That will make ISV applications more valuable, as piece parts can be mixed and reused elsewhere for other services.

Today's Development Gap

There is a development lag between SDP and OSS that is becoming more apparent (see “SDP and OSS/BSS: On a Collision Course or Ready for Integration?” December 2006, p.10). There are demos where SIP servlets are developed in minutes, but the service creation takes a long time because of the lag between OSS/BSS and SDP. The lag happens because new services are still communicated upstream to a product life cycle management (PLM) system that usually needs a swivel chair and a pair of hands on a keyboard.

Some believe a key to closing the gap will be to integrate the service creation environment (SCE) and upstream systems like PLM. “PLM can bridge the gap between OSS and SDP when preparing for fulfillment,” says Wilmes. “It can tie to billing and assurance planes as the bridge between OSS and SDPs.”

An SCE should have an IMS SIP server and several OSS systems working together with SOA governance systems. “The belief is that then, the SCE can fill the gap between the systems doing PLM and the OSSs and SDPs,” Wilmes says. “The PLM is needed to integrate with upstream billing and CRM, and into the SDP and service creation environment. That way, the SCE can quickly invent and deploy services on SIP services, and that information can quickly be disseminated in SDP systems.”

Because SOA integrations depend on SDP and OSS elements, new projects are underway to try to resolve the issues. For example, OSS Blueprint is a fairly recent catalyst initiative to demonstrate live SOA incorporating elements of OSS and SDP.

As these types of projects evolve, it's important for service providers to keep their focus on what's important: The goal is to have SOA, so that Web Services standards defining how services interact in business transactions will enable a broad array of people to build and integrate business applications.

For that reason, ISVs will probably feel growing pressure to expose their internal processing so that not only IT and engineering types can view applications more readily, but also non-technical people as well.

Additionally, systems integrators will increasingly seek to work with vendors

demonstrating flexibility in how data and processes are exposed, which is further incentive for ISVs to map out information about implementation specs and the inner workings of their provisioning and activation systems.

As more becomes transparent about the processes involved in fulfillment and billing, the more likely it is that open, interoperable interfaces will be used to replace the hardwiring in what traditionally have been monolithic, opaque systems.

With the exposure of many layers of information, other applications will be able to reuse data so that business processes can run in application layers, thus fostering agility and flexibility—and perhaps even creating a true SOA.

<http://www.billingworld.com/content/print.cfm?ArchiveId=7820>