CAN EAI AND WEB SERVICES WORK TOGETHER?

May 2003 Volume 3 Issue 5

WebServices JOURNAL

.NET J2EE XML

FOCUS ON EAI

RETAILERS PLEASE DISPLAY UNTIL JUNE 30, 2003
$6.99US $7.99CAN
05
0 71486 03420 9
SYS-CON MEDIA

WebServices JOURNAL
.NET J2EE XML

INTERNATIONAL ADVISORY BOARD
Andrew Astor, David Chappell, Graham Glass, Tyler Jewell, Paul Lipton, Anne Thomas Manes, Norbert Mikula, Frank Moss, George Paolini, James Phillips, Simon Phipps

TECHNICAL ADVISORY BOARD
Steve Benfield, Bernhard Borges, JP Morgenthal, Andy Roberts, Ajit Sagar, Michael A. Sick, Simeon Simeonov

EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
EDITORIAL DIRECTOR
Jeremy Geelan jeremy@sys-con.com
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Joe Mitchko joe@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
TECHNICAL EDITORS
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
EXECUTIVE EDITOR
Gail Schultz gail@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com
ASSOCIATE EDITORS
Jamie Matusow jamie@sys-con.com
Jean Cassidy jean@sys-con.com
ASSISTANT EDITOR
Jennifer Van Winckel jennifer@sys-con.com

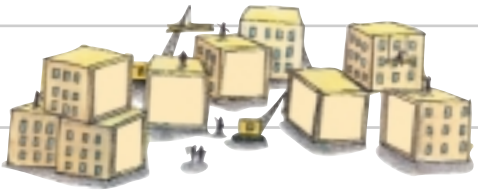
PRODUCTION
PRODUCTION CONSULTANT
Jim Morgan jim@sys-con.com
LEAD DESIGNER
Richard Silverberg richards@sys-con.com
ART DIRECTOR
Alex Botero alex@sys-con.com
ASSOCIATE ART DIRECTOR
Louis Cuffari louis@sys-con.com
ASSISTANT ART DIRECTOR
Tami Beatty tami @sys-con.com

CONTRIBUTORS TO THIS ISSUE
Andy Astor, Mark Little, Paula Lubet, Jim Mackay, Joe Mitchko, Carol Murphy, Adelene Ng, Sanjay Patil, Eric Pulier, Sean Rhody, Nick Simha, Jim Webber

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9637
WEB SERVICES JOURNAL (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending
Montvale, NJ 07645 and additional mailing offices
POSTMASTER: Send address changes to:
WEB SERVICES JOURNAL, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

SYS-CON MEDIA

Let Go My LEGO



Written by
Sean Rhody

Author Bio:
Sean Rhody is the editor-in-chief of Web Services Journal and managing editor of WebLogic Developer's Journal. He is a respected industry expert and a consultant with a leading consulting services company. SEAN@SYS-CON.COM

When I was a kid, I loved to put things together. I especially liked building things with LEGO. Never mind the TV, dump a huge pile of LEGO blocks in front of me and I'd be quiet for hours, building a LEGO version of the Eiffel Tower, or some other construct that only a kid could imagine.

One of the things that made it possible for me to test the limits of LEGO architectural stability was the fact that it had all been designed to work together. Every LEGO piece was created with the sole intention of fitting together with another block of LEGO. Didn't matter what color the blocks were, how high, or what shape, they all worked.

Unfortunately, in the IT world software isn't quite LEGO. It doesn't have the advantage of all coming from a single vendor, and even in the more limited cases when it does, it frequently isn't designed to work together. Sometimes it can be like trying to make the LEGO fit to some of those cheap imitation blocks that value price stores try to sell you – a painful proposition, but sometimes it worked. Sometimes it can be like trying to build with LEGO and an Erector set; both are good tools, but neither fits the other's idea of how to put the world together.

Enterprise application integration (EAI) is a software approach to solving this problem. Much like the approach you might take to making LEGO and the Erector set work together, EAI serves as the glue between applications.

It also solves the input impedance problem. Take three systems. If you wire them each to work with the others, you have three times three, or nine adapters to write. If you add a fourth system, you end up with sixteen, and so on. EAI, by acting as a bus on which communication takes place, reduces that number to one adapter per system, vastly reducing the complexity of application integration.

EAI also offers the ability to manage aggregation of software into a higher-level business service. This is not a trivial task, as it requires coordination of transactions, systems, and a mechanism for compensating transactions for the cases where a transaction must be rolled back. The value of this capability is enormous. Many organizations have a heavy investment in technologies that they cannot or will not modify for strategic reasons. Their business processes, on the other hand, change and grow dynamically, and as new software becomes available, parts of functionality tend to migrate from one application to another.

Take, for example, the definition of a "customer." Typically this information resides in a system of record. But when you add a CRM application, and allow for Web-based self-registration, the location of the customer information may migrate from the existing legacy system into the CRM system. But the old system still needs the information. EAI helps solve this problem.

Web services has become the next step in the evolution of EAI. While it is easy to believe the concept that Web services provides all you need for EAI, it's not really the case. UDDI, WSDL, XML, and SOAP provide the basic underpinning of application cross-communication, that's very true. But the real value added by EAI is not just in the plumbing, it's in the management of services as they are needed across the enterprise bus. While Web services standards such as WS-Orchestration are designed to address parts of this issue, it's clear that EAI already has a large head start in that area, and will be well positioned to take advantage of those specifications.

This issue focuses on Web services and EAI – how to use them, how they impact each other, and how they complement each other. Kind of like LEGO and the Erector set: once you can get them to work together, you can accomplish amazing things. Now where did I put my glue gun?

Mindreef

www.mindreef.com

EAI Industry Health Check

A future that may not be bleak

Demand for business integration continues to intensify across a broad range of industries. Yet disappointing returns from enterprise application integration (EAI) projects and increased competition are conspiring to squeeze pure-play integration vendors from all directions. In this article, we'll explore some of the current challenges faced by enterprise integration vendors.

AUTHOR BIO:

Carol Murphy is a partner with CSC Consulting and the solution architect for its enterprise application integration (EAI) practice. Carol educates clients on using EAI technology for business integration, including vendor selection, project planning and implementation best practices. She coordinates development of CSC Consulting EAI service offerings and is the primary author of the CSC Business Integration Practice Guide.

CMURPH23@CSC.COM

The Integration Imperative

For most industries – manufacturing, retail, financial services, healthcare, government, aerospace and defense, natural resources – visibility and access to information flowing through integrated business processes has never been more important. Integrated business processes include activities involving multiple systems and/or people, often crossing organizational boundaries such as departments, divisions or enterprises. Examples include order-to-cash, trade-to-settlement, claims processing, service provisioning, logistics management, and customer self-service.

Enterprises continually strive for agility through improved business process management, enhanced trading partner connectivity, supply chain optimization, and collaboration in design and manufacturing. Many enterprises want to adopt common processes for business functions yet support local variations in certain activities. Removing latency (non–value-added time) in business processes is a major driver for straight-through processing in financial services. Increasing information portability and security is the primary goal of the HIPAA mandate in health care. Increased collaboration between independent entities is key for U.S. homeland security and for collaborative design initiatives in aerospace and manufacturing. The common thread across these efforts is what *CIO* magazine calls "the integration imperative."

From an IT standpoint, these initiatives all require increased intra-enterprise integration between front- and back-office applications, databases, legacy systems, Web portals, and other Web applications. Inter-enterprise integration is also often involved among trading partners using XML-based protocols, EDI, or Web services. To facilitate rapid integration, differences between operating platforms, communication protocols, data representations, and application interface styles must be minimized. Enterprise integration must be distributed, scalable, reliable, manageable, and secure. Often it must be "transactional" – recoverable in case of error. (Cross-application transaction processing is conceptually similar to two-phase commit. However, rather than directly rolling back failed transactions, "compensating" activities are invoked to effectively undo the original, failed transaction.)

To remain competitive, most medium-to-large enterprises now recognize that application integration must become a core competency. Sophisticated enterprise architectures incorporate integration services as a required component. Core integration services include messaging and message routing, adapters for connecting to applications and other data sources, data transformation, business process automation, trading partner management, metadata management, and system management and monitoring. Pure-play EAI vendors such as SeeBeyond, webMethods, TIBCO, Vitria, and many others have developed technology specifically to address the integration imperative.

Business Integration Challenges

Despite, or perhaps because of, the hype surrounding EAI two or three years ago – similar to the hype surrounding Web services today – return on investment (ROI) for some EAI projects has been disappointing. Contributing factors include lack of business sponsorship, unrealistic or poorly managed expectations, weak program governance, product instability, and inappropriate use of the technology (which happens more often than you might think). Two key factors include underestimating the complexity of the task and lack of an architectural approach for using sophisticated EAI technology.

Underestimating Complexity

Caught up in their own hype, EAI vendors oversold the ease with which EAI technology enables business integration. EAI marketing presentations would lead customers to believe they can achieve complex integration in a few short weeks. Customers should realize that these estimates typically only include the time to actually configure the EAI software; they conveniently ignore the time needed to determine what should be configured and why. Analyzing

specific integration requirements, designing a reusable solution, and hardening it for a production environment are fundamental activities for any industrial-strength solution using EAI technology. These will substantially increase a project development schedule beyond original EAI vendor numbers.

Customers undertaking EAI projects often underestimate the organizational commitment required to achieve business integration, which is not the same as application interfacing. Business integration is about improving business processes to ensure that information flows seamlessly and automatically throughout the process. Application interfacing is about physically enabling applications to send or receive data, with little regard for what happens beyond the application's boundaries (i.e., the integration context). While often considered a tedious chore by developers and project managers, application interfacing is critical to but not sufficient for effective business integration.

Actively involving both business process owners and IT integration experts in planning and design activities is a critical success factor for business integration. With active participation from the business community, we've been able to identify $3 in process-related improvements from EAI technology for every $1 in cost reduction. Thus, engaging the business and IT communities early, not only bolsters the business case needed for executive sponsorship, it also helps build the excitement and organizational alignment needed for effective business integration.

You would never hand someone a box of tools and expect them to build a skyscraper. Skilled architects must design the building, taking into account functional requirements and making trade-offs based on design criteria such as building codes, materials, and project budget. Only then can experienced tradesmen from many disciplines construct the building. They use best practices, experience, and state-of-the-art tools to ensure that it is secure, scalable, and reliable. The same is true for EAI projects. In other words, business integration doesn't magically happen just because you've installed the EAI toolbox.

Ad Hoc vs Architecture

Another common mistake is to take an ad hoc rather than architectural approach to using EAI technology. Most people now accept that point-to-point integration is expensive, yet it is still easy to build such solutions using EAI tools. Remember, none of these tools will prevent you from creating a poor implementation! Like the skilled tradesmen mentioned earlier, successful business integration projects require experienced practitioners using a proven methodology and best practice techniques.

For example, CSC developed a methodology that takes a disciplined and holistic approach to EAI implementation, with a strong focus on analysis and optimization of integrated business processes and on the architectural issues surrounding use of EAI technology. We encourage business and application technical experts to jointly develop business integration requirements such as:

- What are the activities, events, and information flows for each integrated business process? What are the pre-conditions or post-conditions (if any)? Is this process similar to any others in the enterprise?
- What is the nature of the information passing through each activity? How do other processes or applications use this information?
- How are process exceptions handled? Are compensating activities required?
- Are there any service levels or timing constraints that must be maintained?

It is also important during this phase to address areas not fully covered by EAI vendor products or methodologies such as configuration management, change control, organizational impact, security, system integration testing, and production system deployment. Addressing such requirements early in the architecture phase helps identify integration patterns and uncover opportunities to create reusable integration artifacts (business process models, integration entities (data), adapters, transformations, etc.). This is crucial for delivering on a fundamental value proposition of EAI – reuse – that is often not realized in practice.

EAI projects that minimize or ignore the architectural approach do so at their peril. One client launched multiple independent integration projects using a shared EAI infrastructure, with little coordination between the projects. In the short term, this strategy worked. However, around the time that the fifth or sixth project was deployed, the client's EAI IT infrastructure fell over! Upon investigation, they discovered that system resources were depleted, due in large part to each project having created redundant schema that consumed a large amount of memory (e.g., four almost-identical definitions for a purchase order). With proper architecture and project coordination, these could have been designed for shared use, dramatically reducing both the system load and the effort expended by each project.

Another EAI project left configuration management until very late in the project. Much to their chagrin, the development team discovered, just prior to system integration test, that it was very difficult to migrate code between development, test, and production environments in an automated manner using their EAI tool. In an 11th hour scramble, they were forced to hand-migrate all the code in order to meet the project deadlines. Taking a holistic approach that includes configuration management as part of architecture could have avoided this embarrassing situation.

Increasing Competition

EAI vendors are continually challenged to recast their value propositions. More and more integration services are being standardized by the J2EE and .NET platforms, and industry consortia such as W3C. Emerging integration standards include JMS for messaging, JCA for application connectors, XSLT for data transformation, and multiple Web services standards for application service definition, invocation, and coordination. While these developing standards still lack the functionality currently available from leading pure-play EAI vendors, it's clear that many of the services offered by the pure-play integration vendors are rapidly becoming commodity items. Consequently, EAI vendors are being forced to move up the value chain. They now emphasize such capabilities as business process management (BPM), business activity monitoring (BAM), Web services integration, and increased support for vertical industry processes or protocols.

Competitors are coming out of the woodwork. In terms of functionality, business process management vendors – including Fuegotech, Savvion, and Intalio – continue to push the envelope of process-centric business integration. BPM provides visibility and control over the work of people, systems, and partners. This goes beyond simply integrating applications and automating business processes; sophisticated BPM systems also support business process design, process simulation, process execution, process monitoring, and process analysis. In terms of price, a host of newer vendors such as Fiorano and Sonic Software now offer JMS-based integration suites at a fraction of the cost of established players. Additional competition comes from vendors in related disciplines such as extract-

transform-load (ETL) and electronic data interchange (EDI), who are redefining their offerings as integration solutions.

Most threatening are the industry "800 lb. gorillas," who are clearly taking aim at the business integration market. These offerings will appeal to those customers who prefer to stick with an industry leader even if a smaller, less proven company may have superior technology. IBM announced significant advancements to its WebSphere Business Integration portfolio that provides businesses with a broad set of functions to integrate, manage, and monitor business processes across an enterprise and with trading partners. BEA announced WebLogic Integration 8.1, which provides customers with a unified framework for business integration, simplified production and management, and a new extensible architecture for the rapid assembly and integration of applications, business processes, and trading partner communities. Microsoft's upcoming "Jupiter" technologies will include revolutionary business process management and monitoring capabilities; additional support for XML Web services standards, including Business Process Execution Language for Web Services (BPEL 4WS); and richer developer and information worker support through enhanced integration with Visual Studio .NET and Office.

The leading application vendors are also joining the party. Having finally recognized that they are not the center of the universe, most have responded by adding integration brokers and/or pre-built business processes to their application suites. Some have partnered with EAI vendors to provide integration technology; others opted to build it themselves. SAP recently announced NetWeaver, the foundation of the SAP xApps and mySAP Business Suite solution, which offers a comprehensive integration and applications platform. PeopleSoft's Integration Broker offers integration using Web services, flat files, custom code, or a JMS-based connector to IBM MQSeries. Yantra's enterprise software for real-time coordination of the extended supply chain includes a business process–oriented integration platform and prebuilt processes for coordinating and controlling distributed orders and inventory across multiple business units, customers, suppliers and trading partners. Siebel has taken a slightly different tack by creating the Universal Application Network (UAN), which claims to transform application integration "from a complex and expensive technical challenge into the strategic ability to implement customer-facing business processes across and beyond the enterprise." (All the leading pure-play EAI vendors, as well as IBM and Microsoft, have announced support for UAN.)

Economically, it was a tough year for the pure-play EAI software vendors. All have seen dramatic revenue reductions as EAI projects are delayed, scaled back, or cancelled outright. Several vendors are responding to the perception that EAI projects are too expensive by reducing license costs. At least one vendor now reports more revenue from software services than software licenses (an untenable situation for a software product vendor). Most pure-play EAI vendors have reduced their workforces and all are rapidly burning through their cash reserves; some have seen their stock market valuations essentially evaporate.

Despite competing claims, the integration marketplace is still very much a horse race. Although there are half a dozen "usual suspects," none can yet really be considered the dominant player. Stiff competition means everyone must continue to spend significantly on marketing, sales, and engineering in order to win market share, while continuing to reduce costs. Now that the established industry players are joining the fray, the competition will only intensify.

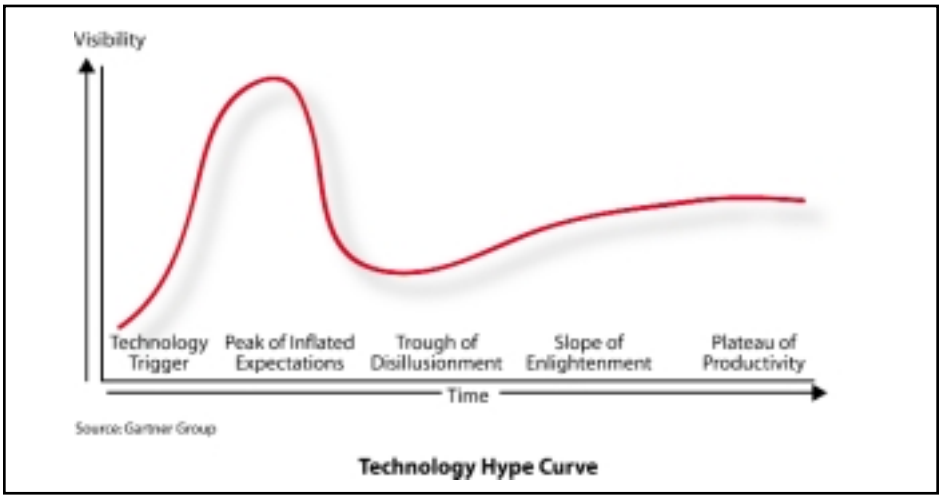


FIGURE 1 **Gartner Group's Technology Hype Curve ("When to Leap on the Hype Cycle", Gartner Group Strategic Planning, SPA-ATA-305, written by J. Fenn, January 11, 1995.)**

Going Forward

Though EAI technology has been available for several years now, significant challenges still exist to successfully deploy it. To be fair, not all the problems are directly related to EAI technology itself; they have as much to do with effectively managing complex programs, managing change in large organizations, and nuts-and-bolts system integration expertise. As with any new technology, there have been some teething problems but the integration approach is essentially sound. While the road to business integration has been rocky, it would be a mistake to conclude that the future for EAI is bleak.

Far from being a flash in the pan, it seems clear that EAI technology is following the historical path described by the Gartner Group's Technology Hype Curve (see Figure 1).

Initially, EAI technology climbed the Peak of Inflated Expectations, overhyped by analysts and vendors alike as the next new thing. As we've discussed, challenges in early business integration deployments resulted in disappointing ROI for some projects. Others predicted that new technologies like Web Services would make EAI obsolete. (Personally, I believe Web services and EAI are complementary rather than competing technologies, as I discussed here last May [***WSJ***, Vol. 2, issue 5).) This encouraged doomsayers to push EAI into the Trough of Disillusionment. Yet increasing demand for intra- and interenterprise integration validates the merits of the EAI approach, as evidenced by efforts to standardize integration-related services and the race to embrace business integration by leading application and platform vendors. Today, many customers across a broad range of industries are steadily climbing the Slope of Enlightenment using EAI. They have more realistic expectations and the benefits of lessons learned on how (and how not) to implement this powerful technology. Thus, the future for EAI technology (if not for certain vendors) is still pretty bright. It should only be a matter of time until we reach the Plateau of Productivity, in which real-world benefits of business integration are broadly demonstrated and accepted.

Written by Jim Webber & Mark Little

Introducing WS-Coordination

A big step toward a new standard

In July 2002, BEA, IBM, and Microsoft released a trio of specifications designed to support business transactions over Web services. These specifications – BPEL4WS, WS-Transaction, and WS-Coordination – together form the bedrock for reliably choreographing Web services–based applications, providing business process management, transactional integrity, and generic coordination facilities respectively.

This article introduces the underlying concepts of Web Services Coordination, and shows how a generic coordination framework can be used to provide the foundations for higher-level business processes. In future articles, we will demonstrate how coordination allows us to move up the Web services stack to encompass WS-Transaction and on to BPEL4WS.

Coordination

In general terms, coordination is the act of one entity (known as the coordinator) disseminating information to a number of participants for some domain-specific reason. This reason could be in order to reach consensus on a decision like a distributed transaction protocol, or simply to guarantee that all participants obtain a specific message, as occurs in a reliable multicast environment. When parties are being coordinated, information known as the *coordination context* is propagated to tie together operations that are logically part of the same coordinated work or activity. This context information may flow with normal application messages, or may be an explicit part of a message exchange and is specific to the type of coordination being performed. For example, a security coordination service will propagate differently formed contexts than a transaction coordinator.

Despite the fact that there are many types of distributed applications that require coordination, it will be no surprise to learn that each domain typically uses a different coordination protocol. In transactions, for example, OASIS Business Transactions Protocol and Object Management Group's Object Transaction Service are solutions to specific problem domains that are not applicable to others since they are based on different architectural styles.

Given the domain-specific nature of these protocols (i.e., loosely coupled transactional Web services and tightly coupled transactional CORBA objects) there is no way to provide a universal solution without jeopardizing efficiency and scalability in each individual domain; and not everyone wants to (or can afford to) have a full-blown transaction processing system in order to do coordination. However, both of these protocols have the underlying requirement for propagating contextual information to participants, and therefore it would make some sense if that mechanism could be made generic, and thus reused. On closer examination, we find that even solely within the Web services domain we encounter situations where coordination is a require-

AUTHOR BIO:

Dr. Jim Webber is an architect at Arjuna Technologies, a spin-off from Hewlett-Packard that develops transaction technologies for J2EE and Web services. He was the lead developer on the world's first BTP-based transaction-ing system at Hewlett-Packard and continues to work with Web services transaction technology (like BTP and WS-Transaction). Jim's current research focuses primarily on Web services and Grid technology. He holds a his Ph.D. from the University of Newcastle upon Tyne, UK.
JIM.WEBBER@ARJUNA.COM

Dr. Mark Little is chief architect for Arjuna Technologies Limited, a spin-off from Hewlett-Packard that develops transaction technologies for J2EE and Web services. Previously, Mark was a distinguished engineer and architect at HP Middleware, where he led the transactions teams. He is a member of the expert group for JSR 95, and JSR 117, is the specification lead for JSR 156, and is active on the OTS Revision Task Force and the OASIS Business Transactions Protocol specification. Mark is the coauthor of J2EE 1.4 Bible and Java Transactions for Architects.
MARK.LITTLE@ARJUNA.COM

ment of several different types of problem domain, such as workflow management and transaction processing, but where the overall models are very different yet that same requirement for coordination is still present.

WS-Coordination

The fundamental idea underpinning WS-Coordination is that there is indeed a generic need for propagating context information in a Web services environment, which is a shared requirement irrespective of the applications being executed. The WS-Coordination specification defines a framework that allows different coordination protocols to be plugged in to coordinate work between clients, services, and participants (see Figure 1).

The WS-Coordination specification talks in terms of activities, which are distributed units of work involving one or more parties (which may be services, components, or even objects). At this level, an activity is minimally specified and is simply created, made to run, and then completed.

In Figure 1, we suggest that the framework is useful for propagating security, workflow, or replication contexts, though this is by no means an exhaustive list. Nonetheless, whatever coordination protocol is used, and in whatever domain it is deployed, the same generic requirements are present:

- ***Instantiation (or activation) of a new coordinator for the specific coordination protocol*** for a particular application instance
- ***Registration of participants with the coordinator*** such that they will receive that coordinator's protocol messages during (some part of) the application's lifetime
- ***Propagation of contextual information*** between the Web services that comprise the application
- ***An entity to drive the coordination protocol*** through to completion

The first three points are directly the concern of WS-Coordination, while the fourth is the responsibility of a third-party entity, usually the client application that controls the application as a whole. These four roles and their interrelationships are shown in Figure 2.

Activation

The WS-Coordination framework exposes an Activation Service that supports the creation of coordinators for specific protocols and their associated contexts. The process of invoking an activation service is done asynchronously, so the specification defines both the interface of the activation service itself, and that of the invoking service, so that the activation service can call back to deliver the results of the activation – namely a context that identifies the protocol type and coordinator location. These interfaces are presented in Listing 1, where the activation service has a one-way operation that expects to receive a CreateCoordinationContext message, and correspondingly the service that sent the CreateCoordinationContext message expects to be called back with a CreateCoordinationContextResponse message, or informed of a problem via an Error message.

Registration

Once a coordinator has been instantiated and a corresponding context created by the activation service, a Registration Service is created and exposed. This service allows participants to register to receive protocol messages associated with a particular coordinator. Like the activation service, the registration service assumes asynchronous communication and so specifies WSDL for both registration service and registration requester (see Listing 2).

When a participant is registered with a coordinator through the registration service, it receives messages that the coordi-

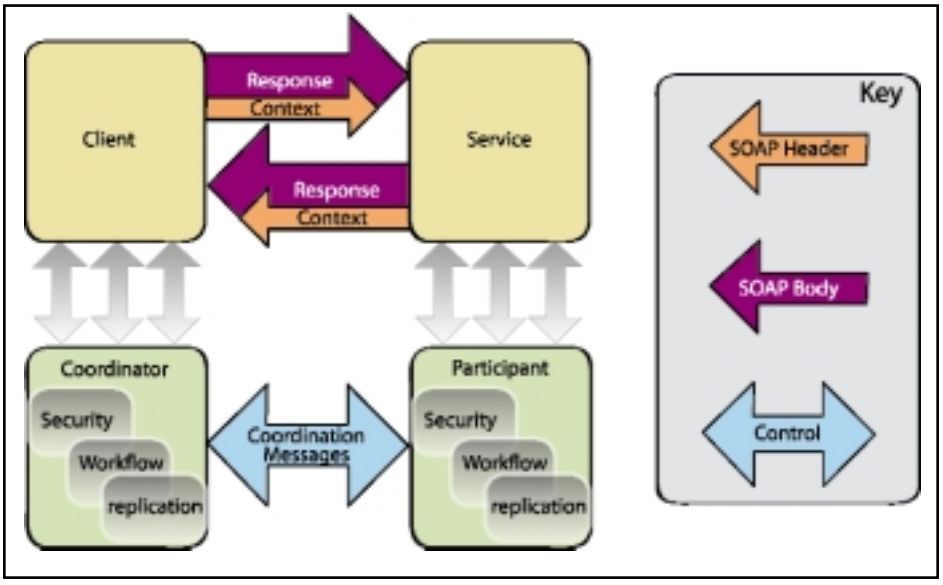


FIGURE 1 | **WS-Coordination architecture**

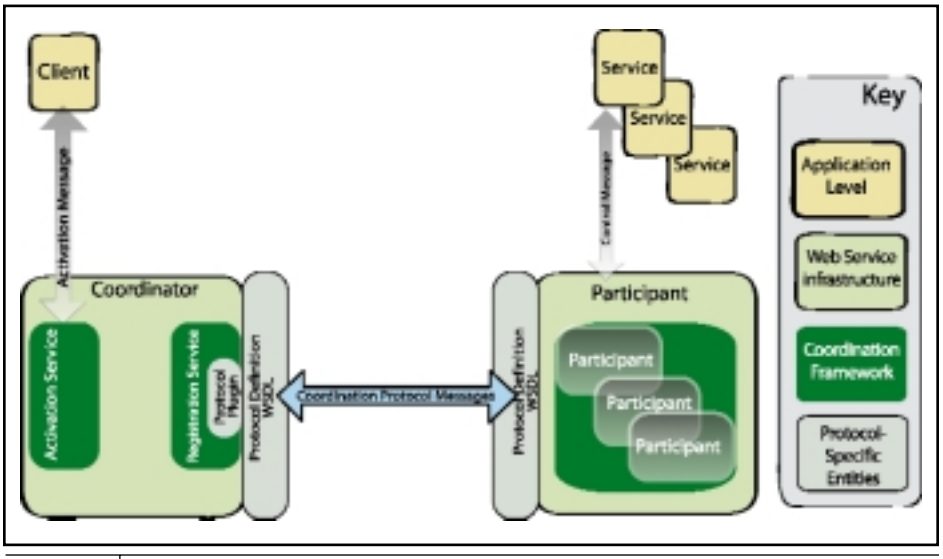


FIGURE 2 | **WS-Coordination infrastructure**

nator sends (for example, "prepare to complete" and "complete" messages if a two-phase protocol is used); where the coordinator's protocol supports it, participants can also send messages back to the coordinator.

Completion

The role of terminator is generally played by the client application, which at an appropriate point will ask the coordinator to perform its particular coordination function with any registered participants – to drive the protocol through to its completion. On completion, the client application may be informed of an outcome for the activity, which may vary from simple succeeded/failed notification through to complex structured data detailing the activity's status.

Context

The context is critical to coordination since it contains the information necessary for services to participate in the protocol. It provides the glue to bind all of the application's constituent Web services together into a single coordinated application whole. Since WS-Coordination is a generic coordination framework, contexts have to be tailored to meet the needs of specific coordination protocols that are plugged into the framework. The format of a WS-Coordination context is specifically designed to be third-party extensible and its contents are as follows:

- ***A coordination identifier with guaranteed global uniqueness*** for an individual coordinator in the form of a URI
- ***An address of a registration service endpoint*** where parties receiving a context can register participants into the protocol
- ***A time-to-live value*** that indicates for how long the context should be considered valid
- ***Extensible protocol-specific information*** particular to the actual coordination protocol supported by the coordinator

While the first three points are common sense, the fourth is somewhat more interesting. Since WS-Coordination is generic, it is of very little use to applications without augmentation, and this is reflected in the part of the WS-Coordination XML schema for contexts. In Listing 3, the schema states that a context consists of a URI that uniquely identifies the type of coordination that is required (xs:anyURI), an endpoint where participants to be coordinated can be registered (wsu:PortReferenceType), and an extensibility element designed to carry specific coordination protocol context payload (xs:any), which can carry arbitrary XML payload. (*Note:* This type also inherits some useful features from its parent in the form of a time-to-live value and an identifier.)

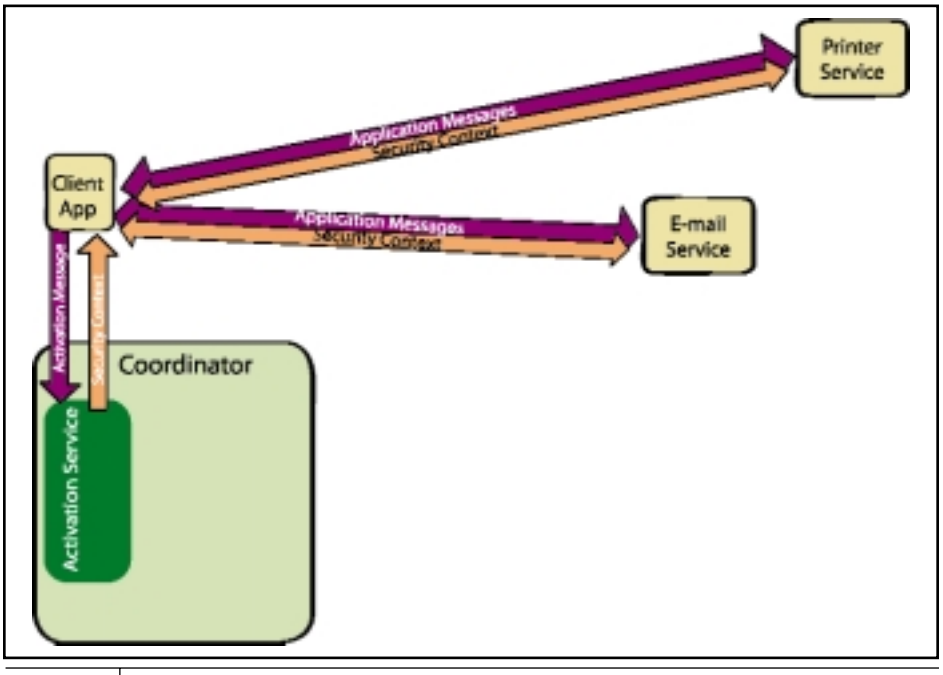


FIGURE 3 | Security coordination architecture: activation and application activities

Coordinating Business Processes on the Web

To show WS-Coordination in action, we'll consider a centralized sign-on service that enables a client application to authenticate once, and then use given credentials to access a number of Web services, and to de-authenticate from the system with a single operation irrespective of the number of Web services that are invoked. (*Note:* It's important to note that although the coordination strategy outlined here is reasonable enough, the patter as a whole isn't industrial strength since we avoid clouding the coordination issues by drawing on other useful technologies such as XML encryption and XML signature, which a truly trustworthy implementation would utilize. You should remember while following this example through that a real implementation would draw heavily on security standards like XML-encryption to provide the necessary privacy and XML digital signatures to provide authenticity.) The initial coordination pattern for this scenario is captured in Figure 3.

Here we see the initial stages of the application. The client application locates an activation service and sends it a message asking for the creation of a security coordinator and a corresponding security context, passing appropriate user credentials as part of the activation process as shown in Listing 4.

Assuming that a security coordination service has been registered with the coordination framework, a coordinator is created (and exposed as a registration service) and a context like that in Listing 4 is duly returned to the client application as part of the CreateCoordinationContextResponse message.

The client application interacts with its component Web services sending and receiving messages as normal, with the exception that it embeds the coordination context (which carries the security information) in a SOAP header block in its messages to provide authenticity credentials for those services that are invoked.

Let's assume that a service understands the protocol messages associated with our simple centralized sign-on service, and furthermore hasn't registered a participant previously. Once the service receives a SOAP message containing a security context header (see Listing 5), it registers a participant with the coordinator using the details

provided in the context (via the WS-Coordination registration service URI, for example). This registration operation occurs every time a service receives a particular context for the first time, which ensures that all services register participants within the activity.

When the client decides to terminate its session and log out of the services it has been using, it sends a completion message to the coordinator; in turn, the coordinator informs each registered participant to revoke the privileges for the client application, preventing it from using their corresponding services. Any subsequent calls by the client to that service with the same context will result in the service being unable to register a participant since the context details will no longer resolve to a live coordinator to register with (see Figure 4).

At some point, the client application finishes its work and must run the completion protocol to force its own system-wide logoff. To do this, it sends a security protocol logoff message to the security coordinator. This message is entirely out-of-scope of WS-Coordination and is instead defined by the specification of our security protocol which plugs in to the WS-Coordination framework. The completion message is shown in Listing 6.

In response to receiving this message, the security coordinator informs each of its registered participants to terminate the user's current session. To do this, it sends each of the participants a signOut message to which they respond with a signedOut message, confirming that the user is no longer authenticated with that particular participant's associated service. The pertinent parts of the signOut and signedOut messages are shown in Listing 7.

Once a signedOut message has been received from each of the enrolled participants, it can report back to the client application that its session has been ended. The final message in our WS-Coordination protocol is the loggedOut response message from the security coordination to the client (see Listing 8).

Advanced Usage Scenarios

In our security coordination example, the overall architecture is relatively static and known in advance of the coordination. However, it may be that in a business-to-business scenario we would like the ability to coordinate arbitrary groups of Web services as part of a single, logical, coordinated application. WS-Coordination supports this through a scheme known as *interposition*.

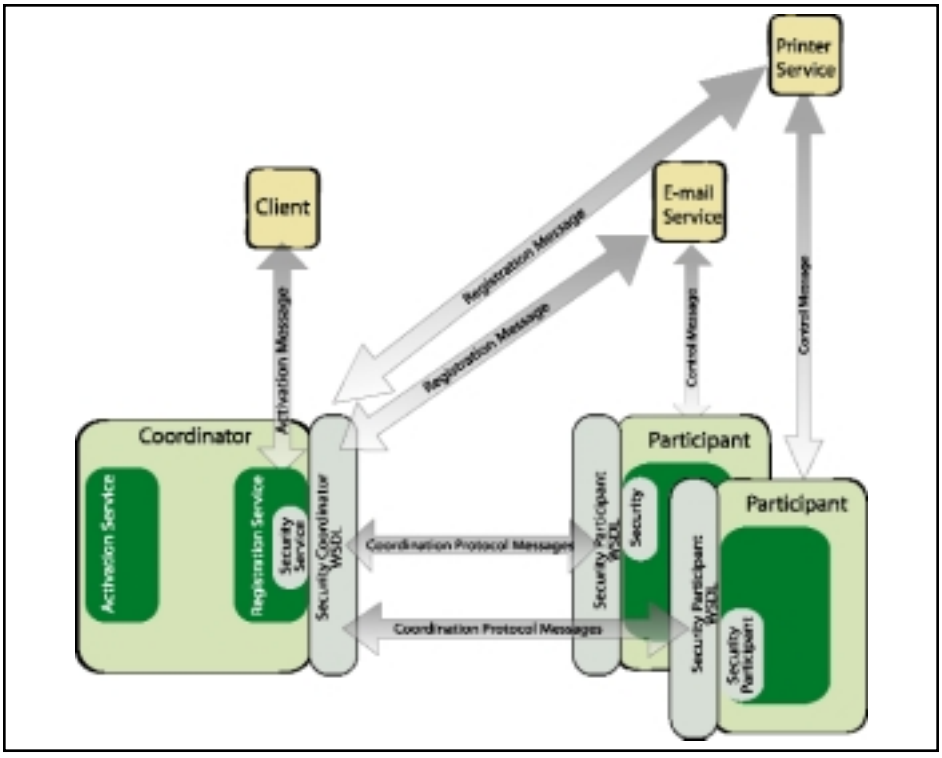


FIGURE 4 | **Security coordination architecture: registration and coordination**

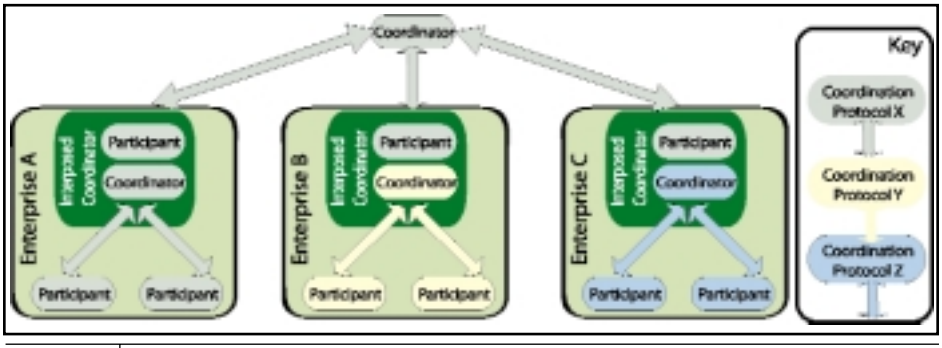


FIGURE 5 | **B2B interposed coordination pattern**

Interposition is a way of creating a hierarchy of coordinators, each of which looks like a simple participant to coordinators higher up the coordination tree, yet acts just like a normal coordinator for participants lower down the tree. Coordinators become registered within this hierarchy if the client application sends a CreateCoordination Context message to an activation service along with a valid context. When the receiving activation service creates the new context (and associated coordinator/registration service), it uses the original context to determine the endpoint of its superior coordinator (a.k.a. registration service) and enrolls the new coordinator with it.

In Figure 5 we see a typical interposed coordinator arrangement spanning three different enterprises using two different coordination protocols. This arrangement is arrived at through a client application creating a top-level context and then invoking Web services within the bounds of its partner enterprises. In the noninterposed case, upon first receipt of a context embedded in a SOAP header a service registers a participant with the coordinator identified by the context. However, in this situation, for reasons such as security or trustworthiness, the service enrolls its own coordinator by sending an activation message loaded with the top-level context to a local activation service, and then registers with the newly created local coordinator. (*Note:* By using ts own coordinator, the service or domain in which it resides only exposes the coordina-

tor to the superior and not the individual participants. This may be useful in restricting the amount of information that can flow out of the domain and hence be available to potentially upotentially unsecure or untrusted individuals/services.)

Having received a context with an activation message, the newly created coordinator duly registers itself with the registration service that the context advertises. The top-level coordinator is unaware of this arrangement since it sees the interposed coordinator as a participant, while the local participants are coordinated by their own local coordinator, which confers the following advantages:

- ***Increased performance:*** Since only completion messages need to be propagated over the Internet to the top-level coordinator, the more numerous coordination protocol messages remain on low-latency, high-bandwidth networking within the enterprise.
- ***Flexible coordination:*** Since the coordination within the enterprise is not visible to parties outside, the interposed coordinator can use whatever coordination protocol is most suitable for the type of application being executed within the enterprise. This may or may not be the same coordination protocol as that used at the top level, and so interposed coordinators can be used as a kind of "bridge" between coordination domains.

In Figure 5 Enterprise A uses the same coordination protocol as the top-level coordinator. In this case, Enterprise A's coordinator coordinates local participants according to the same protocol as the top-level. However, since only the outcome of the local coordination needs to be sent over the Internet to the top-level coordinator, and not the more abundant coordination protocol messages, this approach is performance-optimized, compared to registering Enterprise A's participants directly with the top-level coordinator.

For Enterprise B and Enterprise C, the same performance benefit exists, although the real focus of coordinating these participants is the fact that they are coordinated with different protocols that suit the particular enterprise's needs and not necessarily the same coordination protocol used at the top level. Since the local coordinator for each enterprise is effectively "bilingual" in the coordination protocols they understand (knowing both the participant aspects of the top-level coordination protocol and the coordinator aspects of their own internal coordination protocols), different coordination domains can easily be bridged without adding complexity to the overall architecture.

Summary

WS-Coordination looks set to become the adopted standard for activity coordination on the Web. Out of the box, WS-Coordination provides only activity and registration services, and is extended through protocol plug-ins that provide domain-specific coordination facilities. In addition to its generic nature, the WS-Coordination model also scales efficiently via interposed coordination, which allows arbitrary collections of Web services to coordinate their operation in a straightforward and scalable manner.

Though WS-Coordination is generically useful, at the time of this writing only one protocol that leverages WS-Coordination has been made public: WS-Transaction We'll look at this protocol in our next article.

Listing 1: Activation Service WSDL Interfaces

```
<!-- Activation Service portType Declaration -->
<wsdl:portType name="ActivationCoordinatorPortType">
  <wsdl:operation name="CreateCoordinationContext">
    <wsdl:input
     message="wscoor:CreateCoordinationContext"/>
  </wsdl:operation>
</wsdl:portType>

<!-- Activation Requester portType Declaration -->
<wsdl:portType name="ActivationRequesterPortType">
  <wsdl:operation
    name="CreateCoordinationContextResponse">
    <wsdl:input
     message="wscoor:CreateCoordinationContextResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Error">
    <wsdl:input message="wscoor:Error"/>
  </wsdl:operation>
</wsdl:portType>
```

Listing 2: Registration Services WSDL Interface

```
<!-- Registration Service portType Declaration -->
<wsdl:portType name="RegistrationCoordinatorPortType">
  <wsdl:operation name="Register">
      <wsdl:input message="wscoor:Register"/>
  </wsdl:operation>
</wsdl:portType>

<!-- Registration Requester portType Declaration -->
<wsdl:portType name="RegistrationRequesterPortType">
  <wsdl:operation name="RegisterResponse">
      <wsdl:input message="wscoor:RegisterResponse"/>
  </wsdl:operation>
  <wsdl:operation name="Error">
      <wsdl:input message="wscoor:Error"/>
  </wsdl:operation>
</wsdl:portType>
```

Listing 3: WS-Coordination Context Schema Fragment

```
<xs:complexType name="CoordinationContextType"
    abstract="false">
  <xs:complexContent>
    <xs:extension base="wsu:ContextType">
      <xs:sequence>
        <xs:element name="CoordinationType"
            type="xs:anyURI" />
        <xs:element name="RegistrationService"
            type="wsu:PortReferenceType" />
        <xs:any namespace="##any" processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Listing 4: Security Coordination Activation Message

```
<soap:Envelope
    xmlns:soap="http://www.w3.org/2002/06/soap-envelope">
  <soap:Body>
```

```
    <CreateCoordinationContext
    xmlns="http://schemas.xmlsoap.org/ws/2002/08/wscoor"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
    xmlns:security="http://security.example.org/authentication">
      <ActivationService>
        <wsu:Address>
          http://activation.example.org
        </wsu:Address>
        <security:loginID>
           a.n@other.com
        </security:loginID>
        <security:password>
          foobar
        </security:password>
      </ActivationService>
      <RequesterReference>
        <wsu:Address>
          http://workstation.example.org/station101
        </wsu:Address>
      </RequesterReference>
      <CoordinationType>
        http://security.example.org/single-logon
      </CoordinationType>
    </CreateCoordinationContext>
  </soap:Body>
</soap:Envelope>
```

Listing 5: A Security Context

```
<soap:Envelope
    xmlns:soap="http://www.w3.org/2002/06/soap-envelope">
  <soap:Body>
    <CreateCoordinationContextResponse
    xmlns="http://schemas.xmlsoap.org/ws/2002/08/wscoor"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
    xmlns:security="http://security.example.org/authentication">
      <RequesterReference>
        <wsu:Address>
          http://workstation.example.org/station101
        </wsu:Address>
      </RequesterReference>
      <CoordinationContext>
        <CoordinationType>
          http://security.example.org/single-logon
        </CoordinationType>
        <RegistrationService>
          <wsu:Address>
            http://security.example.org/registration
          </wsu:Address>
          <security:token>
            1234-5678-9ABC-DEF0
          </security:token>
        </RegistrationService>
      </CoordinationContext>
    </CreateCoordinationContextResponse>
  </soap:Body>
</soap:Envelope>
```

Listing 6: Security Protocol Completion Request Message

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2002/06/soap-envelope"
  xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/08/wscoor"
  xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
  xmlns:sec="http://security.example.org/authentication"
  xmlns:comp="http://security.example.org/authentication/
      completion">
  <soap:Body>
    <comp:logout>
      <wscoor:RequesterReference>
        <wsu:Address>
          http://workstation.example.org/station101
        </wsu:Address>
      </wscoor:RequesterReference>
      <sec:loginID>
        a.n@other.com
      </sec:loginID>
      <sec:password>
        foobar
      </sec:password>
    </comp:logout>
  </soap:Body>
</soap:Envelope>
```

Listing 7: signOut and signedOut Message Contents

```
<!--The signOut message contents -->
<comp:signOut>
  <wscoor:RequesterReference>
    <wsu:Address>
      http://security.example.org/logout-service
     </wsu:Address>
  </wscoor:RequesterReference>
  <sec:loginID>
    a.n@other.com
  </sec:loginID>
</comp:signOut>

<!--The signedOut message contents -->
<comp:signedOut>
  <wscoor:RequesterReference>
    <wsu:Address>
      http://printer.example.org/logout-service
    </wsu:Address>
  </wscoor:RequesterReference>
  <sec:loginID>
    a.n@other.com
  </sec:loginID>
</comp:signOut>
```

Listing 8: The loggedOut Message Indicates Session Termination

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2002/06/soap-envelope"
  xmlns:wscoor="http://schemas.xmlsoap.org/ws/2002/08/wscoor"
  xmlns:wsu="http://schemas.xmlsoap.org/ws/2002/07/utility"
  xmlns:sec="http://security.example.org/authentication"
  xmlns:comp="http://security.example.org/authentication/
      completion">
  <soap:Body>
    <comp:loggedout>
      <wscoor:RequesterReference>
        <wsu:Address>
          http://security.example.org/authentication
        </wsu:Address>
      </wscoor:RequesterReference>
      <sec:loginID>
        a.n@other.com
      </sec:loginID>
    </comp:loggedout>
  </soap:Body>
</soap:Envelope>
```

Download the code at

sys-con.com/webservices

EXPOSING LEGACY APPLICATIONS

Written by Adelene Ng

AN APACHE SOAP FRAMEWORK THAT PROVIDES AN EXCELLENT IMPLEMENTATION

SOAP (Simple Object Access Protocol) is a wire protocol that is similar to CORBA's Internet Inter-ORB protocol (IIOP) for communicating between applications running on different operating systems, with different technologies and programming languages. Unlike IIOP, which is binary in nature, SOAP is text based. This XML-based protocol lets you call an application, or even an individual object or method within an application, across the Internet via HTTP. HTTP does not pose any compatibility and security problems, unlike RPC, since all Internet browsers and servers support HTTP.

Typical applications for SOAP include:

- ***Business-to-Business Integration (B2Bi):*** Businesses can develop their applications and make these available to other companies via SOAP.
- ***Distributed applications:*** SOAP allows applications to be accessed and remotely managed via the Internet.
- ***Web applications:*** This typically allows a user to query a Web server through a Web browser and results are displayed to the user through the Web page. SOAP, on the other hand, allows a user to query the Web server and run a program on it through a SOAP client running on the user's computer.

AUTHOR BIO:

Adelene Ng is a software architect with Xerox Corporation based in Rochester, NY. She holds a Ph.D. in computer science from the University of London, and an M.Sc. from the University of Manchester.

ADELENE.NG@XSSC.SGP.XEROX.COM

In this article, I propose a solution using SOAP that allows legacy client methods to be called from a SOAP client. A SOAP client is just like any program that can be run from the user's computer. Even though other solutions are possible, for example using XML-RPC, SOAP offers the following advantages:

- Support for a richer set of data types
- Allowing users to specify a particular encoding
- Namespace aware
- Asynchronous
- Support for request routing and preprocessing of requests via SOAP headers

System Architecture

I've used a three-tier architecture in this implementation. Figure 1 shows the overall system architecture.

The first tier is the SOAP client, which accesses the legacy application through SOAP. The legacy application resides in another client/server application. The legacy client resides on the Web server – the middle tier. A wrapper layer is provided around the legacy client application. This allows for the clean separation of legacy calls from the wrapper class, and it

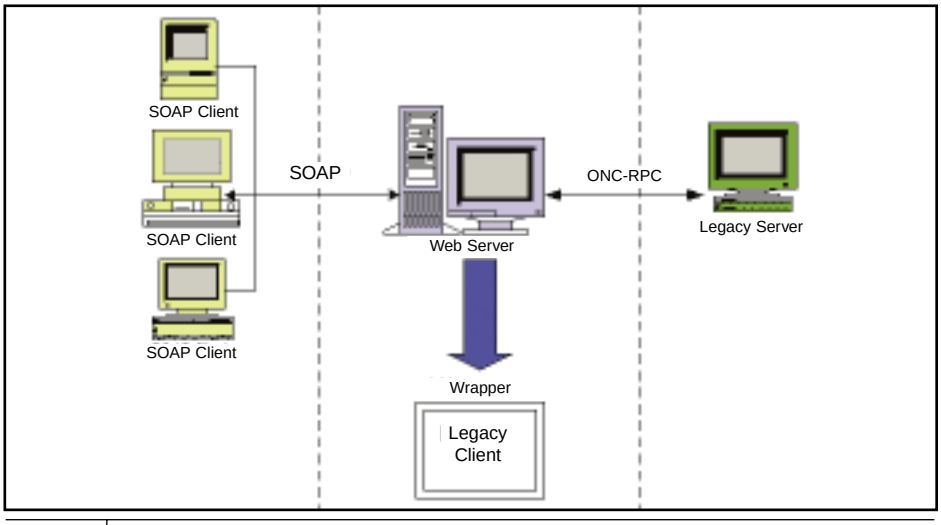


FIGURE 1 | System architecture

isolates the legacy code to a single location, which makes for easy maintenance. The SOAP client accesses the legacy client application on the Web server through the wrapper class. The legacy client, in turn, communicates with the legacy server through ONC-RPC. The legacy server may reside on a different machine and is the third tier in our architecture.

Tools

The entire system was written in Java running on Win 2000 Professional. The tools used to build this system include:

- ***Java 2 Platform, Standard Edition (J2SE):*** http://java.sun.com/j2se
- ***ONC RPC for Java:*** A commercial ONC-RPC package: www.distinct.com
- ***Apache SOAP 2.3.1:*** http://xml.apache.org/soap/index.html
- ***JavaMail 1.3 (used for the SMTP transfer protocol support included in Apache SOAP):*** http://java.sun.com/products/java mail
- ***JavaBeans Activation Framework 1.0.2:*** http://java.sun.com/products/javabeans/glasgow/jaf.html
- ***Java XML Pack Summer '02 Update Release:*** http://java.sun.com/xml/downloads/javaxmlpack.html. We are interested only in the JAXP for XML Processing Package.
- ***Apache Tomcat 3.2.4 (Servlet Engine):*** http://jakarta.apache.org/tomcat/index.html

Why SOAP?

SOAP was selected over other mechanisms such as XML-RPC because:

1. It's a simple protocol to use.
2. The SOAP protocol is lightweight, incurring a minimal amount of overhead.
3. It runs over HTTP, making it easy to integrate SOAP-based applications with other Web-based applications.
4. It provides a better error handling mechanism through the SOAP Fault object.
5. It supports a variety of encodings that are governed by rules that determine how application-defined data type instances are exchanged.
6. Ease of handling of custom parameter types: Data exchange takes place via parameters passed to the SOAP remote service; and getting back a response from the remote service.

Developing a SOAP Application

Now let's outline the steps necessary to develop a SOAP application:

Define a service class that is having its methods invoked remotely.

Define a deployment descriptor that tells the SOAP server:

- ***The URN of the SOAP service accessible to clients***
- ***The list of methods available to the client(s):*** If the arguments and return types of these methods are simple types, then the deployment descriptor in suffices. If the arguments and return types of these methods are user-defined custom parameters, then the deployment descriptor will have to be augmented with a "mappings" element to tell the SOAP server how to handle custom parameters. The "mappings" element defines the serializer and deserializer to use. If your custom parameters are in the JavaBean format, then the BeanSerializer class provided by the Apache SOAP implementation will be used for the serialization and deserialization. In the example presented here (see Figure 3), the serializer and deserializer classes are part of the Apache SOAP framework, org.apache, soap.encoding.soapenc. BeanSerializer.
- ***Serialization/deserialization handlers for custom classes (if any):*** This is required when the user-defined parameter is not a simple data type that can be handled by the Apache SOAP implementation nor does it conform to the JavaBean format. In this case, you will have to write your own serializer and deserializer. Any user-defined serializer and deserializer will have to implement the org.apache.soap.util.xml.Seria lizer and org.apache. soap. util.xml.De serializer interfaces. Since user-defined serialization and deserialization is beyond the scope of this article, this technique will not be discussed here.

The deployment descriptor is an XML file. In this example, it is called Accounting Service.xml. The contents are shown in Figure 2.

Before deploying the service, make sure that your Web server is up and running. Here we are using the Apache Tomcat Server, which has several scripts under the C:\jakarta-tom-

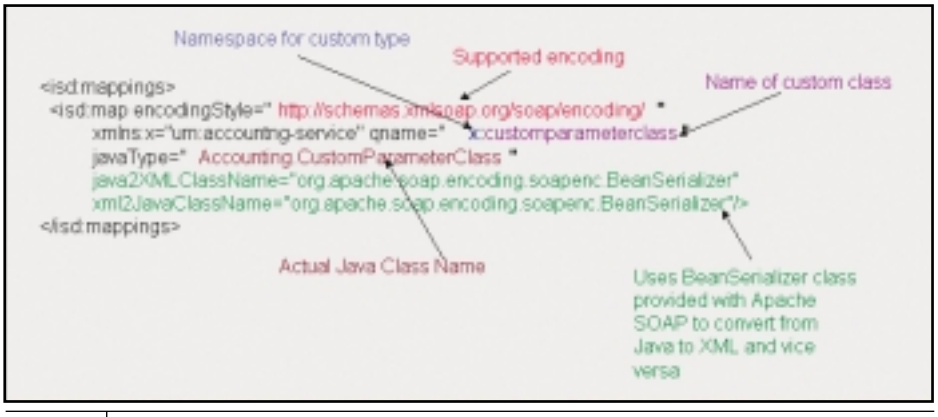


FIGURE 2 | **AccountingService.xml – Deployment Descriptor**

FIGURE 3 | **Addition of "Mappings" Element to handle Custom parameters**

cat-3.2.4\bin directory. To start the Apache Tomcat Server, go to the directory, C:\jakarta-tomcat-3.2.4\bin and type startup. Figure 4 shows the Tomcat 3.2 Console Window when the Tomcat Server is started.

Deploy the Service.

- Make your service class visible to your SOAP server. If you are using the Apache SOAP server, this service class is "Jarred" up and copied into the \lib directory.
- Restart your servlet engine.
- Your service class can now be deployed by using the utility class, org.apache.soap.Server.ServceManager from Apache Soap. Deploy it by running the following command from the command line,

```
C:\> java ServiceManagerClient
http://localhost:8080/soap/serverlet/rpcr
outer deploy AccountingService.xml
```

Check that your service has indeed been added via

```
C:\> java ServiceManagerClient
http://localhost:8080/soap/serverlet/rpcr
outer list
```

This command should show the service that you have just added plus all the current services available on that server. Figure 5 shows the messages that are displayed when this command is run.

Run the legacy server application (if it is not currently running).

Run the SOAP client.

Once you are done with the service, you can undeploy it by running

```
C:\> java ServiceManagerClient
http://localhost:8080/soap/serverlet/rpcr
outer undeploy urn: accounting-service
```

To make things simple, I have created batch files for the last five steps to facilitate the execution of these commands.

Developing a SOAP Client

For the purposes of this article, the SOAP client is invoked from the command line. The basic boilerplate steps involved in every SOAP call are:

1. Define the type mapping for the custom parameters. This is required only if the application uses custom parameters.
2. Build the call object.
3. Create a parameter list (for your method that you will be invoking).
4. Add the parameter list to the Call object.
5. Invoke it.
6. Error Handling.

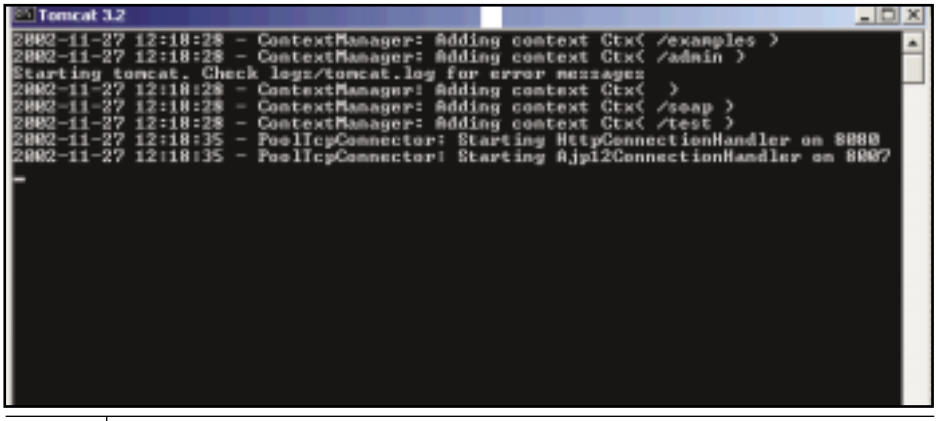


FIGURE 4 | **Tomcat 3.2 Console Window**

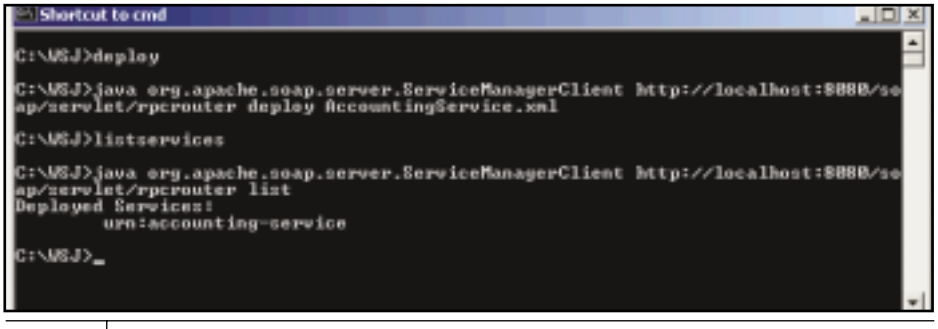


FIGURE 5 | **Listing of all services deployed**

Creating the SOAP Mapping Registry and Defining the Mapping Types

The following steps are required only if the application uses a custom or user-defined object and the user-defined object follows the JavaBean convention.

1. Create a user-defined object.
2. Create the SOAPMappingRegistry and Bean-Serializer objects.

```
SOAPMappingRegistry reg = new
SOAPMappingRegistry();
BeanSerializer b_serializer = new
BeanSerializer();
```

3, A QName object is also created to represent the user-defined type.

```
QName q_name = new QName("urn:accounting-
service", "x.y.z.UserDefinedClass");
```

4. The type mapping is created by calling the method, mapTypes(), on the SOAPMappingsRegistry object. This method takes in as arguments the encoding style, the QName object, the JavaBean class, and serializer and deserializer objects.

```
reg.mapTypes(Constants.NS_URI_SOAP_ENC,
q_name, x.y.z.UserDefinedClass, b_serial-
izer, b_serializer);
```

Building a Call Object

The first step is to create a new Call object:

```
Call call = new Call();
```

If the SOAPMappingRegistry was created previously, we have to tell the call object to set the type mapping to the SOAP mapping registry via:

```
call.setSOAPMappingRegistry(reg);
```

Next, set the URI of the SOAP Service to use:

```
call.setTargetObjectURI("urn:accounting-
service");
```

Specify the method to call:

```
call.setMethodName("GetNumberOfRecords_W"
);
```

Set the encoding style to use:

```
call.setEncodingStyleURI(Constants.NS_URI
_SOAP_ENC);
```

Creating a Parameter List

A new Vector object is created to store the parameter list:

```
Vector params = new Vector();
```

For each parameter in the method, create a new Parameter object and add that to the params vector list:

```
params.addElement(new Parameter("file-
name", String.class, filename, null));
```

Finally, add the resulting Vector, params, to the Call object:

```
call.setParams(params);
```

Invoking the Call

The SOAP method is called:

```
Response response = call.invoke(url,"");
```

The Response object might contain the return value of the remote method that was invoked or the fault/error values as a result of the fault generated.

Error Handling

Errors are handled using the Fault object. The two main methods used are getFault String() and getDetailEntries(). The first method lets you know that an error has been encountered. However, it does not provide enough information to help you track down the root cause of the problem. The second method provides more details about the problem encountered. This call returns a vector of objects. Each element of this vector is a DOM element with detailed error information. The following code snippets show how the Fault object is used to obtain detailed error information.

Get the Fault object:

```
Fault fault = response.getFault();
```

Get the vector of DOM Elements:

```
Vector entries =
fault.getDetailEntries();
```

Iterate through this Vector and obtain the DOM elements for each entry:

```
for (Iterator i = entries.iterator();
i.hasNext(); ) {
org.w3c.dom.Element entry =
(org.w3c.dom.Element)i.next();
System.out.println(entry.getFirstChild().
getNodeValue());
}
```

Core Supporting Class and Methods

The core methods to support the SOAP Client are contained in the AccountingLog_SC class. These are:

- ***public void BuildDOMTree(ArrayList al):*** Takes in an ArrayList and builds the DOM tree structure.
- ***public void TransformIt(String outputfilename):*** Takes the DOM tree produced by BuildDOMTree() and produces the resulting XML file.

The Web Server

The Web Service

The service to be deployed on the Web server is a Java class that exposes a number of methods callable from the SOAP client. In a two-tier application, this is typically the end point. However, in this article we show how in a three-tier application, the service residing on the Web server acts as a client to a legacy application that resides on a different server. The legacy client code that resides on the Web service has a wrapper layer over it. This will facilitate a clean separation from the legacy calls. Any future enhancements and fixes made to the lega-

Listing 1

```
%
%/*
% * RPC protocol definition for the public interface to
% * Customer Accounting data
% *
% * Account.x
% */
#define EXPORT_PROTOCOL 0x20000094

enum status {
        failure = 0,
        success = 1
};

struct get_number_of_records_args {
        string filename<>;
};

struct get_number_of_records_rtn {
        status the_status;
        int number_of_records;
};

struct get_file_data_args {
        string filename<>;
};

struct string_array {
        string the_string<>;
};

struct get_file_data_rtn {
        status the_status;
        string_array records<>;
};


program EXPORT_PROG {
 version EXPORT_VERS {
        get_number_of_records_rtn GET_NUMBER_OF_RECORDS (get_number_of_records_args) = 1;
        get_file_data_rtn GET_FILE_DATA (get_file_data_args) = 2;
        } = 1;
} = EXPORT_PROTOCOL;
```

Download the code at sys-con.com/webservices

cy client application can be contained and isolated here.

The Wrapper Web Service

To expose the legacy client as a Web service, a wrapper class is built over the legacy client class. In this way, the legacy client code can be isolated. The two methods callable from the SOAP client are Get NumberOfRecords _W() and GetFileData _W(). Within each of these methods, calls are made to the legacy client methods.

The Legacy Client

In the legacy client implementation, calls are made to the stub methods, which in turn invoke the server functions. The underlying Distinct ONC/RPC runtime libraries and supporting framework handles all the marshalling/unmarshalling of data. These data handling details are transparent to the user; each of the calls made to the stub methods are placed in separate methods – GetNumberOf Records() and GetFile Data() respectively. Separating these calls into different method calls will make it easier to integrate with the wrapper class.

The Legacy Server and Supporting Classes

The Legacy Applications

The legacy application is a typical client/server application. It uses ONC-RPC as a mechanism of communication between the two entities. RPC calls made by the client behave as if they were invoking a local procedure call. All the marshalling and unmarshalling of data is taken care of by the underlying ONC-RPC framework.

The X IDL file definition for the legacy application defines the interface between the client and server through a language-neutral construct, similar to the CORBA IDL definition.

The X IDL file definition contains two interface calls. One returns the number of records in the file. The other returns the data records, defined as GET_NUMBER_OF_ RECORDS() and GET_FILE_DATA() respectively. Listing 1 shows the contents of the X file.

The Jrpcgen compiler translates the standard RPC/XDR (External Data Representation) interface definition files into Java classes that implement the client- and server-side stubs and the XDR conversions for the described interface. This means that Jrpcgen implements a certain Java language mapping for .x IDL files.

The Distinct ONC RPC/XDR for Java package contains the runtime libraries that conform to RFC 1831 and RFC 1832. The API consists of classes that allow you to write pure Java clients for standard RPC servers that can be embedded in applets and run by a standard Web browser. It also allows you to develop ONC RPC stand-alone servers. The package allows connections over TCP, UDP or encapsulated over HTTP.

Core Supporting Classes and Methods on the Legacy Server

The core methods required to support the legacy server are contained in the AccountingLog_LS class. The supporting methods are:

- ***public int GetNumberOfRecords_LS(String filename):*** Opens the file specified as the argument, parses it, and returns the number of records read.
- ***public ArrayList BuildRecords(String filename):*** Opens the file specified, and returns all the records read as an ArrayList.

The Legacy Server

The legacy server implements the interface methods defined in the X IDL file. It makes use of the underlying core methods defined in AccountingLog_LS class to support the required behavior.

In the server implementation of get_ number_of_records_1(), it calls the core method in the AccountingLog_LS class, GetNumberOf Records_LS(). This method opens the file specified in the argument list, counts the number of records, and returns the result to the caller.

Likewise, in the implementation of get_ file_data_1(), the core method, BuildRec ords(String filename), is called, again with the specified file as argument. The core method returns the records read as an array list (part of the Java Collections interface). Since XDR does not support Collections, the array list returned has to be converted into an array of strings, which is returned to the calling client.

Having the SOAP client invoke this call through the wrapper class results in the creation and storing of an XML file on the legacy server and may pose security problems. A better place to create the XML file is on the client end. After the records are read, the XML file can then be created and stored on the client.

"Chaining" and Interactions Between Tiers

"Chaining" can occur at the Web services or the legacy application level. In the former, the Web services on one Web server can act as a client accessing the Web services located on another Web server (see Figure

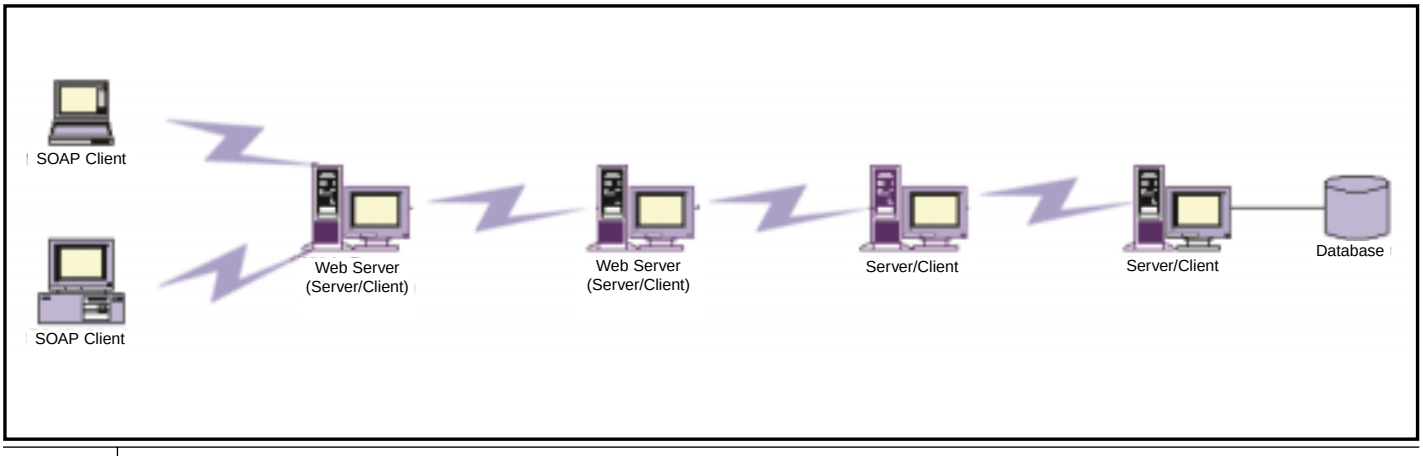


FIGURE 7 | **"Chaining" at the Web server tier**

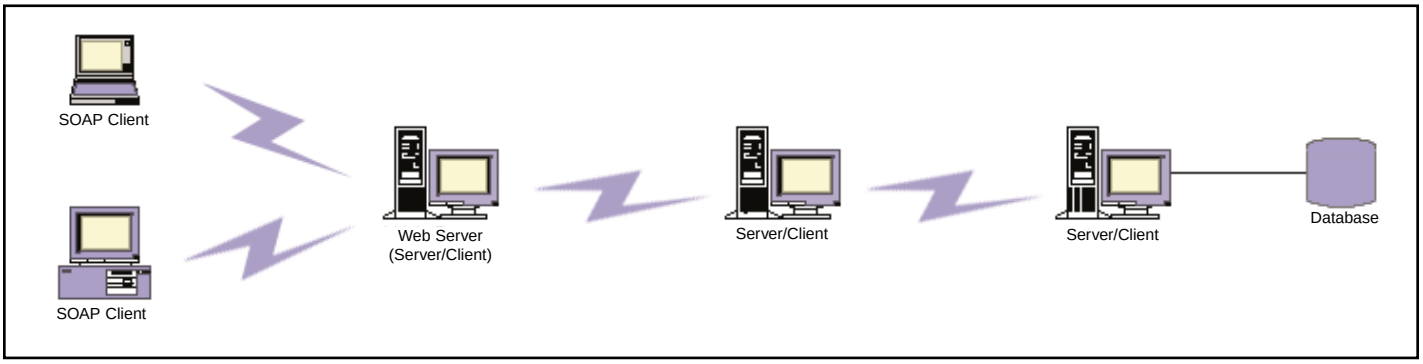


FIGURE 8 | **"Chaining" at the Legacy System Tier**

7). In Figure 8, the legacy client on the Web server accesses methods, objects, or even other applications on the legacy server. That legacy server in turn acts as a client to other servers on the system. For example, the legacy server could act as a database client querying the database server.

Flow Sequence

To show how the different pieces are tied together, Figures 9 and 10 illustrate the call sequence between the SOAP client, Web server (containing the wrapper class and legacy client), and legacy server. The sequence diagram summarizes the interactions between the different tiers and clearly shows which platform each of the classes reside on.

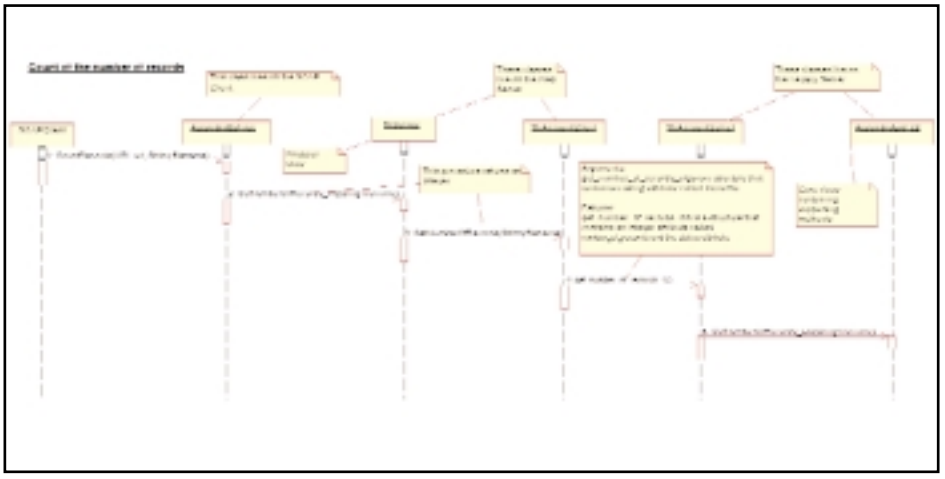


FIGURE 9 | **Sequence diagram for "Counting the Number of Records"**

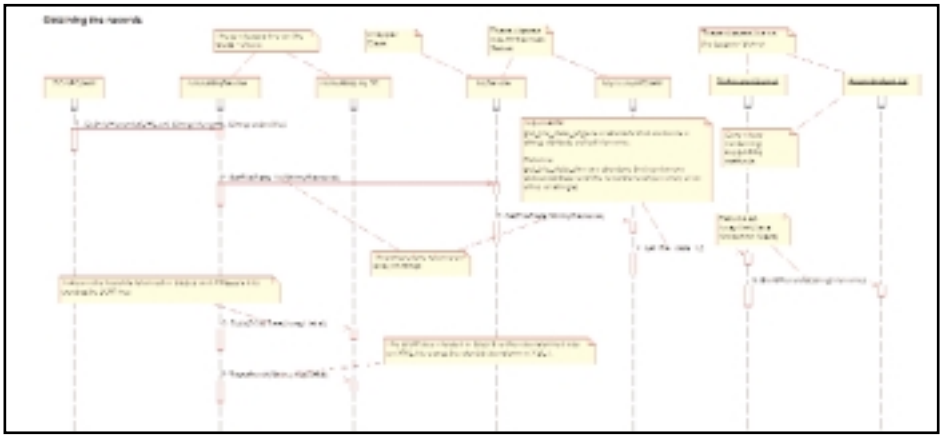


FIGURE 10 | **Sequence diagram for "Obtaining the Records"**

Future Enhancements

Let's look at some of the possible enhancements to the application:

- Providing better GUI support for the application.
- Ability to perform "database-like" queries on the XML file. This will utilize the currently emerging "XML Query" technology.
- Transforming the resultant XML file into other formats using XSLT.
- Using XSL-FO to produce formatted reports.

Conclusion

This article has shown how legacy applications can be exposed as a Web service and accessed through the SOAP client. I've also discussed the possibilities of "chaining" on both the Web server and legacy application tiers. The Apache SOAP implementation provides an excellent framework in which to develop SOAP applications. The underlying mechanism (Call object) makes all the SOAP communications appear to work seamlessly together. The user does not need to be exposed to the nitty-gritty details of SOAP in order make a SOAP call.

References

- Srinivasan, R. "RPC: Remote Procedure Call Protocol Specification Version 2, RFC 1831" (Aug 1995). www.ietf.org/rfc/rfc18 31.txt
- Srinivasan, R. "XDR: External Data Representation Standard, RFC 1832" (Aug. 1995). www.ietf.org/rfc/rfc1832.txt
- *SOAP Specification:* www.w3.org/TR/SO AP
- McLaughlink, *Java and XML*, 2nd Edition. O'Reilly
- Englander, (2001). *Java and SOAP*, O'Reilly.
- Gupta, Samudra. "Using SOAP and Java, Parts 1, 2 & 3". http://javaboutique.internet.com/tutorials/SOAP/index.html

Acknowledgements

I want to thank Tzu-Khiau Koh and Wen-Yun Chang for their invaluable feedback on the initial drafts of this article.

Reviewed by Joseph A. Mitchko

Microsoft UDDI SDK 2.0

Enabling .NET programmers to interact with UDDI services

Author Bio:
Joe Mitchko is a technology specialist working for a leading Internet service company and is product review editor and contributing writer for Web Services Journal. JMITCHKO@RCN.COM.

Microsoft

COMPANY INFO
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
Tel: (425) 882-8080
E-fax: (425) 706-7329
Web: www.microsoft.com

EVALUATION DOWNLOAD
http://msdn.microsoft.com/uddi

LICENSING INFORMATION: Free

TESTING
OS: Windows-2000 Professional
Hardware: IBM ThinkPad T30

One of the least appreciated standards in the Web services world is Universal Description, Discovery, and Integration – UDDI. The concepts of the UDDI standard are fairly simple – link Web service consumers to providers. If you are a provider, you want to advertise who you are, the nature of your services, and technical information regarding the types of interfaces you provide. For a consumer, you want to be able to quickly look up services according to what type of business you are interested in, the type of service, and other factors.

Like anything these days, what sounds simple gets rather complicated, and as a system developer you need to be an expert in various Web technologies, including HTTP and XML protocols. Luckily, we have development toolkits to make it easier for the average Joe (like me) to get the job done. This review will not only help you understand what the Microsoft UDDI SDK version 2.0 provides, but should also give you a head start to do your own interfacing with UDDI services using .NET.

The UDDI SDK provides all the tools necessary for a .NET developer to interact with a UDDI registry. The examples I'll use in this review will be coded in C#, but with the Command Language Runtime (CLR) architecture of .NET programming in other languages such as Visual Basic will be very similar. In fact, they will use the same assembly components across all languages. The same assembly can also be used from COM clients such as the languages in the Visual Studio 6.0 product.

Installation

Before you install, you'll need to do a quick check of the software prerequisites on your machine. To start, you need to have Microsoft Visual Studio .NET 2003 installed on your machine, along with the 1.1 version of the .NET framework.

The UDDI SDK version 2.0 is included in the Core section of the Microsoft Windows Software Development Kit and instructions for download are available from the UDDI section on Microsoft Developer Network (MSDN). Installation is straightforward: just be sure to include the "Core SDK" components and you should be done in a few minutes. The installation includes an extensive amount of documentation and coding examples, a wizard for publishing services from within Visual Studio .NET 2003, and sample applications that we will cover later. The actual framework code is in the Microsoft.Uddi.dll assembly, located in the bin directory of the Microsoft SDK installation. Run the UddiSdkRegister executable and the assembly will be registered and ready to use.

Finding a Business Partner

Locating a Web service in UDDI starts with finding your business partner in the UDDI repository. For this, the SDK provides the Microsoft.Uddi.FindBusiness object to help you do the search. The object accepts the addition of one or more entries to its "name" property representing the names of the business you are interested in (including wildcards), followed by a call to its Send() method. The call to the Send() method returns a BusinessList object, which may contain one or more business that match your criteria in the "Name" property. Of course, you may not get any back depending on what's listed in the UDDI repository.

Once you have the BusinessList object, you can quickly ascertain whether you had any businesses listed by checking the number of BusinessInfo objects in BusinessInfos collection using the "Count" property (see Listing 1). Now it's just a matter of iterating through each of the BusinessInfo objects to read the business description and other information about the business. One of the "key" data items is the "BusinessKey", which is a globally unique identifier (GUID) used by the UDDI registry to identify the business. UDDI also provides these keys for services, bindings, and a thing called a tModel. All of these keys can be used to work your way down from the top level of the repository (the business name) to the real technical descriptions of the service bindings. UDDI represents this information in a tree-like fashion, and the UDDI SDK is geared to help you navigate easily through the hierarchy to get the data that you need.

Other objects in the SDK help you look up Web services by the service type, binding, and tModel. You may be asking at this point, what is this "tModel" thing? In a nutshell, it's a unique signature defining the service at a very technical level (implementation, platform specific, etc.) that helps Web service consumers who are looking for an exact fit of a producer. It's really up to the business partners to define what that exact fit is but it often

equates to an interface definition expressed in the Web Service Description Language (WSDL).

Publishing Your Service

Okay, we've looked at the UDDI SDK's ability to help you look things up in UDDI; what about on the publishing side? Suppose you want your Web service implementation to automatically publish to the UDDI registry when installed, most likely providing the installer or administrator with an interface where they enter the UDDI publishing URL and authentication credentials. And once the system is in place and has its services properly published, you want to provide the capability for the system to maintain the information in the registry, even changing service binding or definitions on a data-driven process. The UDDI SDK has all you need to register your service down to the binding and tModel level. This time, instead of using the SDK objects to navigate through the hierarchy, you will use objects to build the hierarchy (or make modifications to it) level by level.

To begin, we'll need to configure the "UddiConnection" object with the URL of the UDDI server, and enter a user name and password that will be authenticated by the server. In most cases, the URL will use secure HTTP (https://) for the publishing activity. You don't want hackers in there publishing things that you as a business don't support (or worse). Next, you need to create a SaveBusiness object and set the various properties in the object about your business including name, description, and other vital data. Once you're done setting up your business information, you again make a call to the Send() method and wait for a response from the UDDI server (see Listing 2).

One thing I haven't mentioned so far is that the UDDI SDK has an extensive number of exception handling objects to help ascertain whether things are properly published or looked up. All you need to do as a C# developer is to surround the SDK code in a try block and wait to see whether any exceptions are caught during the operation. No catch means everything performed correctly.

Support

That's enough from the API side of things. Here are a few things to keep in mind when using the SDK. First, the UDDI SDK requires that you install the 1.1 version of .NET Framework and Visual Studio .NET 2003, both of which were in final beta at the time of this writing but these can run alongside previous versions.

In addition, the SDK is based on the 2.0 version of the UDDI specification, which is the primary version of the specification that is implemented by the various UDDI server vendors. The UDDI SDK provides a great opportunity to learn all about how UDDI works and offers you practical programming experience interfacing with a UDDI registry. The API set provided in the SDK is easy to pick up and learn. Also, you will find that almost all of the public UDDI registries out there implement the UDDI 2.0 specification, and they are available through the UDDI SDK examples that come with the package.

Finally, you will find several other Microsoft UDDI SDKs out there for download. There is another beta version out for .NET framework 1.0, and a further version for the UDDI 1.0 specification. These are all similar from the programming standpoint but not as comprehensive or as well documented as the version 2.0 release.

Conclusion

There you have it. There is a whole lot more that I didn't cover on the UDDI SDK version 2.0, but you should have enough of the basics from both a programming and UDDI specification standpoint to start you off. The UDDI specification may be the more obscure Web service specification, but nevertheless has enormous potential as a "yellow pages" of sorts for the global Web service world. The UDDI SDK provides a convenient and robust set of programming APIs to help you publish and browse though the book.

Listing 1: C# Code Example for Publishing a Web service

```
using Microsoft.Uddi;
using Microsoft.Uddi.Businesses;

// Add the following lines to the Main method:
try
{
    // Create a connection to the UDDI node that is to be
accessed.
    UddiConnection myConn = new UddiConnection(
  "http://test.uddi.microsoft.com/inquire",
        "https://test.uddi.microsoft.com/publish");

    // Create authentication credentials for save operation.
    myConn.AuthenticationMode =
AuthenticationMode.UddiAuthentication;
    myConn.Username = " *** insert your username *** ";
    myConn.Password = " *** insert your password *** ";

    // Create a named business entity.
    BusinessEntity myBiz = new BusinessEntity(" *** insert your
business name *** ");

    // Use business entity to create an object to save a busi-
ness.
    SaveBusiness sb = new SaveBusiness(myBiz);

    // Send the prepared save business request.
    BusinessDetail savedBiz = sb.Send(myConn);

    // Interpret the returned business detail to examine the
      allocated business key.
    Console.WriteLine("Business: " +
savedBiz.BusinessEntities[0].Names[0].Text);
    Console.WriteLine("  Allocated key: " +
savedBiz.BusinessEntities[0].BusinessKey);
}
catch (Microsoft.Uddi.UddiException e)
{ Console.WriteLine("UDDI error: " + e.Message); }
catch (Exception gen)
{ Console.WriteLine("General exception: {0}", gen.Message); }
```

Listing 2: C# Code Example for Finding a Business in UDDI

```
using Microsoft.Uddi;

// Add the following lines to the Main method:
try
{
 // Create a connection to the UDDI node that is to be
accessed.
 UddiConnection myConn = new
UddiConnection("http://test.uddi.microsoft.com/inquire");

 // Create an object to find a business.
 FindBusiness fb = new FindBusiness("Fabrikam");

 // Send the prepared FindBusiness request over the connection.
 BusinessList bizList = fb.Send(myConn);

 // Report the summary result.
 Console.WriteLine("Found " + bizList.BusinessInfos.Count + "
businesses");
}
catch (Microsoft.Uddi.UddiException e)
{ Console.WriteLine("UDDI error: " + e.Message); }
catch (Exception gen)
{ Console.WriteLine("General exception: {0}", gen.Message); }
```

Written by Sanjay Patil & Nick Simha

Integration Approaches:

Web Services vs Distributed Component Models

PART II

Don't reinvent the wheel

Last month we described the enterprise integration environment, as well as the integration problem domain and entailing architectural requirements. This month, we'll look at how Web services address these architectural requirements, and provide a sidebar that examines the key differences between Web services and various component technologies.

Solution: Web Services

Web services are based on several (emerging and de facto) standard technologies – primarily SOAP, WSDL, and UDDI. SOAP is an XML-based packaging protocol and is neutral with respect to the network access protocols or component models of:

- The interaction patterns the endpoint supports
- The abstract models for the data structure supported by the endpoints
- The concrete bindings of the abstract data models (which are needed to exchange data formats on the wire)
- The concrete bindings of endpoint services to transport-specific network addresses

Technically, WSDL (Web Services Definition Language) isn't essential, and a SOAP-based infrastructure alone may be sufficient for accomplishing an integration solution. WSDL provides a formal notation for defining the endpoint capabilities in an abstract manner, and a format for encoding the concrete bindings for actual interactions – without which the clients of a service would have to fully understand and hard code the service invocation logic. WSDL captures the endpoint capabilities in an abstract manner so the service can be implemented independently and the necessary client-side interfaces can be generated using tools. The abstract nature of WSDL also makes it possible to develop a reusable client program that can bind dynamically to any service provider who offers the same WSDL interface. In addition, other business scenarios, such as discovery or orchestration of services, depend heavily upon the WSDL abstraction.

Similarly, the Universal Description, Discovery, and Integration (UDDI) interface is not essential to Web services integration, but provides additional capabilities that promote the use of Web services. UDDI provides a standard, interoperable means by which applications and organizations make themselves known, and find each other dynamically over the Internet.

Do They Meet the Requirements?

Let's analyze whether these Web services technologies in fact meet the requirements outlined in the previous section. As we noted before, the primary requirement for the integration solution is to support neutrality along various dimensions by establishing an abstraction layer that hides the specifics of the endpoint implementations. WSDL, along with its use of W3C XML Schema for defining data types, is best suited to define, in an abstract manner, the data types and service interfaces of the disparate endpoints. These data types and service interfaces can be mapped to different languages, middleware interfaces, and so on, thereby providing language/platform neutrality.

WSDL also fully supports transport neutrality as it allows separate specifications of the abstract service interfaces and their bindings for each specific transport protocol. SOAP is the key to supporting a transport-neutral infrastructure for the actual production and consumption of messages, as it supports binding to different transports.

The self-describing XML documents and self-describing SOAP messages make it possible to build a loosely coupled, document-style integration environment. The extensibility of SOAP is critical to bridging the disparate security and transaction models of the applications behind the endpoints. SOAP extensibility facil-

AUTHOR BIOS:

Sanjay Patil is a distinguished engineer within the Web Services Integration Platform group at IONA Technologies. He is actively involved with many e-business standards; his areas of interest are distributed infrastructure for B2B and enterprise integration, Web services, and Internet technologies.
SANJAY.PATIL@IONA.COM

Nick Simha is a sales engineer working with IONA's customers to help architect e-business solutions. His areas of interest are application integration, Web services, supply chain management, and security.
NSIMHA@IONA.COM

itates the standardizing of constructs that carry the minimal necessary context for bridging these different security and transaction models.

Web Services

Web services go a long way toward fulfilling the architectural requirements for enterprise integration. They are neutral with respect to object models and types, and provide asynchronous communication of high-level, semantically rich, XML business documents. Web services are based on emerging standards, and have achieved broad acceptance in the industry in a very short time. We note that Web services standards accomplished this much more quickly than other standards efforts (such as CORBA and J2EE), and we believe this trend is likely to continue.

In addition to addressing most of the existing enterprise integration requirements, Web services also provide an opportunity for standardizing certain capabilities that are either supported in a proprietary manner by integration products or completely missing from the other distributed computing technologies. For example, it is commonly accepted that custom integration solutions that bridge endpoints to meet specific integration needs do not offer much flexibility in adapting the solutions to changing needs.

The changing needs may involve integrating with new endpoints or changes in the business logic that controls the integration application flow. A business process management environment – one that supports complex business rules as well as integrated endpoints in a standard manner – is the real solution in this regard. Although traditional EAI solutions support most of the capabilities needed for business process management, there has been little standardization in this area.

Web services offer a great opportunity in precisely this area. While distributed technologies such as CORBA provide excellent support in integrating disparate endpoints, they do not inherently (or at least not easily) support building business process management solutions. Because Web services provide a semantically rich integration environment, they make it much easier to build business process management solutions that:

- "Orchestrate" multiple business functions from disparate applications
- Allow the system to apply "business rules" dynamically, e.g., for content-based routing of messages
- Provide semantic mappings between multiple XML business documents in a declarative fashion

Web services expose endpoint capabilities in a self-describing and semantically rich manner. The two basic kinds of capabilities are the document types the endpoint supports and the services the endpoint offers. In addition to making it possible to solve many existing integration problems in a standard manner, these semantically rich descriptions open up new avenues for achieving integration efficiencies.

For example, the use of XML for data format description makes it possible to apply XML-based technologies such as XPATH and XSLT to manipulate the data. Powerful tools can be built to enable business-level features such as content-based routing and data aggregation. Similarly, because the service interfaces are defined in WSDL (which is itself an XML application), XML technologies can be applied to interrogate the service capabilities and integrate discovered functions as Web services orchestrations.

Obviously, new uses of Web services pose new challenges, such as a need for developing standards for Web services orchestration and for encoding the routing logic. Substantial effort is currently being applied by standards bodies, consortia, and private companies to investigate these new avenues and standardize the technologies.

Conclusion

Web services and other distributed technologies, such as J2EE and CORBA, are complementary (see sidebar), and the choice of one over another is a matter of choosing the right tool for the right job. While this is a cliché, it is an illuminating one. After all, is a hammer a "better" tool than a screwdriver? Clichés typically contain a nugget of wisdom, and an even more important one to keep in mind is, "If the only tool you have is a hammer, everything looks like a nail."

Although, theoretically, one technology can be contorted to solve problems that are better solved by another technology, such an effort introduces certain complexities. It is critical to understand that no technology can be a panacea, and that a successful technology always has a particular problem domain that guides its design center.

For example, CORBA is better suited for building distributed applications in controlled environments, where it is possible to share a common object model among all distributed entities, where there are no restrictions on the granularity or volume of the communications between the distributed entities, and where the deployment is more or less permanent such that the system may find a benefit in mapping the network addresses directly to object references. However, the integration scenarios described above require a loose coupling, where CORBA may not be the best fit.

The problems that are paramount in one domain may simply not exist in the other domain. For example, Web services deal mostly with the coarser-grained integration interfaces; therefore, Web services technology, as such, does not have to address object modeling. Object architecture and modeling are hidden from the Web services, as these aspects are relevant only for the design and implementation of the applications, of which only some coarse-grained functions are exposed as Web services.

Extending this notion further, it can be argued that Web services technologies are relieved of the complexities involved with handling object life cycle, object persistence, associating properties with objects, and so on. The need to deal with these complexities in the building of an individual service justifies the rich set of services CORBA and J2EE offer to the application developer, so that complex applications can be developed in a portable manner.

In essence, CORBA and other such technologies are best suited for building complex applications requiring detailed object models and their fine-grained interfaces. Web services technologies are best suited for integrating such applications using high-level, coarse-grained interfaces in a neutral manner.

Given that the problem domain of Web services is quite different from that of other distributed technologies, a feature-by-feature comparison won't be helpful. Using such a comparison to establish the overall "superiority" of one technology over another is counterproductive. The sidebar discusses some of the commonly applicable issues and their relevance to each of the two technologies.

It is possible that the problem domain of Web services will eventually grow to require the definition of a new set of services. Although it is too early to make predictions, the nature of new abstractions and services in the Web services problem domain will most likely involve exploitation of the exposed semantics of data and services by Web services applica-

Web Services and Component Technologies

Distributed component technologies are similar to Web services in many respects, but there are subtle, and not so subtle, differences. In the list below, we use CORBA as a representative distributed component technology, but these points apply to J2EE and DCOM as well.

1. ***Standardized API vs standardized message:*** Both CORBA and Web services provide interoperability across programming languages, operating systems, and hardware platforms. However, CORBA assumes that all interacting entities conform to a standardized object model. Web services standardize the messages exchanged by the interacting entities, which can then be mapped to an arbitrary object model.
2. ***Programming vs declarative paradigms:*** CORBA follows the familiar programming paradigm while Web services are more declarative in nature. For example, IDL does not explicitly support building applications by combining different IDL definitions, but Web services standards (such as BPEL4WS) support the creation of process flows without any programming.
3. ***ACID transactions vs business transactions:*** CORBA's Object Transaction Service (OTS) offers the combined benefits of the transaction and object paradigms, which implicitly require that transactions that access resources like databases are short lived, under the control of the same organization, and connected via a fast, reliable network. In contrast, the "business transactions" implicit in integration scenarios may last days or weeks and may span autonomous business systems, connected over any type of network.
4. ***Object vs message paradigm:*** Most CORBA systems use a Factory pattern for object creation, in which a client interacts with an object instance returned by the factory. This pattern is not relevant for the Web services' message-driven model, in which the type and content of the message determine how it is processed.
5. ***IOR vs URL:*** CORBA Interoperable Object References (IORs) provide location transparency for objects and are strongly typed. Web services are addressed via URLs, which are not strongly typed. A client sends a message to a URL, which may be a proxy that routes the message to the appropriate application.
6. ***IDL vs WSDL:*** Some of the differences are that WSDL combines both an interface definition and an (optional) endpoint definition (i.e., the location of the service on the network) whereas IDL only defines the interface; WSDL uses the W3C schema for a type system, which is richer than IDL's type system (for example, W3C schema lets you specify a valid range of values for a data member); and WSDL provides the ability to bind to multiple transports (while it is possible to bind IDL to multiple transports, IIOP is the only one that is widely supported).
7. ***Static binding vs dynamic binding:*** In CORBA, an object reference is bound to a particular instance of the service provider object, making it hard to support proxy patterns that dynamically select and change client binding to a new service. Web services support development of reusable clients based on the abstract service interfaces in the WSDL. At runtime, the client can bind to specific instances of any service that complies with the same WSDL. For developing loosely coupled systems, dynamic binding provides more flexibility, and Web services are therefore a good fit.
8. ***Naming/Trading service vs UDDI:*** The Naming service maps a logical name to an object reference. The Trading service lets you find a service based on specified service properties. UDDI is more like a Trading service in that a client can query a UDDI registry to find the technical details of a Web service. UDDI supports categorization of services using multiple criteria, and is a general-purpose registry that can be used to query any kind of service, not just Web services. Finally, UDDI provides access to far more information, such as bindings to the services.
9. ***Application platform vs integration platform:*** CORBA and J2EE are best suited for developing distributed object-oriented applications. Web services are best suited as the underlying technology for an integration platform for developing message-oriented integration solutions.
10. ***Intra-domain vs. extra-domain Integration:*** CORBA is best for developing complex applications within a domain, with centrally designed roles and responsibilities of all participants in the context of an overall application model, where deployments are less subject to change and static bindings provide higher efficiencies. But it is not always feasible to get an agreement for a shared application model, nor to live with dependency across the domains. In these scenarios, a loose coupling is preferable, whereby domains can independently design and deploy their applications, and the agreement for interoperability can be reduced to message and coarse-grained service definitions.

tions such as XML search engines and databases, and will be quite different than those existing today in distributed technologies such as CORBA and J2EE.

For example, there are ongoing efforts to standardize the publishing of additional, feature-specific requirements a service may have for security context or transactional context. These standards are primarily based on WSDL extensibility for defining the feature-specific constructs. Once WSDL begins to carry more information in this way, it is natural that applications will start to dynamically query different parts of (one or more) WSDL definitions to make decisions about which service to invoke, how to invoke it, and so on.

Some queries may take the form of applying well-defined business rules, which themselves are candidates for query by design-time tools. Overall, more semantics will be standardized in the form of message definitions and service interfaces, promoting a wider variety of new applications. Fundamental to supporting such a semantically rich environment will be XML and XML technologies. XML is important for representing the various types of information in a self-describing manner, and XML technologies will be important for processing the XML document instances in various ways, such as queries, aggregation, decomposition, and content extraction.

We hope that we provided some illumination on the use of Web services to solve integration problems, and that we have helped you see the overall context and relationship among various distributed computing technologies. The bottom line is that when it comes to integration technology, one size does not fit all. Web services are a welcome addition to the toolbox, and are in no way an attempt to "reinvent the wheel" for distributed computing.

•••

Once again, note that these differences are useful for better understanding the two technologies and the problem domains in which each one is the best fit.

Acknowledgement

We wish to thank John Parodi, principal writer at IONA, for writing support, technical editing, and asking the right questions to keep the article focused.

JavaOne

www.java.sun.com/javaone/sf

Written by Jim MacKay

Business Processes: Turning Integration Upside Down

With maturity comes acceptance

In the IT world, integration became an issue as soon as the second computer with the second application came online. Many different approaches to solving the complex problems associated with integration have been tried since that time, some of them more successful than others. At this point it's safe to say that integration is still an expensive, usually difficult, aspect of every major IT infrastructure. The need to collaborate across multiple businesses as well as large geographical and cultural divides has only added to the list of issues.

AUTHOR BIO:

Jim Mackay serves as senior vice president of webMethods' Business Technology Group, a group within webMethods Global Alliances that serves the technical requirements of the ever-growing number of strategic partnerships signed by the company. Previously, Jim served as CTO of i2 Technologies, where he was responsible for guiding their assessment and creation of technologies and standards to continually improve their market-leading value chain management solutions.
JIM.MACKAY@WEBMETHODS.COM

Ten years ago, integration within an enterprise was typically built in one of two ways: by synchronizing data elements around a database or data warehouse, or by installing a complex, expensive ERP system and modifying every application to conform to that system's data model and interfaces. These types of integrations are commonly called "intelligent bridges." Communicating outside the enterprise with business partners was accomplished via EDI, again requiring expensive infrastructure, as well as a difficult mapping and synchronization exercise. The majority of companies still use one or both of these methods today.

Over time, middleware became available, usually based around a messaging backbone that used its own proprietary interface and protocols for transporting data. The data itself was typically in a format that was defined by the company or the departments that needed to exchange it, and again the data went through complex mappings and transformations to and from the databases and applications that shared it.

Standards

Eventually, standards initiatives came along. Groups such as OAG and OMG worked on standards for representing and transporting data. Huge steps were made in simplifying information access, mapping, and transformation. However, these early standards were rigid and tightly coupled, and didn't allow for the constant changes that are always present in business environments. This made implementations difficult and maintenance costly.

We are now in an era where more interoperable, flexible standards have started to emerge and mature: XML has become the prevalent way of representing shared data, XSLT and XPATH for translation and business logic, WSDL and SOAP for application definition and access to services, BPEL and BPML for business process management. Of course, some of these standards have a long way to go, and there are numerous competing standards in many areas.

Despite their presence, currently no suite of standards addresses the complete set of integration and interoperability problems that exist. However, the great news about standards is that they allow these problems to be broken down and addressed piece by piece. It is now possible to separate application logic from a canonical data model, and both of these from the business process workflow. Applications can have a consistent interface, allowing them to be versioned, or swapped out wholesale for other applications. Data models can be enhanced or modified without changes to the applications that interact with them. Business workflows can be enhanced or configured without changes to the applications that they depend on for their extensive, complex business logic.

A few commercial integration solu-

tions have become available during the last few years that leverage these standards. Through these solutions, an important evolution has started to emerge. The economic disadvantage of building and maintaining an ever increasing number of point-to-point integration solutions within an enterprise became obvious, and the move to building an integration backbone became a siren call for forward-thinking CIOs. Having an integration backbone in turn allowed corporations to achieve many of the benefits that have been promised by believers in packaged integration solutions and the standards that they leverage.

But there is an even more significant outcome from this maturity of integration standards and solutions. The evolution described here allowed the industry to shift its focus from gluing a few applications together and syncing them with a database to focusing instead on the business processes that drive financial operations, supply chains, health care practices, retail transactions, and so on.

The first software vendors to grasp this tectonic shift in focus were the infrastructure players. New companies emerged that focused on the business process management layer. Integration vendors either enhanced their products to include business process and workflow support, or they acquired business process–focused solutions.

Then a consolidation took place among these groups. When the dust settled, all serious integration vendors had solutions that empowered business users to design business processes and their logical flows, independent of the application components that supported these details. Further, they were able to tie these processes into the user interaction, deploy them in a distributed fashion, and manage them in a consistent manner.

Meanwhile, standards have continued to evolve and mature to support this new way of thinking. BPEL4WS has been touted with great fanfare by its founders – IBM, Microsoft, and BEA – as the language that should be used to describe the execution of business processes. Another serious contender, BPML, is a less commercial effort that has been well thought out and endorsed by many. In fact, BPEL is currently mired in intellectual property rights issues that, if not resolved soon, will virtually ensure BPML's growth and further acceptance.

> "What's next for business processes? In a word, maturity."

Integration

All of this focus on the business process has led to a change in how commercial vendors approach the problem of integration. Instead of touting the ability of customers to join applications together for data sharing and synchro-

nization, the new mantra is the ability to provide business processes on top of a service-oriented architecture, usually broken down by industry verticals. Vendors provide a huge head start in deployment because the large, common piece of the business process has been predeveloped. All that remains to be done is to extend or customize the process for the business practices specific to the deployment.

> "We are now in an era where more interoperable, flexible standards have started to emerge and mature."

Systems integrators were the first to jump on this idea. The expertise they gained over the years from implementations of business processes using their own methodologies, combined with their expertise in bringing together best-of-breed composite applications, led to a compelling value proposition. In fact, some of the major firms now have architectural frameworks to leverage business processes that harness heterogeneous components. Examples are the CSC e3 framework, and BearingPoint's Solution Integration Architecture.

Of course, integration software vendors have also seized this opportunity. Early attempts (such as CrossWorld's work in business process modeling) failed because they were too far ahead of the maturity curve of the supporting standards, and lacked the credibility of a larger player or a more mature market. In the last year webMethods, Tibco, and IBM have all gained footholds in the business process–centric approach.

Perhaps the largest group to latch onto business process integration is enterprise system vendors. The first was J.D. Edwards, with its XBP/XPI approach. SAP developed its version, known as xApps, on top of its NetWeaver platform.. PeopleSoft has AppConnect, and the most recent, and probably most publicized, is Siebel's UAN initiative.

Each vendor approaches the solution in a slightly different way. SAP and PeopleSoft, for the most part, have built all of the underlying technology themselves as an extension of the latest generation of their core infrastructure and middleware. At this point, they both still outsource adapter support. JD Edwards, on the other hand, embeds webMethods technology as their XPI infrastructure, then builds their XBPs, or Cross Business Processes, using webMethods middleware. Siebel has decided that, to achieve true openness and interoperability they need to define platform-neutral business processes using standards, and then work with multiple vendors, including, webMethods and Tibco, to create packaged solutions to deploy these processes on their respective platforms.

Whether employing business processes through the offerings of integration vendors, system integrators, or enterprise software vendors, a key set of requirements must exist in order to be successful:

- First, a compelling business problem that will return the investment in a short time. Generally, these are business problems endemic to specific industries, so implemented solutions are as close to "packaged" as possible.
- Next, the composite applications that make up the detailed functionality supporting these business processes have to be available, and exposed as services through standards.
- Finally, a service-oriented integration infrastructure for the design, development, extension, deployment, maintenance and management of the processes must be in place.

This integration infrastructure needs to leverage the decoupling that was discussed earlier. It needs to encourage and leverage the separation of the process flow from the human interaction from the data documents and their associated representation and translation. It also must be able to instantiate business processes that were defined without the end implementation and deployment in mind. And last, it must facilitate the management of multiple business processes executing simultaneously across a distributed environment.

> "What's next for business processes? In a word, maturity."

Summary

Management of the business process has become the next frontier. Vendors are rushing to make the real-time enterprise a reality. The end goal is to be able to identify a set of critical metrics associated with a business process, and to track and measure performance against these metrics as part of the day-to-day execution of the process. This should happen as near to real time as possible, allowing an end user to react to any anomalies as soon as they occur. It requires the ability to capture information as it flows through business processes, analyze it against historical information, and propagate any red-flag issues through the proper channels.

What's next for business processes? In a word, maturity. As is typical in our industry, the hype runs well ahead of the reality. Implementation of business processes at this point is still a very professional services-oriented effort, and very few canned solutions (beyond simple data synchronization) from the enterprise vendors have been more than 70% repeatable. Infrastructure and middleware will continue to mature and embrace open standards as the standards themselves mature. More and more applications will expose a service layer. And the marriage of components required for the real-time enterprise will continue.

The notion of the business process leveraging a set of underlying heterogeneous application services is here to stay. The focus has moved to this top-down model, rather than the bottom-up approach of wiring applications and databases together point to point. Integration initiatives as we knew them have, in fact, been turned upside down.

introductory subscription offer!

A TRULY INDEPENDENT VOICE IN THE WORLD OF .NET

.NET Developer's Journal is the leading independent monthly publication targeted at .NET developers, particularly advanced developers. It brings .NET developers everything they need to know in order to create great software.

Published monthly, ***.NET Developer's Journal*** covers everything of interest to developers working with Microsoft .NET technologies – all from a completely independent and nonbiased perspective. Articles are carefully selected for their prime technical content – technical details aren't watered down with lots of needless opinion and commentary. Apart from the technical content, expert analysts and software industry commentators keep developers and their managers abreast of the business forces influencing .NET's rapid development.

Wholly independent of both Microsoft Corporation and the other main players now shaping the course of .NET and Web services, ***.NET Developer's Journal*** represents a **constant, neutral, expert voice** on the state of .NET today – the good, the bad, and the ugly…no exceptions.

SYS-CON MEDIA

SUBSCRIBE ONLINE!

www.sys-con.com/dotnet/

or Call

1 888 303-5282

Here's what you'll find in every issue of .netdj:

- Security Watch
- Mobile .NET
- .NET Trends
- Tech Tips
- Standards Watch
- Business Alerts
- .NET News
- Book and Software Announcements

.NET Developer's Journal is for .NET developers of all levels, especially those "in the trenches" creating .NET code on a daily basis:

- For beginners:
  Each issue contains step-by-step tutorials.
- For intermediate developers:
  There are more advanced articles.
- For advanced .NET developers:
  In-depth technical articles and columns written by acknowledged .NET experts.

Regardless of their experience level, *.NET Developer's Journal* assumes that everyone reading it shares a common desire to understand as much about .NET – and the business forces shaping it – as possible. Our aim is to help bring our reader-developers closer and closer to that goal with each and every new issue!

SAVE 16% OFF

THE ANNUAL COVER PRICE

Get 12 issues of .NETDJ for only $69.99!

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

ANNUAL COVER PRICE:

~~$83.88~~

YOU PAY

$69.99

YOU SAVE

$13.89

OFF THE ANNUAL COVER PRICE

Putting Data and Business Process Integration in Context

Requirements that can go from user interface to database

Concerns about economic efficiency and risk reduction always weigh heavily on IT organizations as they embark on the implementation of any new technology. This is especially true when integrating enterprise applications that must operate over intranets and the Internet. While seamless integration and across-the-board automation may be highly visible IT goals, the business process needs of employees, customers, business partners, and suppliers are equally important. All enterprise constituents must be able to rapidly access knowledge and resources based on their role, function and need – and they must be able to get this information in an intuitive and contextual manner. This rigorous set of demands dictates that a new breed of technology solutions be used to simplify the integration of multivendor business applications and vast amounts of decentralized data.

AUTHOR BIO:

Paola Lubet is the vice president of Marketing for PeopleSoft AppConnect and oversees its global marketing strategy, including the Enterprise Portal, Enterprise Warehouse, and Integration Broker. She has more than 15 years of experience in the high-tech industry. Before joining PeopleSoft, Paola served as senior vice president of Marketing and Products at SoftwareMarkets, where she was responsible for marketing, business development and product strategy.

She holds a master's in computer science from Italy's Universita' degli Studi di Torino.

DAVID_SAYED@PEOPLESOFT.COM

Historically, CIOs wanting to integrate their data and business processes have had to select individual user portal, enterprise application integration (EAI), and data warehousing technologies. Once all the products were purchased, substantial amounts of custom programming had to be performed to fit the technologies together, let alone scale them and enable operation in real time. This explains why more than one third of the total IT budget can be exhausted on application integration and customization. A new generation of pre-integrated platform has emerged to address this portal, data, and business process dilemma.

Achieving equilibrium between technologies and business processes – while realizing greater economic efficiency at reduced risk – has prompted IT organizations to seek out new and better solutions that provide application integration, data warehousing, and user portal frameworks. The latest generation of standards-based suites pre-integrate these functions through Web services, and support the applications and middleware of third-party vendors, thereby offering the mechanisms needed to balance the enterprise's business process and technology requirements.

Bringing Together People, Processes, and Data

Pre-integrated suites built on open-standards frameworks are designed to bring together a multitude of applications, databases, and information types. These offerings, able to run over the Internet with no specific client requirements, employ a consistent data model and user interface that allow business analysts to enhance applications that support the enterprise's business processes without performing detailed coding. Key areas where this is especially beneficial include security, roles, look-and-feel, languages, documentation, maintenance, and installation.

By using off-the-shelf and pre-integrated suites, enterprise data and business processes can be brought into equilibrium quickly. Highly customizable user portals can be created that allow customers, suppliers, partners, and employees to intuitively access and use information in the context of their roles and responsibilities. Enterprise applications, legacy systems, and business processes become functional in ways that help people analyze business scenarios using consistently warehoused data, thereby allowing them to make better-informed decisions faster.

> “...rapid access to reliable information allows employees, customers, partners and suppliers to make confident business decisions...”

WebServices .NET J2EE XML JOURNAL

LEARN WEB SERVICES. GET A NEW JOB !

The Best .NET Coverage Guaranteed!

SUBSCRIBE TODAY TO THE WORLD'S LEADING WEB SERVICES RESOURCE

Get Up to Speed with the Fourth Wave in Software Development

- **Real-World Web Services**: XML's Killer App*!*
- How to Use **SOAP** in the Enterprise
- Demystifying **ebXML** for success
- **Authentication**, **Authorization**, and **Auditing**
- **BPM** - Business Process Management
- Latest Information on **Evolving Standards**
- Vital technology **insights** from the nation's leading Technologists
- Industry **Case Studies** and **Success Stories**
- Making the Most of **.NET**
- **Web Services Security**
- How to Develop and Market Your Web Services
- **EAI** and Application Integration Tips
- **The Marketplace**: Tools, Engines, and Servers
- Integrating **XML** in a Web Services Environment
- **Wireless**: Enable Your **WAP** Projects and Build **Wireless Applications** with Web Services*!*
- Real-World **UDDI**
- **Swing**-Compliant Web Services
- *and much, much more!*

Only $69.99 for 1 year (12 issues)*

* Newsstand price $83.88 for 1 year

Subscribe online at www.wsj2.com or call 888 303-5282

*Offer subject to change without notice

SYS-CON MEDIA

SYS-CON Media, the world's leading *i*-technology publisher of developer magazines and journals, brings you the most comprehensive coverage of Web services.

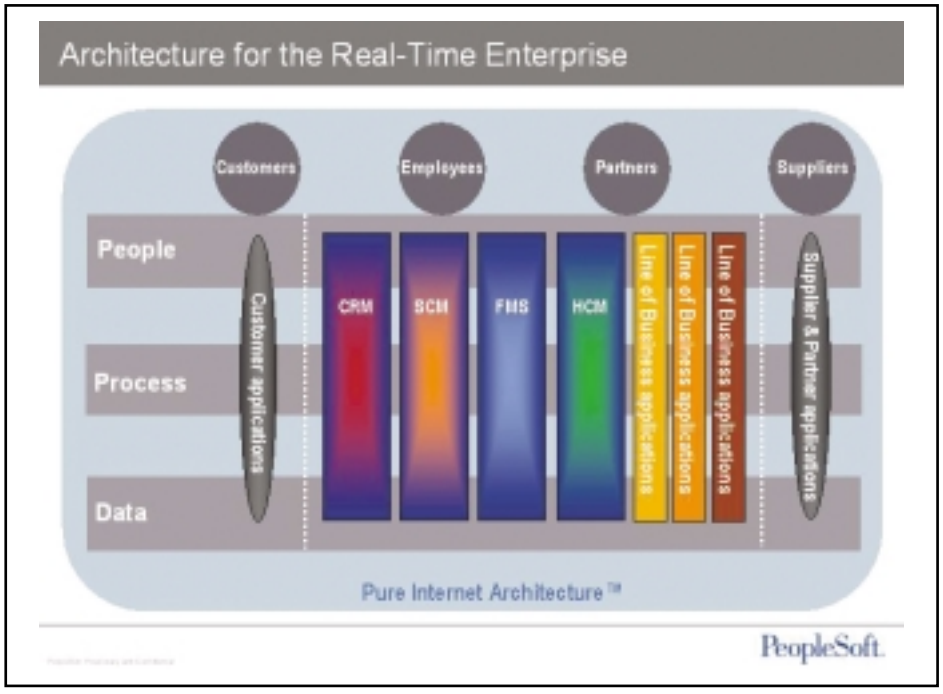


FIGURE 1 **Off-the-shelf pre-integrated solutions are built on open standards frameworks. They span applications and bring together business processes and data through use of portals, integration technology, and advanced data warehousing techniques. These solutions offer full compatibility with Web services, J2EE, and .NET technologies; all prevailing databases; and legacy applications.**

Because enterprise knowledge and information is digitized and readily accessible, reliance on traditional information transfer approaches such as writing notes, making multiple data entries, or working through intermediaries can be nearly eliminated. Information can flow seamlessly across business applications, delivering speed and improved accuracy. This provides economic efficiency and gives users greater confidence in the integrity of their information (see Figure 1).

Open Infrastructure Frameworks

An open framework approach to application integration is essential if the enterprise is to future-proof itself. The demands of business – some of which must occur in real time – require connecting people, enabling business processes, and unifying data at a global level. Adopting an information technology strategy based on an Internet framework with preconfigured portal, integration, and warehouse solutions can help create these stable environments. This allows IT organizations to sidestep potential technology traps while saving time and money. It also allows business processes to be extended directly to customers, partners, suppliers, and employees. The enterprise's applications can be run, data can be aggregated and consistently warehoused, and analytic processing can be performed and delivered so people can interpret results dynamically. Best of all, each user can access knowledge using an array of integrated tools that are made available through their individual portal and a standard Web browser.

> "...from a financial perspective the management and maintenance of multiple information repositories can be reduced through consolidation of the infrastructure investment."

Portals and Processes

Portals, as the name implies, are the means by which timely and relevant information is conveyed to the user, who may be internal or external to the enterprise. They serve as a personalized, role-based hub to access business information and applications, and to support transactions and services from internal and external sources. When portals and the applications accessed through them are intuitive and easy to navigate, users embrace them. When the opposite occurs, users will continue working but tend to retain data in side files and revert to traditional work habits. This causes the application's status as the system of record to be diminished. Conversely, by employing a Web browser as the single point of entry, the applications accessed through each user portal become the system of record by default.

Implementing advanced portal features can further simplify and streamline the interaction between users and processes. Using intelligent context management, additional valuable information can be identified and presented to a user performing a specific task. For example, if an accounts receivable employee were to query and pull up customer accounts containing outstanding and overdue bills, they could view key performance indicators (KPIs) about those customers to determine if sending a collections notice would impact pending sales opportunities. All pertinent and consolidated intelligence would be presented via the portal and a better-informed credit decision could be made.

Integration Engines Power Data and Business Processes

Business processes have always spanned individual organizations to varying degrees. Yet the internal enterprise systems that support them have barely communicated and shared data in a consistent format. Achieving optimal enterprise efficiency requires the breaking down of these communications and interpretational barriers. It requires an understanding that real informational value comes from seamlessly connecting business processes in ways that are transparent to users. Integration engines that employ Web services–based technologies play an important role in satisfying this requirement. These engines are designed to manage interoperability between core, third-party, and legacy systems; and provide the automated brokering needed to handle, route, and transform data be-

tween applications. Operating in a hub-and-spoke manner, integration engines eliminate the programming and time required to create point-to-point integrations between individual applications, resulting in reduced ongoing maintenance costs. Enterprise integration points and process packs define and structure the application integration paths based on knowledge of the underlying application. By taking advantage of these technologies, the time and cost associated with manual programming can be overcome.

Integration engines are designed to understand the semantics and manage the use of business objects in order to interpret and process essential information. Categories include, but are not limited to, roles, security, and languages that drive how the enterprise's systems infrastructure works with portals, applications, processes, and data. From employee hiring to manufacturing, shipping, billing, and servicing customers, integration at the business process level is essential and it must be cohesive. For example, integrated business processes with embedded business objects would know how to interpret and handle multiple currency fields and language-sensitive content. This process-with-objects approach obviates the need for expensive low-level programming while providing faster system response and greater informational continuity.

Coupling Data Warehousing with Analytics

Individual silos of information have historically shown a propensity for growth in environments where applications are not integrated. Extracting this data for use across the enterprise requires the use of business analytics. Powered by the integration engine, information from all types of systems, applications, and data repositories can be acquired and assimilated even across the Internet. By embedding analytics directly into business processes, information can be aggregated, analyzed, and reported using Web services. This approach allows users to connect with any data warehouse, application, or portal that employs this technology.

This enterprise-level warehousing approach consolidates global data to create a single information source that feeds current performance information to the users' desktop dashboards. This way, the central data warehouse can become the enterprise's core business intelligence platform. From it, multidimensional analyses can be performed on data regardless of its point of origin. Applying business analytics, dynamic and forward-looking business processes can be adopted, allowing the enterprise to overcome its traditional reliance on historical data as the primary decision support mechanism.

By coupling the data warehouse with business analytics, enterprise constituents can be given a global view of quantitative data based on their job function and be provided with the qualitative means to efficiently interpret it, analyze it, and put it to work. As a result, business functions can be brought into better alignment through improved visibility, future events can be better anticipated, and courses of action can be altered quickly to meet opportunities as they arise. This rapid access to reliable information allows employees, customers, partners, and suppliers to make confident business decisions in ways that cross software, systems, and enterprise boundaries. Further, from a financial perspective the management and maintenance of multiple information repositories can be reduced through consolidation of the infrastructure investment.

Summary

The business process requirements of the global enterprise span the user interface to the database. By leveraging open frameworks that support advanced, pre-integrated solutions, software applications – regardless of vendor or lineage – and data can be brought together into a cohesive, scalable, and dynamic environment. This strategic model accommodates business needs today and adapts to technology changes tomorrow, which future-proofs the investment in business processes. By adopting an open framework approach to enterprise application integration, new functionality can be rolled out incrementally and transparently using Web services. It can be done according to schedule and budget, and at much lower risk. Internet technologies created a worldwide environment that supports an open flow of information between systems. Now it comes to the enterprise.

WebServices JOURNAL
.NET J2EE XML

PRESIDENT AND CEO
Fuat A. Kircaali fuat@sys-con.com

VP, BUSINESS DEVELOPMENT
Grisha Davida grisha@sys-con.com

ADVERTISING
SENIOR VP, SALES & MARKETING
Carmen Gonzalez carmen@sys-con.com
VP, SALES & MARKETING
Miles Silverman miles@sys-con.com
ADVERTISING DIRECTOR
Robyn Forma robyn@sys-con.com
DIRECTOR, SALES & MARKETING
Megan Ring-Mussa megan@sys-con.com
ASSOCIATE SALES MANAGERS
Carrie Gebert carrie@sys-con.com
Alisa Catalano alisa@sys-con.com
Kristin Kuhnle kristin@sys-con.com

SYS-CON EVENTS
PRESIDENT, SYS-CON EVENTS
Grisha Davida grisha@sys-con.com
CONFERENCE MANAGER
Michael Lynch mike@sys-con.com

CUSTOMER RELATIONS / JDJ STORE
SENIOR CUSTOMER CARE / CIRCULATION SPECIALIST
Niki Panagopoulos niki@sys-con.com
CUSTOMER SERVICE REPRESENTATIVE
Shelia Dickerson shelia@sys-con.com
JDJ STORE MANAGER
Rachel McGouran rachel@sys-con.com

SYS-CON.COM
VP, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com
ONLINE EDITOR
Lin Goetz lin@sys-con.com

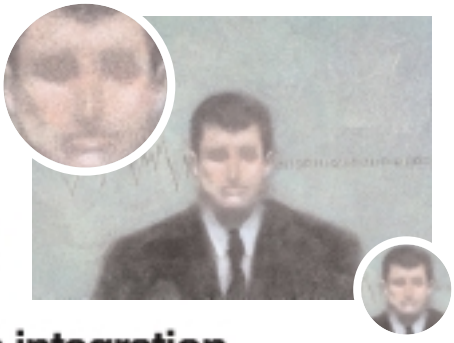
ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS RECEIVABLE
Kerri Von Achen kerri@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1-888-303-5282
FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,
PLEASE SEND YOUR LETTERS TO SUBSCRIPTION DEPARTMENT
COVER PRICE: $6.99/ISSUE
DOMESTIC: $69.99/YR (12 ISSUES)
CANADA/MEXICO: $89.99/YR
ALL OTHER COUNTRIES: $99.99/YR
(U.S. BANKS OR MONEY ORDERS)

SYS-CON MEDIA

Written by Andrew Astor

Patterns in Web Services Projects



The future of enterprise integration

Having been endorsed by virtually every technology vendor on the planet, Web services are now evolving from "feature" to "fabric." They are moving from the latest buzzword (hot new feature) to a mature and accepted technology (fabric of the technology landscape). The hype is fading; it's no longer interesting to develop Web services simply as a proof of technology, or as an end in themselves.

This article is the first in a series that explores the use of Web services in real-world situations, with the purpose of identifying usage patterns. The idea behind the series is to help answer questions like: Where and how do Web services deliver value? Where might they be counter-indicated? What works? What doesn't?

For the first time, there is enough data from real-world, business-driven projects to allow us to begin to recognize patterns. The data analyzed for this series is drawn from production Web services case studies culled from two sources:

- Sand Hill Group's study, "The Web Services Derby" (see www.sandhill.com)
- webMethods customers employing Web service–based integration (WSBI)

Where possible, case studies are attributed to specific companies. However, in many cases, organizations requested anonymity, generally because their projects are not public and are considered competitive differentiators.

Project Objectives

In this article, I explore common objectives of Web services projects. In this context, "objectives" are project executives' stated reasons as to the main purpose of the effort. Clearly, most projects are (or should be) undertaken to increase revenues, decrease costs, or improve customer satisfaction. Such high-level goals, though, are not what we're investigating here. We're attempting to categorize project success criteria into a more meaningful set of types or styles. We are looking for patterns of project objectives.

Four major project types emerged from the analysis. A few initiatives were too broad or too unique to categorize, but the vast majority – more than 60 of the 70 case studies reviewed – coalesced around the following four major themes:

- Real-time application interoperability
- Common business services
- Architectural agility
- Customer or partner self-service

Objective 1: Real-Time Application Integration

By far the most common project objective was real-time application integration of heterogeneous systems. This is the most fundamental reason for Web services projects. After all, the most basic premise behind Web services is to get applications to provide services to each other, typically in real time. And since it is a fact of life that every IT organization today is multiplatform, multivendor, and multilanguage, it isn't surprising to find an increasing demand for real-time integration of heterogeneous business systems. (*Note*: The idea that integration is at the center of Web services should come as no surprise. Regular readers will have seen this author's view expressed previously that Web services are the name given to a set of standards associated with integration. For more on this topic, see "From Subroutines to Web Services: An Evolutionary View – Beyond Client/Server Computing," ***WSJ***, Vol. 2, issue 2.)

Business drivers behind these projects were quite varied, but fell into a few key themes.

Single View of Data

A key motivator of application integration projects is the consolidation of data from multiple systems into a single view. For example, a major insurance carrier used Web services to unify its customer data from multiple systems to a single view for its sales and service staff. Similarly, a biotechnology manufacturer created a single global view of its inventory for procurement decision-making.

Author Bio

Andy Astor joined webMethods in 2002 as vice president for Enterprise Web Services and is responsible for driving the company's Web services strategy and execution. Prior to joining webMethods, Andy was a vice president at D&B, where he led worldwide customer-facing products, including all Web- and Internet-based applications. His work at D&B also included the development and launch of one of the earliest commercial Web services. He is on the Editorial Advisory Board for *Web Services Journal*, and was IT Strategy track chair for the 2002 Web Services Edge conference in New York City. Andy holds a B.A. in mathematics from Clark University and an MBA from the Wharton School.

ANDY.ASTOR@WEBMETHODS.COM

Business Process Automation and Acceleration

Web service-based integration (WSBI) projects are often undertaken to improve manual or batch-oriented business processes, or to tie together individual business processes that can be streamlined.

For example, the sales force of an athletic footwear manufacturer spent nearly 80% of their time with their smallest accounts, since these accounts did not have EDI capability and needed to place their orders manually. Using Web services, this company was able to expose order-processing capability to these smaller customers, freeing up the sales force for more profitable account management activities

In another example, an outsourced electronics manufacturer differentiated itself from its competition through process orchestration and differentiated services. They replaced their EDI systems with event-driven solutions such as real-time partner invoicing and delivery date forecasting.

Information Sharing and Collaboration

The need to share information instantly with branch offices, trading partners, and customers is another common theme within WSBI projects. At Infinity Pharmaceuticals, for example, hundreds of thousands of chemical compounds are evaluated regularly for their drug potential. As soon as one of these compounds is recognized as unpromising and no longer worth pursuing, other company systems are instantly informed, and the compound is dropped from further analysis everywhere.

In another example, a high-tech integrated communications manufacturer used a WSBI approach to collaborate with its partners to predict demand, and fed that information to collaborative inventory and ordering systems that enabled them to have the correct stock at the correct time.

Objective 2: Common Business Services

Another very common objective for Web services projects is to aggregate individual application capabilities into coarsely grained business services that are exposed for use by multiple systems. Businesses find this approach compelling for at least three reasons:

- ***It shields applications from each other's complexities:*** A well-established interface is developed for each business service, and applications that use the service need not understand how it accomplishes its task.
- ***It promotes reuse:*** Since the services are defined in coarse, business terms, developers have an easier time locating them and understanding their purpose.
- ***It simplifies the technical environment:*** The number of connections between applications can be dramatically reduced, since each application communicates only with the common business service. If "n" is the number of applications, then the number of connections can be theoretically reduced from n-squared to n.

Common business service projects where multiple front-end systems need to access multiple back-end systems are very common. For example, at Future Electronics, a $2 billion distributor of electronic components, orders can arrive through several channels (e.g., e-mail, phone, Web), and these orders may need processing by several back-end systems as they arrive. It is both complex and expensive to develop these systems, and even more difficult to maintain them. To alleviate these difficulties, Future recently built a set of common Web services for the creation of all customer orders. Regardless of the channel through which the order is entered, the same set of services is used to create the order, and these services are responsible for navigating the required processing in back-end systems. In addition to the benefits listed above, this solution delivered the ability to know at any instant, in real time, the current order backlog in the system. The CIO's comment: "How do you put a value on that?"

Objective 3: Architectural Agility

Another common purpose for many Web services projects is to develop an agile and responsive application infrastructure. Over time, most large organizations have built up a complex and inflexible systems infrastructure, made up of applications that were either built without regard for intersystem communications, or which were hard-wired to each other. This condition, coupled with the unprecedented pace of change in both business and technology today has led to a situation where required changes simply cannot be implemented quickly enough. Increasingly, many forward-thinking organizations are responding to this challenge by increasing their systems agility via Web service–enabled solutions.

Generally, the goal of these projects is to replace legacy infrastructures with a service-oriented framework. But such an effort cannot be done all at once, especially in large enterprises; there are simply too many systems that need to be changed. The most common approach is implemented in two phases. In the first, legacy applications are wrapped into a Web services framework and exposed in a single, unified manner, via clearly defined interfaces. Once this step is complete, the individual legacy systems are then replaced over time. Since their interfaces don't change, the end user experience can be left unaffected.

In one example of an architectural agility project, a major retail banking institution sought to present a single face to its customers after acquiring another bank. The architectural team recognized that a complete merge of the two banks' systems would take too long, and that the project risk of a "big bang" implementation approach was simply too high. Instead, they placed a "veneer" on their back-end systems, wrapping them in Web services and exposing them to a common front-end application. The Web service wrapper reconciled the interfaces of the two systems, exposing them to the front end in a common manner. To the customers, the bank appeared to have completely unified its systems. The bank is now working to merge its back-end systems, and to eliminate redundancy through application sunsetting.

In another financial services example, a large brokerage re-architected its infrastructure to be more responsive to rapidly changing market conditions and the frequent need to develop new products. Its margins were shrinking due to increased competition from discount brokerage houses, and this company decided to develop "integration agility" – the ability to quickly develop functionality that could not be offered with its old infrastructure. To achieve this, they organized their applications into elemental business services (using Web service technologies) that could be rapidly snapped together to provide flexible services to the business.

Objective 4: Self Service

Self service – for employees, partners, and customers – is the last pattern we will discuss. In these scenarios, companies improve their service levels, reduce costs, and/or increase revenue opportunities by exposing businesses beyond the normal application borders.

At NEC Electronics, a new system provides access to real-time inventory and production information collected from different plants. It consists of a Web portal and an integration framework that gives customers, partners, and even branch offices unprecedented, secure access to their data. Prior to the delivery of this system, wait times of several days were not uncommon.

Dun & Bradstreet uses a Web service–based application to give its customers access to its vast credit database. By embedding this access directly into their applications, customers automate their credit analysis process, dramatically reducing decision times. In addition, since access to D&B is embedded within the customer business process, alternative credit sources are more difficult to exploit, making D&B's services the default approach. This can clearly have a positive impact on D&B's revenues.

In a similar approach, a large logistics and transportation company has exposed its shipping and tracking capability to its customers via Web services, allowing customers to embed this access directly into their own applications. By becoming part of the customer's system, this company's ability to hold onto its customers may be enhanced.

Conclusion

As we review the patterns of project objectives identified in this article, it's useful to note that none of them specifically required Web service technologies to implement. In almost every case, project executives were able to name at least one other technology that could have been used as an alternative. This raises the question then, "Why Web services?"

The answer lies in a point made earlier in the article: "Web services" is the name that has been given to the emerging standards associated with integration. Mature integration technologies are available for everything Web services deliver and more. Project objectives, then, are integration-oriented objectives that are not specific to Web service standards. Nevertheless, standards always win out in the end. They lower project costs, risk, and time to completion. It should therefore come as no surprise that Web services–based solutions are the only possible future for enterprise integration.

***

In my next "Web Services in the Real World," I'll explore the technical drivers behind the selection of Web service technologies for integration projects.

SaturdaySessions

We understand the pressures of work and how difficult it can be to get time off. That is why we have designed this workshop to be held in one day and, as a special bonus, on the weekend, so no days off from work. **Your boss will be happy!**

Alan Williamson
JDJ Editor-in-Chief

JDJ Workshop with Alan Williamson

M T W T F **S** S

Coming to you...

May:	**June:**
NEW YORK	ATLANTA
DALLAS	BOSTON

Performance > Efficiency > Reliability

This one-day intensive workshop is designed for developers who wish to increase the efficiency and reliability of their code development.

1) The day will begin by looking at the various hints and tips you can utilize at the code level to improve the quality and reduce the number of bugs you have to contend with.

2) The next part will look at Apache's Ant and how you can use this freely available tool for your own development, irrespective of your IDE.

3) Last, and most important, as the old saying goes: "You can never do enough testing." This session will look at JUnit and show you how to start building test harnesses for your code so you can begin your testing strategy.

>Performance

Java is a powerful language. While it offers a rich array of tools, the fundamentals mustn't be overlooked. Improving your code at the core layer will result in great improvements in efficiency and produce (hopefully) less bugs. We'll look at the do's and don'ts of programming and learn many hints and tips that will accelerate your Java coding.

>Efficiency with Ant

Apache's Ant is a powerful scripting tool that enables developers to define and execute routine software development tasks using the simplicity and extensibility of XML. Ant provides a comprehensive mechanism for managing software development projects, including compilation, deployment, testing, and execution. In addition, it is compatible with any IDE or operating system.

> Reliability with JUnit

A critical measure of the success of software is whether or not it executes properly. Equally important, however, is whether that software does what it was intended to do. JUnit is an open-source testing framework that provides a simple way for developers to define how their software should work. JUnit then provides test runners that process your intentions and verify that your code performs as intended. The result is software that not only works, but works in the correct way.

What you will receive...

- ✓ *INTENSIVE ONE-DAY SESSION*
- ✓ *DETAILED COURSE NOTES AND EXCLUSIVE ONLINE RESOURCES*
- ✓ *JDJ CD ARCHIVE*

ONLY $295*
*JDJ subscribers ($395 non-subscribers)
*GROUP DISCOUNTS AVAILABLE

To Register

www.sys-con.com/education

Call 201 802-3058

SPONSORED BY
JAVA DEVELOPER'S JOURNAL

PRODUCED BY SYS-CON EVENTS

PRESENTED BY SYS-CON i-TECHNOLOGY EDUCATION

PLE MAGAZINES ONLINE

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

Pick a 3-Pack, a 6-Pack or a 9-Pack					
☐ 3-Pack	☐ 1YR	☐ 2YR	☐ U.S.	☐ Can/Mex	☐ Intl.
☐ 6-Pack	☐ 1YR	☐ 2YR	☐ U.S.	☐ Can/Mex	☐ Intl.
☐ 9-Pack	☐ 1YR	☐ 2YR	☐ U.S.	☐ Can/Mex	☐ Intl.

☐ Linux Business & Technology

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ Java Developer's Journal

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ Web Services Journal

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ .NET Developer's Journal

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ XML-Journal

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ WebLogic Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ ColdFusion Developer's Journal

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

☐ Wireless Business & Technology

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

☐ WebSphere Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

☐ PowerBuilder Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

3-Pack

Pick any 3 of our magazines and save up to $275.00

Pay only $175 for a 1 year subscription plus a FREE CD

- 2 Year – $299.00
- Canada/Mexico – $245.00
- International – $315.00

6-Pack

Pick any 6 of our magazines and save up to $350.00

Pay only $395 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $669.00
- Canada/Mexico – $555.00
- International – $710.00

9-Pack

Pick 9 of our magazines and save up to $400.00

Pay only $495 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $839.00
- Canada/Mexico – $695.00
- International – $890.00

OFFER SUBJECT TO CHANGE WITHOUT NOTICE

The Reality, Challenges, and Enormous Potential of Web Services

Changing the way we think

Web services certainly have the potential to improve and simplify the process of enterprise application integration (EAI). By establishing a nonproprietary, universally accepted standard of communication between applications, Web services can succeed where other approaches have struggled. With Web services, organizations can integrate key applications without relying on costly, time-consuming, proprietary, and maintenance-intensive solutions. That said, Web services alone are not in and of themselves a complete integration platform, but rather merely the enabling standards. As a result, Web services cannot serve as a complete substitute for an EAI platform in many cases. With robust management solutions, however, Web services are evolving to complement and strengthen traditional EAI methodologies, making integration projects less time consuming, inflexible, and costly.

AUTHOR BIO:

Eric Pulier has been a pioneer in the software and digital interactive industries for over 15 years. As CEO of Digital Evolution, he devotes himself to a singular focus: the standards-based real-time enterprise. Toward this end, Eric is currently driving Digital Evolution's development and joint venture relationships with some of the largest companies in the world.

ERIC@DIGEV.COM

Problems with Integration Projects Persist

Most integration projects fail. A recent Forrester Research report stated that almost 65% of EAI projects run behind schedule and are significantly over budget. On average, the cost is approximately $6.4 million per project by the time the smoke clears. In general, application integration is a far longer and more expensive proposition than the outcome seems to justify.

Change management is perhaps the greatest source of pain with EAI initiatives. With conflicts in data message definitions or mismatches in proprietary components, a change to either side of a set of integrated systems necessitates a cumbersome and often costly change implementation process. The unintended result of this situation is the creation of a "job for life" for a developer with specialized EAI package skills. When combined with the obligation to pay recurring maintenance fees on proprietary packages and buy add-on EAI components as systems grow and change, an EAI project can create a long-term maintenance overhead that is untenable.

Long development and implementation timeframes are other factors that mitigate the effectiveness of EAI. The average project life cycle is 20 months. The time required to sort out political issues in the organization (a factor that can kill the integration project before it even begins), select the vendors, gather requirements, and then implement the solution, may take so long that the early goals of the program are obsolete by the time it is completed.

Whether the integration is done by a proprietary EAI package or through a custom development program, the result is usually an "island" of integration between two applications that will then be difficult or expensive to connect with other applications. When integration is then extended beyond the firewall, the issues become even more complex.

Figure 1 shows an example of poor EAI performance and high cost. With three "islands of integration," integrating between unconnected systems would be prohibitively expensive – and the cost of maintaining the islands is very high. Change management is complex, time-consuming, and costly. *Bottom line:* this EAI initiative is a costly trap with limited extensibility.

Figure 2 illustrates the missed opportunities for integration. Including systems out-

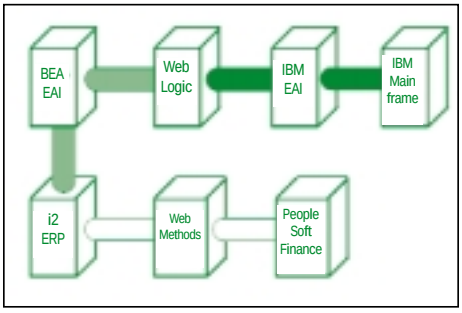


FIGURE 1 **Islands of integration: i2 integrated with BEA WebLogic using a BEA EAI solution; BEA WebLogic integrated with IBM Mainframe using IBM EAI solution; i2 integrated with PeopleSoft Finance using webMethods**

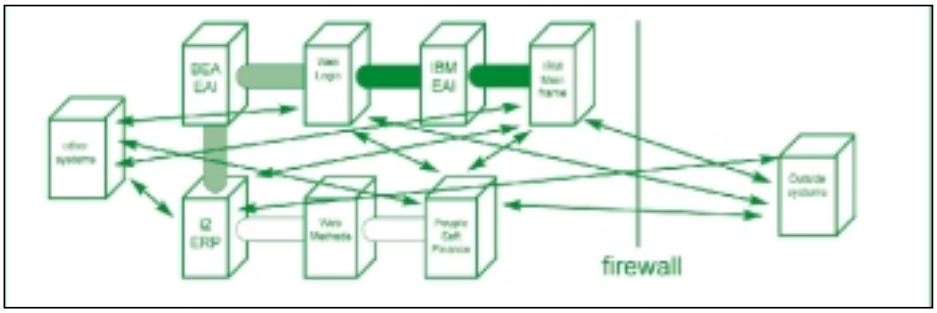


FIGURE 2 | **Missed opportunities for integration in the enterprise**

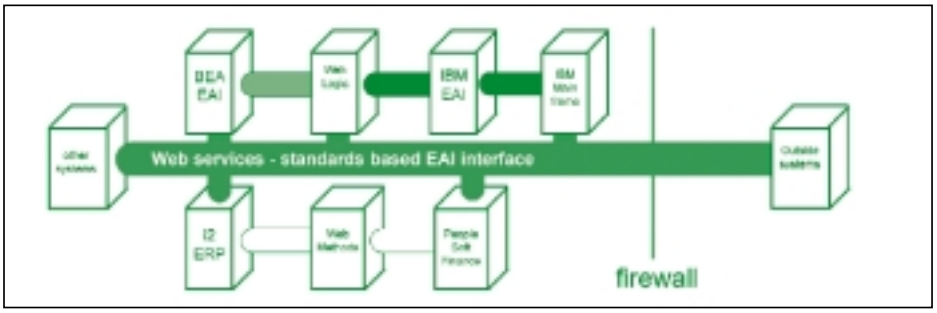


FIGURE 3 | **How Web services could facilitate EAI**

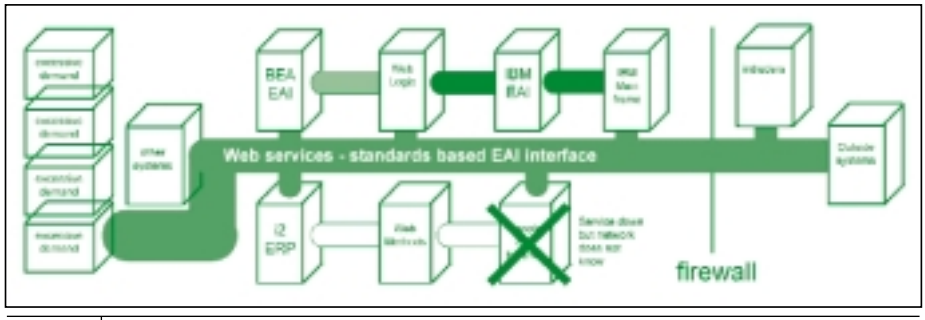


FIGURE 4 | **Potential problems in an open, unmanaged, Web services-based EAI initiative**

side the firewall and other internal systems, there are 12 potential connections that cannot be realized in this configuration.

The Immediate Impact of Web Services on EAI

Using the example in Figures 1 and 2, Web services can enable integration of the "islands" without any further proprietary or custom development. As shown in Figure 3, by exposing the key features of each application as a Web service, it becomes possible for the applications to exchange data and functions freely without the need for a proprietary or custom interface. The applications exchange messages in SOAP using Internet protocols that already exist in the enterprise infrastructure. Developers responsible for the integration can locate the required functionality using WSDL found in the UDDI. The Web services act as EDI-like components to extend EAI solutions outside the enterprise and into collaborative customer domains.

The Web services–based EAI system is extensible without requiring additional software licenses from a proprietary system. Change management is greatly simplified. Data integrity issues become easier to solve through the universality of the XML data exchange. And of course, it becomes easier to extend beyond the firewall because there is no longer an issue of communication or procedure call standards.

Ultimately, Web services act as building blocks for a "virtual application" built using orchestrated Web service components. This is a departure from the old way of building distributed or monolithic applications in that it explicitly allows for reuse while maintaining a distributed, scalable architecture and standardized components.

Solving Some Problems, but Causing Others

Web services are easier to design, implement, and deploy than any other traditional distributed technology. Web services can span their influence by easily exposing and integrating the services on the Web. The rationale behind Web services is to divide and conquer large business problems into small services. Each service addresses a specific business need; therefore, the integration of these small services solves the larger business problems at hand.

There is no need to own the implementation of a service purchased through only one solution provider, as you may discover services and orchestrate them into agile business processes regardless of the underlying platform.

Promising as they are, however, Web services are no panacea for integration challenges. Making Web services serve the complex needs of EAI in a real-world enterprise requires solving a number of challenges (see Figure 4):

1. Security is a significant problem in Web services if not managed correctly.
2. Without management and monitoring, it is impossible to know how the Web services EAI infrastructure is operating, or in fact if it is operating at all.
3. Web services must be able to travel across multiple messaging protocols such as HTTP and JMS.
4. Enterprises that scale EAI projects using Web services will result in large networks of Web services. Those networks will have to be managed in terms of routing, fail-over, and load balancing. Furthermore, a deep UDDI will be required to keep track of all the critical metadata attached to each service in the network.
5. EAI platforms remain important for transactionality and reliable transport.

Enterprises that intend to rely on Web services as a method for implementing critical business processes must prepare to manage those Web services by building, buying, or subscribing to a suitable Web Services Management (WSM) solu-

tion. Over time, WSM will become a critical part of any Web services strategy. Organizations will incorporate WSM into their enterprise architecture to take advantage of service orientation without losing reliability, scalability, and measurability. Although Web services technology provides the promise of seamless interoperability of disparate systems, just exposing business functionality as services is not only insufficient, but also dangerous. Only with comprehensive management can the full benefits of Web services in application integration be realized, and the risks mitigated.

> "Web services alone are not in and of themselves a complete integration platform, but rather merely the enabling standards"

The Future Impact

As standards become more stable and robust, and WSM solutions more common, Web services should move up the integration "value chain" to encompass more than just message transport and interconnect. In short order we will see the influence of Web services technology on higher-value business process management and business analytics applications (e.g., making it easier to share and incorporate business processes from outside trading partners and bringing the concept of the real-time enterprise closer to reality). Eventually, Web services also have the potential to mature and grow into a ubiquitous platform for core application development itself. We could see it breaking out of the integration box entirely, becoming as useful for constructing new applications as for integrating existing ones. Consider the following three types of Web services usage:

Type 1: Interoperability

Early use of deployed Web services was seen primarily as a form of middleware to integrate various applications across different platforms and development languages. During this phase of Web services experimentation, developers are "service-orienting" legacy business processes or transactions into components. These types of legacy processes, such as order processing, shipping status, or inventory queries, are natural targets for initial Web services deployments. The early focus on creating Web services from legacy systems shouldn't be surprising as legacy systems support most fundamental business processes in the world's largest organizations. Across the Fortune 500, somewhere around 70% of strategic applications still reside on legacy applications.

Type 2: Moving Up and Out: B2B Integration

A major driver behind the Web services trend is that it facilitates integration beyond enterprise boundaries in situations where you lack control of the development platforms, languages, and middleware. This will highlight the expense and constraints of today's EDI, and usher in a new era of standards-based B2B transaction exchange.

Type 3: Enterprise Agility and Awareness: Application Orchestration and Cross-Process Analytics

Web services evolved to enable a hyper-efficient platform for application development via the orchestration of modular, loosely coupled pieces. Web services component assembly processes will begin to transform the way we design and build software, affecting programmatic and user interfaces. By using Web services–enabling tools, developers can increase their productivity by gaining integration-free, component-based access to distributed business functions. The clean separation of business logic from the user interface results in explosive value, freeing the business from the constraints of entrenched IT infrastructure while leveraging legacy investment. In this model, older applications are exposed via standards as "black boxes," the underlying platform is rendered irrelevant, and new applications roll out in a manner suited for seamless interoperability. These disparate services are orchestrated into virtual applications, each piece used and reused by different processes depending on the context.

> "Web services are easier to design, implement, and deploy than any other traditional distributed technology"

Most important, the shift to a service-oriented architecture will enable true visibility and agility in the enterprise on a level that is unimagined today. The prohibitive cost of attaining true enterprise awareness has left most organizations running blind, making decisions based on incomplete or inaccurate views of their environment. Web services will soon become a competitive differentiator, allowing agile organizations to make better decisions faster, and change their processes in real time in response to an ever-shifting market opportunity.

Summary

In the future, a consultant helping a company implement a CRM system will be able to develop application user interfaces by applying workflows to a rich collection of preexisting Web services. This could be done at the speed of prototyping. In their purest form, Web services may not come to full fruition until late 2003 or 2004, but their influence is already being felt in IT shops across the globe. Today virtually 100% of organizations with $1 billion dollars in revenue or more have declared a loosely coupled, standards-based architecture their goal. Of that group, more than 70% have started to prepare for the transition to a distributed architectural framework and roll out initial Web services pilots. Within a few years, Web services will radically change the way we think about, design, build, deploy, and integrate business applications.

Premiering
June 2003
at
JavaOne

www.sys-con.com

Millions of Linux Users One Magazine

Linux Business and Technology

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *Linux Business & Technology* is aimed squarely at providing this group with the knowledge and background that will allow them to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *Linux Business & Technology* will not feature low-level code snippets but will focus instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month will see a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features will include:

- *Advice on Linux Infrastructure*
- *Detailed Software Reviews*
- *Migration Advice*
- *Hardware Advice*
- *CEO Guest Editorials*
- *Recruiting/Certification Advice*
- *Latest News That Matters*
- *Case Studies*

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY

$49.99

12 ISSUES/YR

*OFFER SUBJECT TO CHANGE WITHOUT NOTICE

SUBSCRIBE TODAY!

WWW.SYS-CON.COM

OR CALL

1-888-303-5282

LINUX EDGE conference&expo

June 3–5.....LONDON
June 24–26.....BERLIN
September..HONG KONG
October.....CALIFORNIA

FOR ADVERTISING INFORMATION:

CALL 201 802.3020 OR VISIT WWW.SYS-CON.COM

SYS-CON MEDIA

The World's Leading i-Technology Publisher

Hynes Convention Center: Boston, MA

Web Services Edge 2003 East

INTERNATIONAL WEB SERVICES CONFERENCE & EXPO

Russ' Tool Shed: Russ Fustino shows how to use Visual Studio .NET

When SYS-CON Media's sister company, SYS-CON Events, began preparing last year for this spring's "Web Services Edge" Conference & Expo, one consideration was paramount: every effort in the nine-month preparation cycle should be geared toward making it indisputably the world's largest independent Java, .NET, XML, and Web services event.

When SYS-CON Media's sister company, SYS-CON Events, began preparing last year for this spring's "Web Services Edge" Conference and Expo, one consideration was paramount: every effort of the nine-month preparation cycle should be geared towards making it indisputably the world's largest independent Java, .NET, XML, and Web services event. Also, the show would mirror, if you will, the Web services paradigm itself – undergirded by XML, and with .NET and J2EE sitting on top of a middleware layer ensuring application integration and platform interoperability. Accordingly there were four separate tracks: Java, .NET, XML, and Web services

Held on March 18–20, 2003, at the centrally located Hynes Convention Center in Boston, Massachusetts, Web Services Edge 2003 East made its mark right from the get-go, with delegates registering from the widest possible variety of companies, both in terms of technology and geography. Not only were they attracted by the four specific session tracks; they had also come to take advantage of the all-day *i*-technology tutorials, whether it was the Sun Microsystems Java University, the IBM XML Certified Developer Fast Path, Russ Fustino's .NET workshop (Russ' Tool Shed) or Derek Ferguson's Mobile .NET tutorial.

The show opened with a very well-attended keynote from Oracle's John Magee, VP of Oracle 9*i* Application Server. Magee stressed that the key to understanding why Web services, unlike its distributed-computing forerunners like COM and CORBA, is prevailing in the enterprise space is that Web services do more than merely enable interoperability between platforms and integration between applications… they also do so simply.

What drives their simplicity, Magee explained to the large audience that had gathered to hear him, is standards.

The afternoon keynote offerings on Day One of the conference were equally well received. First came a panel coordinated by the Web Services Interoperability Organization (WS-I). The WS-I is an open industry organization chartered to promote Web services interoperability across platforms, operating systems, and programming languages, and the panel discussion took place against the backdrop of the WS-I Basic Profile 1.0, consisting of a set of nonproprietary Web services specifications. The working draft for this, the audience learned, was approved just 4 weeks before the Web Services Edge 2003 East Conference & Expo opened.

But security, the panel agreed, was the primary priority. Now that corporations like Merrill Lynch and DaimlerChrysler have joined the organization, the importance of ensuring that everyone adheres to the same specification is more important than ever before. Web services is moving beyond mere SOAP, WSDL, and UDDI toward addressing security, messaging, reliability, and transactions. Eric Newcomer, chief technology officer of IONA Technologies, emphasized the importance of the World Wide Web Consortium (W3C) approach to these challenges, an effort that centers on the W3C's Web Services Specification Effort.

The Web services keynote panel was quickly followed by the highlight of Day One for many of the delegates gathered in the keynote hall: an address by Miguel de Icaza, the impossibly young and extremely gifted founder and leader of the GNOME Foundation, cofounder of Ximian, Inc., and .NET expert extraordinary – as anyone needs to be who leads a project designed to port .NET to the Linux operating system.

The Mono Project, as de Icaza's project is called, clearly fascinated the broad mix of developers attending the Conference.

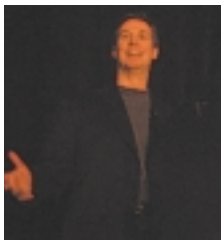
"I Don't 'Get' Web Services" Admits de Icaza

Admitting that he didn't (personally) "get it" as far as Web services was concerned, de Icaza noted that Gtk# is Ximian's main interest for GNOME development, because it is cross platform (looks like Windows under Windows, Mac OS X under Mac OS X/X11, and Gtk+ under X11).

At the end of his keynote address, scores of developers of every stripe surrounded de Icaza for further questions. The response to his good humor, rapid delivery, technical savvy, and sheer charm had been overwhelming and with his keynote Web Services Edge 2003 (East) passed a significant milestone: no previous conference in the SYS-CON conference series had previously included so wide a range of technical content. Like Web services itself, our aim was to be cross-platform, and it was definitely a case of mission accomplished.

Day Two saw Sun's Mark Herring take the stage, and his mastery of the whole Web services paradigm was clearly in evidence. Extended coverage of both his Java keynote and of the subsequent .NET keynote address by Jesse Liberty, are available on SYS-CON's main conference Web site, www.sys-con.com/WebServices Edge2003East.

The closing keynote discussion panel, which for many turned out to be the high point



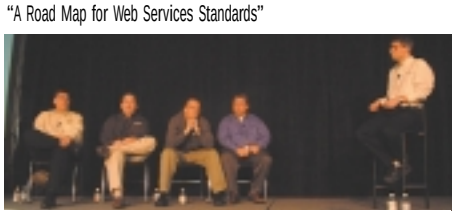
John Magee, Oracle: "Developing in a Services-Based World"

Mark Herring, Sun Microsystems: "Bridging the Gap Between WS-Myth and WS-Reality"

Miguel de Icaza, Ximian: "The Mono Project"

Jesse Liberty, Liberty Associates: ".NET Web Services"

WS-I Panel Discussion: "A Road Map for Web Services Standards"



.NET Panel Discussion: "Real-World .NET"

Java Panel Discussion: "The Future of Java"

of the entire keynote program, was a wide-ranging and sometimes heated debate about "The Future of Java." The panel's moderator was *Java Developer's Journal* editor-in-chief Alan Williamson and the entire intense and highly interactive hour exemplified very well how a SYS-CON i-technology conference program differs from that offered by any other conference organizer. This was panel discussion at its best.

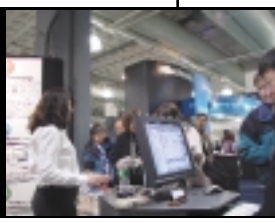
True to the very close links that SYS-CON Media and SYS-CON Events enjoy with the entire software development industry, the participants in this final panel at Web Services Edge 2003 (East) had come to Boston from far and wide. Sun's chief technology evangelist Simon Phipps had flown over from the UK and ***WSJ*** Editorial Advisory Board member Tyler Jewell (BEA's director of technology evangelism) had traveled from Los Angeles. *WSJ* Technical Editor, Sonic Software's vice president and chief technology evangelist Dave Chappell nipped over from Bedford, MA; Aligo's CTO Jeff Capone flew in from San Francisco; and JBoss founder Marc Fleury came up from the JBoss Group's company headquarters in Atlanta, Georgia.

We fully expect our next Conference & Expo, Web Services Edge 2003 (West) in October, to be equally chock-full of the movers and shakers who are helping to shape the software development industry as it continues its headlong progress towards distributed computing with full application integration and interoperability.

All in all it was a marvelous conference, and the Expo hall too was intensely busy from the moment it opened to the moment it closed two days later. .

This is not the end of the Web services "story," nor is it even the beginning of the end; but March 18–20 in Boston's Hynes Convention Center may well have marked the end of the beginning.

Come join WSJ for Phase Two in Santa Clara in October.



SYS-CON Events Would Like to Thank Our Sponsors and Exhibitors for Making Web Services Edge 2003 East a Success

Actional Corporation
www.actional.com

Actional solutions help organizations avoid the cost and complexity of unmanaged Web services deployments. Actional's Web services management platform provides unmatched visibility, flexibility, and active control across the entire Web service network – ensuring uptime while dramatically reducing the costs of ongoing Web services management.

Altio, Inc.
www.altio.com

Altio offers an XML presentation layer that allows you to bring a live, desktop-style interface to Web applications. Using AltioLive, businesses can integrate their Web applications, legacy applications, and Web services into one unified interface with drag-and-drop capabilities and real-time data feeds.

Altova
www.altova.com

Altova produces and markets XMLSPY 5 Suite, the ultimate Web services development tools suite, featuring a SOAP Debugger and Tester, and many other tools for developing XML Schema, WSDL and UDDI files, and much more.

asp.netPRO
www.aspnetpro.com

This publication targets professional developers who use Microsoft's ASP.NET (Active Server Pages.NET) technology to build Web-enabled applications and business solutions.

ASPstreet.com
www.aspstreet.com

ASPstreet.com is a collaborative portal and marketplace for application service providers (ASPs) and the Web services industry. In the rapidly expanding Web and .NET world, ASPstreet.com is the one-stop hub for all players interested in this emerging marketplace.

Attachmate Corporation
www.attachmate.com

Founded in 1982, Attachmate Corporation, a Microsoft Gold Certified Partner, is a privately held worldwide supplier of mission-critical host access solutions for Fortune 500 and Global 2000 enterprises. Attachmate provides direct, real-time access to back-office systems, Web- and desktop-to-host solutions, plus flexible enterprise application integration and Web services offerings.

Computer Associates
www.ca.com

Computer Associates International, Inc. (CA), delivers software that manages a company's infrastructure by addressing today's most critical business processes, information and technology management challenges. More than two decades of innovation, commitment and quality make CA the technology partner of choice.

Compuware Corporation
www.compuware.com

Compuware Corporation provides business value through software and professional services that optimize productivity and reduce cost across the application life cycle. Pattern-based, model-driven development with Compuware's OptimalJ provides one of the best means to tackle the challenges development organizations face today.

Conquer-IT! (Trans-World Resources, LLC)
www.twresources.com

Think all J2EE training is similar? Trans-World Resources has provided exceptional J2EE training on behalf of BEA, according to Bill Lawson, senior manager, BEA.

DataPower
www.datapower.com

DataPower provides enterprises with an intelligent, XML-aware network infrastructure to ensure unparalleled performance and security of next-generation XML Web services applications. DataPower's patent-pending XG3 technology powers the industry's first XML-aware networking devices to provide immediate return on technology investments while streamlining application deployments.

Digital Evolution
www.digev.com

Tested in the Fortune 500, the DE Management Server 2.0 is the only Web services management application that delivers enterprise-level security, UDDI, multitransport protocols, routing, and performance monitoring for a complex enterprise. Digital Evolution's products and solutions give organizations the power to manage their business processes within a standards-based, vendor-neutral framework.

EAI Industry Consortium
www.eaiindustry.org

The EAI Industry Consortium is a nonprofit global advocacy group developed to promote enterprise application integration through sponsored research, the establishment of standards and guidelines, best practices, and articulation of strategic and measurable benefits. The member-driven consortium, designed as an EAI information hub, encompasses marketplace education, resource tools, and EAI trends, providing members with a venue to develop, create, and debate.

Ektron Inc.
www.ektron.com

Ektron is the vendor of choice for flexible, scalable, and affordable solutions designed to achieve Web content success – today, tomorrow, and beyond. Worldwide, more than 7,000 organizations trust Ektron to help solve their real-world Web content management problems. What's your Web problem?

Forum Systems
www.forumsys.com

To maintain trust with business partners, companies cannot afford to leave entrusted information susceptible to tampering or theft. While most companies focus on XML security in transit, the network is most vulnerable. Forum Systems offers a complete answer, enabling companies to extend their Web services without extending risk. The FS Sentry 2500 ensures integrity through digital signatures, document structure validation, and document translation. It also provides security, rendering encrypted information useless to violators on the network or in transit.

Hewlett-Packard
www.hp.com

HP showcased Web application and services management tools and solutions that enable software developers to easily design and develop for manageability. HP OpenView and Eclipse development tools and new technologies for voice interactive services streamline the development process.

HSPstreet.com
www.hspstreet.com

HSPstreet.com is a collaborative portal enabling buyers to find the appropriate Web hosting plan and partners. In the rapidly expanding Web hosting world, HSPstreet.com is a one-stop hub for all players interested in the HSP marketplace.

IONA
www.iona.com

IONA is the leading provider of Rapid Integration software, with more than 4,500 customers worldwide. IONA's rapid integration software products are built on service-oriented architectures that increase reuse of software assets to deliver lasting results, standards-based software that enables vendor independence, and incremental deployment capabilities that lower the customer's risk.

Itellix Software Solutions
www.itellix.com

Itellix, a software products and services company, focuses on the use of contemporary technologies to realize technology-driven business initiatives. Its flagship product, Wisiba, is a platform-independent, standards-compliant Web services management suite that facilitates organizations to derive commercial value from their Web services initiatives. The Wisiba product suite is composed of Wisiba – Nucleus, Commerce, Optimizer, Intelligence, and Orchestra.

iTKO, Inc.
www.itko.com

iTKO develops some of today's most complex CRM, Web, and back-office applications. iTKO's latest innovation, LISA, made its debut at the Web Services Edge East Conference & Expo and has everyone talking. LISA is a no-code unit, functional, regression, and load-testing product that will change the way you feel about automated testing forever.

Java Developer's Journal
www.sys-con.com/java

Java Developer's Journal is the premier independent, vendor-neutral magazine serving the information needs of the entire community of developers in the Java programming language and Java platform.

JavaWorld
www.javaworld.com

For an audience that demands comprehensive, hands-on information about the news and trends in Java technology, no other information source can match *JavaWorld*'s content. *JavaWorld* is assembled by an award-winning editorial team and authored by seasoned Java developers and industry experts.

Jinfonet Software
www.jinfonet.com

Jinfonet Software is the developer of JReport, a 100% Java reporting tool, written to run on any platform, access any data source, and create any report. JReport Designer is a visual report design interface; JReport Enterprise Server is a high-performance, J2EE-compliant server for deploying reports over the Web.

McCabe & Associates
www.mccabe.com

McCabe & Associates enables IT to deliver better applications by providing products and process that implement a relevant, repeatable, and measurable approach to managing software changes and their effects on the testing and quality of applications. McCabe products include McCabe QA, McCabe Test, and McCabe TRUEchange.

Melissa Data
www.melissadata.com

Melissa Data, founded in 1985, is the leading provider of data-quality solutions to help you achieve the highest level of quality contact information. You'll find our versatile line of software, components, database, and services are easy to use and cost-effective. You'll save money, boost response rates, and increase your bottom line.

Merant
www.merant.com

Merant's PVCS products help you organize, manage, and protect software development assets and improve development efficiency. Leverage the common-use interface of Visual Studio .NET with PVCS to gain greater productivity and control with version/build management, issue and change management, and life-cycle development.

Microsoft Corporation
www.microsoft.com

Introducing Visual Studio .NET – visionary yet practical, the single comprehensive development tool for creat-

ing the next generation of applications has arrived. Developers can use Visual Studio .NET to build the next-generation Internet, create powerful applications fast and effectively, and span any platform or device. Visual Studio .NET is the only development environment built from the ground up for XML Web services. By allowing applications to share data over the Internet, XML Web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

Mindreef, LLC

www.mindreef.com

Mindreef SOAPscope is an easy-to-use, toolkit-independent diagnostic aid for developers, testers, and application support technicians who must isolate Web services problems. SOAPscope has a powerful logger/viewer that shows the SOAP communication flow, making it easy to view, isolate, and debug Web services problems.

My SQL

www.mysql.com

MySQL develops, markets and supports the MySQL database server, the world's most popular open-source database. With an estimated 4 million installations and over 27,000 downloads per day, MySQL is becoming the core of many high-volume, business-critical applications for companies like Yahoo! and Cisco.

.NET Developer's Journal

www.sys-con.com/dotnet

.NET Developer's Journal covers everything of interest to developers working with Microsoft .NET technologies – all from a completely independent and nonbiased perspective.

OASIS

www.oasis-open.org

OASIS is the nonprofit, international consortium that has been providing open solutions for electronic data interchange since 1993. Dedicated from its inception to the technology now known as XML, OASIS is the world's largest independent, vendor-neutral organization for the standardization of XML applications in electronic commerce. The primary mission of OASIS and its members is to identify and resolve interoperability issues that exist between XML applications and technologies.

Oracle Corporation

www.oracle.com

Oracle Corporation is the world's largest enterprise software company, providing enterprise software to the world's largest and most successful businesses. With annual revenues of more than $9.4 billion, the company offers its database, tools, and application products, along with related consulting, education, and support services.

Parasoft Corporation

www.parasoft.com

Parasoft is a leading provider of error-prevention tools that help companies improve their software development processes. These tools assist teams working on C/C++, Java, Web, and enterprise applications to significantly reduce costs by shortening development cycles, improving overall quality, and reducing time to market.

PerfectXML

www.perfectxml.com

The main focus of PerfectXML is XML, Web services, and related technologies for business people and technologists – from both a practitioner and learner perspective. The PerfectXML team works hard to provide best collection of well-organized links, developer-oriented articles and other content, up-to-date news, code samples, and an exhaustive listing of software and tools available.

Rational Software

www.rational.com

Rational Software provides a software development platform that improves the speed, quality, and predictability of software projects. This integrated, full life–cycle solution combines software engineering best practices, market-leading tools, and professional services. Ninety-six of the Fortune 100 rely on Rational tools and services to build better software, faster.

Sams Publishing–Pearson Tech Group

www.samspublishing.com/index.asp

Sams Publishing has more than 500 titles in print, and is one of the most successful computer book publishers in the world. From introductory tutorials to comprehensive reference books, Sams Publishing focuses on teaching tomorrow's programmers, developers, and system administrators the skills they need to build and manage emerging technologies.

SD Times

www.sdtimes.com

SD Times is the newspaper of record for the software development industry. It provides news, news analysis, specialized features, and comprehensive analyses on new products, alliances, and emerging market trends for software and application development managers, IT managers, and ISVs, who manage development projects. Subscriptions are free.

SlickEdit, Inc.

www.slickedit.com

SlickEdit Inc., provides software developers with the most comprehensive and flexible code editor available. Visual SlickEdit, proven across a wide range of programming languages and on Windows, Linux, Unix, and zSeries mainframe platforms, enables even the most accomplished developers to code faster and meet increasingly aggressive deadlines.

Softartisans, Inc.

www.softartisans.com

With enterprise-class products like ExcelWriter and FileUp, SoftArtisans assists clients on any development platform to build robust reporting, file transfer, and Web-based solutions. Over 16,000 customers in more than 70 countries benefit from SoftArtisans products and technical services.

Software AG

www.softwareag.com

Software AG, Inc., is a pioneer in XML solutions and a leading global provider of system software and services enabling enterprise data integration and management. Our products and solutions focus on standards-based XML integration such as Web services and enterprise content management.

SpiritSoft

www.spiritsoft.com

SpiritSoft, the leading provider of integration software using JMS and JCache technologies, enables developers to align IT resources on a unified foundation built on open standards, which lowers the cost of an existing IT infrastructure and boosts performance and return on investment. SpiritWave Message Server, the leading Java Message Service (JMS) implementation, provides reliable, flexible, and secure messaging to enable flexible integration between new and existing enterprise applications. SpiritWave Open JMS Framework also allows developers to integrate proprietary middleware and offers a range of interface/language bindings for legacy enterprise applications.

Sun Microsystems

www.sun.com

Sun was founded with one driving vision. A vision of computers that talk to each other no matter who built them. A vision in which technology works for you, not the other way around. While others protected proprietary, stand-alone architectures, we focused on taking companies into the network age, providing systems and software with the scalability and reliability needed for the electronic marketplace.

Sybase, Inc.

www.sybase.com

Sybase has always delivered solutions that help customers to share data. Sybase is platform independent and integrates everything: platforms, application servers, components, databases, portals, processes, message brokers, and mobile/wireless. Our technologies promote ease of use, leverage best practices, ensure positive ROI, and help your organization build a successful, pragmatic strategy based on next-generation technologies.

Teamstudio, Inc.

www.teamstudio.com

Founded in 1996, Teamstudio develops and markets award-winning, agile software tools that enhance developer productivity and improve application quality. Product lines include solutions for Lotus Notes, Domino, and Java developers, and Web-to-host integration.

Trilog Group, Inc.

www.triloggroup.com

Trilog Group is the only software company that provides a fully integrated platform for J2EE RAD, BPM, and Web services integration. Fortune 500 companies have used FlowBuilder Visual XSP Studio to drastically reduce development and integration costs by capitalizing on its ultra-rapid, highly visual, XML-centric method to manufacture and assemble enterprise application components and Web services.

Vultus, Inc.

www.vultus.com

Vultus speeds the adoption of best-of-breed Web applications that greatly enhance corporate Internet strategies. Our technology provides a flexible, secure platform to facilitate closer business relationships with new and existing customers. Vultus products are designed to extend long-term IT and business ROI for our customers by adhering to the latest industry protocols.

Web Services Journal

www.sys-con.com/webservices

Web Services Journal is the premier publication addressing the technical and strategic depth of Web services. It is for anyone who wishes to apply the new model for creating and using distributed applications across the Internet, utilizing common interfaces for efficient communication and high-level interoperability.

WowGao.com

www.wowgao.com

WowGao.com is an international leader in Web services deployment, hosting, and portal. GAO Research Inc. (www.GaoResearch.com), offers solutions for modem (ADSL, V.92, V.90,. etc), fax, modem/fax relays, telephony, speech, VoIP, and gateways. GAO Web Services Inc. (www.GaoWebServices.com), specializes in Web services. Its powerful, modular, and scalable J2EE-based UDDI is available for license.

XML-Journal

www.sys-con.com/xml

XML-Journal is the world's leading print and online resource for Internet technology professionals involved with the worldwide development and implementation of XML. Each issue contains the latest news concerning enterprise application integration and Web services, XML standards, new developments in e-commerce, product reviews, tutorials, case studies, and interviews with *i*-technology leaders.

Xtremesoft, Inc.

www.xtremesoft.com

Xtremesoft is the leading provider of software solutions that maximize the availability of applications on the Microsoft platform. These solutions enable businesses to transform and process data derived from their applications into business intelligence upon which decisions can be made.

EAI Industry Consortium Partners With Gartner, Giga, DCI, SYS-CON and BrainStorm

(Calgary, Alberta, Canada) – The EAI Industry Consortium, global voice for the industry leaders in enterprise application integration, announced its alliance with BrainStorm Group Inc., DCI, Gartner Inc., Giga, and SYS-CON media in a drive to showcase its member companies' competitive edge in the industry. The partnership with the leading trade show service companies seeks to introduce business process integration within and between organizations using Internet-standard protocols and formats and to inform the marketplace of emerging integration technologies. www.eaiindustry.com

Sun Microsystems Launches Sun Developer Network

(Boston) – Sun Microsystems, Inc., has announced the launch of the Sun Developer Network, a new program focused on providing software developers with the content, training, support, and technology access they require to innovate and deliver applications and system solutions faster. The expanded program and portal located at java.sun.com integrates community dialog, content, access to technologies, and advanced learning that will enable Sun and individual developers to more quickly and efficiently implement applications that span multiple technologies, standards, and operating environments.
http://sun.com

Sun Announces Fully-Integrated Web Services Platform, Proposes JCP Specification

(Boston) – Sun Microsystems, Inc., has announced the availability of the Sun ONE Web Services Platform Developer Edition, the industry's first complete and fully integrated platform for Java-based Web services and application development. The Sun Open Net Environment (Sun ONE) Web Services Platform delivers a complete platform offered at a single price and contains all the elements necessary to develop network-based enterprise applications in a single install.

Sun also announced that it has submitted a proposal to the Java Community Process that would expand the way Java developers build integration solutions by providing a standardized container for business integration components as part of the Java platform. With this new architecture, Sun anticipates that business integration will quickly converge with Web services, helping to accelerate their adoption and reduce integration costs.
http://sun.com

Digital Evolution Unveils Complete Web Services Management Platform

(Santa Monica, CA) – Digital Evolution, Inc., a provider of Web service management products, has announced its new DE Management Server 2.0. Designed to help businesses conduct secure computing in an open-standards environment, the new server offers a comprehensive management platform that takes the complexity out of managing and using Web services.

The DE Management Server 2.0 comes pre-integrated with existing enterprise security packages and systems, allowing organizations to leverage existing security infrastructure for the SOA rollout.
www.digev.com

IONA Leads Standards Discussion at Web Services Edge East

(Waltham, MA) – IONA's chief technology officer, Eric Newcomer, led a panel of software industry experts at the Web Services Edge East Conference in Boston. The panel, entitled, "Web Services Architecture: The Next Big Spec, From the Mouths of the W3C Authors" was part of the conference's Web Services Track. The panel discussed how the W3C will shape the future of Web services with a successful architecture specification. With Mr. Newcomer on the panel were Heather Kreger, Web services lead architect for emerging technologies at IBM; Michael Champion, lead research and development specialist at Software AG; and David Booth, senior research architect at W3C.
www.iona.com

Sybase Delivers Technical Session on Achieving Information Liquidity

(Boston) – Robert Breton, senior director of product strategy for the e-Business Division of Sybase, delivered a presentation at the Web Services Edge 2003 East conference, entitled "Achieving Information Liquidity through Web Services." Sybase defines "Information Liquidity" as the efficiency with which a company transforms data into economic value. Attendees learned how this concept is guiding businesses to gain maximum value from their IT investments through Web services.
www.sybase.com

World's First SOAP/MIME File Transfer Web Service Released

(Boston) – FileUp Enterprise Edition (FileUpEE), the world's first SOAP/MIME file transfer Web service, was released by SoftArtisans at the Web Services Edge 2003 East – International Web Services Conference & Expo. SoftArtisans also announced the release date for WordWriter for .NET, their newest product for creating Microsoft Word documents on the Web.

With FileUpEE, multigigabyte files, even those in excess of 100GB, are securely transferable within a Web farm environment. FileUpEE scales to the most demanding Web sites, consisting of numerous server farms and multiple levels of security. Content is secured at both the Web server and file server level, ensuring high availability and redundancy. FileUpEE File Servers can be isolated and protected far beyond any other method available today.
www.softartisans.com

Parasoft Provides Enhanced Automated Error Prevention Tool

(Monrovia, CA) – Parasoft has announced a preview of SOAPtest 2.0, its automated testing tool for Web services. The new version, due out in May, was previewed at the Web Services Edge East show in Boston.

SOAPtest's automated technologies help development teams prevent errors by performing server functional testing, load testing, and client testing with just the click of a button. In addition, developers can also use SOAPtest as a proxy server to view and verify messages between a client and a Web service.
www.parasoft.com/soaptest

Altova and DataPower Team to Deliver XML Web Services Security

(Beverly, MA and Cambridge, MA) – DataPower Technology, Inc., provider of intelligent XML-aware network infrastructure, and Altova, Inc., provider of XML software tools solutions, have announced the availability of XMLSPY 5 integrated with the Datapower XS40 XML Security Gateway. The unified solution addresses the need for centralized XML Web services security without forcing application developers to alter pre-existing design and deployment practices in any way.
www.altova.com, www.datapower.com

Actional Unveils Web Services Management Server and Console

(Mountain View, CA) – Actional Corporation has unveiled the Actional Looking Glass Web services management server and console. The new offering enables organizations to reduce the time and cost of managing the impact of change inherent in dynamic Web service networks.

WSJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
.NET Developer's Journal	www.sys-con.com/dotnet	888-303-5282	37
Altova	http://xmlj.altova.com/authentic5		17
Ektron	www.ektron.com/ws		9
Global Knowledge	www.globalknowledge.com	1-800-COURSES	19
IBM	www.ibm.com/websphere/portalplay		2
Java Developer's Journal	www.javadevelopersjournal.com	888-303-5282	57
JDJ Store	www.jdjstore.com	888-303-5282	35
JDJ Workshop	www.sys-con.com/education	201-802-3058	45
Linux Business & Technology	www.sys-con.com	201-802-3020	51
Macromedia	www.macromedia.com/go/cfmxad		23
Mind Reef	www.mindreef.com	603-465-2204	4, 5
Parasoft	www.parasoft.com/wsj2	888-305-0041	7
Quest Software	http://java.quest.com/jcsr/ws	800-663-4723	11
SpiritSoft	www.spiritsoft.com/climber		59
SYS-CON Media	www.sys-con.com/suboffer.cfm	888-303-5282	46, 47
SYS-CON Media	www.sys-con.com	888-303-5282	43
Web Services Journal	www.wsj2.com	888-303-5282	39

Advertiser is fully responsible for all financial liability and terms of the contract executed by their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

Actional Looking Glass provides a centralized control console combined with a powerful management server that enables users to quickly visualize, understand, monitor, and manage complex Web service networks.
www.actional.com

Actional Introduces Web Services Initiative, Partners with Microsoft

(Mountain View, CA) – Actional Corporation has announced a detailed initiative to deliver solutions that allow organizations to minimize the impact of constant change inherent in dynamic enterprise Web services environments. As part of this initiative, Actional is announcing a series of new products, strategic partnerships, and customers, demonstrating its market momentum and ongoing commitment to helping organizations realize the full potential of their Web services deployments.

To help deliver on its commitment to helping customers maximize the value of their Web services deployments, Actional is announcing a strategic technology, consulting, marketing, and sales agreement with Microsoft Corporation. The two companies will jointly market and sell their solutions promoting their combined strengths. In addition, James Phillips, senior vice president of marketing and product management at Actional Corporation, has been selected to participate as a member of Microsoft's Infrastructure Advisory Council.
www.actional.com, www.microsoft.com

Altova's AUTHENTIC 5 Has 200,000 New Users in 2 Weeks

(Beverly, MA) – Altova Inc., producer of XMLSPY, has announced the wide-spread adoption and success of their recently released free XML document editor, AUTHENTIC 5. When AUTHENTIC 5 became publicly available under a free license in February, Altova generated approximately 200,000 new users worldwide in 2 weeks.

AUTHENTIC 5 is available immediately for free download at www.altova.com/download_authentic.html and is now offered under a free software license.

Among their users are Agile.Net, CarsDirect.com, the Pocumtuck Valley Memorial Association/ Memorial Hall Museum in Old Deerfield, Massachusetts, the University of Regensburg's MedicMed Project, Oxford Analytica, and UC Irvine.
www.altova.com

Eric Pulier, CEO, Digital Evolution

SYS-CON Radio Host Jeremy Geelan

Alex El Homsi, President and CEO, Trilog

Kieran Taylor, Director of Product Marketing, DataPower

Joshua Gondwe, Account Manager, Jinonet Software

Claire Dessaux, Director Oracle Internet Platform, Oracle Worldwide Marketing

Eric Newcomer, CTO, IONA Technologies

Michael Kuhbock, Chairman, EIA Industry Consortium

Nigel Thomas, Director of Product Marketing, SpiritSoft

Spiritsoft

www.spiritsoft/climber

Swingtide

www.swingtide.com/testdrive