



Business be
nimble,
business be
quick

By Elizabeth Dougherty

JUMP OVER INTEGRATION HURDLES, SAVE MONEY, AND STAY AGILE WITH WEB SERVICES

Web services have attracted a lot of attention over the past couple of years, and for good reason: Web services hold tremendous potential to connect heterogeneous systems without draining the life out of IT resources. But all of that pie-in-the-sky hype sometimes distracts from the important story: Web services have practical applications—such as point-to-point integration—that can revolutionize the way enterprises do business and also save big dollars.

Web services enable communication among programs that are written in different languages and are running on different hardware platforms and operating systems. Sure, we can use software and customized adapters today to make our systems play nice, but any CIO can lament the mountain of time and money that such integration projects typically require. The standardized communication of Web services, on the other hand, helps businesses accomplish integration with faster development, increased reliability,

and huge cost savings. “It’s a story the IT manager loves to hear,” says Jason Bloomberg, senior analyst at ZapThink, a market research firm specializing in XML and Web services.

Be like bamboo

Web services can give your company a competitive edge by increasing your business agility—the ability to respond quickly and advantageously to a changing competitive situation. As the old adage says, you should be flexible enough to bend when a big wind comes—but never break.

From established industries to upstarts, Dell™ customers are relying on Web services—often built using Microsoft® .NET-connected software—to stay agile. For example, a coatings, glass, and chemical manufacturer founded more than a century ago automated its business-to-business electronic commerce using .NET-connected software on Dell PowerEdge™ servers. Not even a decade old, an online broker uses

similar technology internally to exchange financial data with its parent company.

Sorting through the hype

Despite all of the press that Web services have garnered, people are still debating their definition—perhaps because the term “Web services” combines two straightforward terms into a concept that is anything but obvious.

From a business perspective, a Web service is a self-contained business function that runs on standards-based Internet technology using everyday Web servers. On a more technical note, a Web service lets you connect two disparate software systems and automate data exchange between them using XML (eXtensible Markup Language). On a less technical note, you can envision a Web service as the translator between pieces of software that have something to say to each other but don’t speak the same language.

Here’s how it works: The Web service producer describes the information available,

what can be done with it, and an authentication process. The Web service consumer communicates with the Web service to obtain the information, manipulate it, and possibly add new information. For example, a tire manufacturer could make available its inventory and production schedule and allow car makers to place orders and schedule deliveries—even linking to a system that warns of an upcoming shortage of snow tires before ski season. The exchange of information happens via messages based on XML standards.

Sounds pretty routine, but the aspect of Web services that makes businesses “ooo” and “aah” is the fact that all of this communication happens automatically and without much need for human intervention. And because Web services use the inexpensive Internet to function, they are suitable for even the thriftiest companies.

“The Internet” is not to be confused with “the public.” Although Web services use Internet protocols, such as HTTP, they are not necessarily accessible to everyone. For now, security limitations constrain the kinds of Web services that businesses can launch. Each company chooses who can access its Web services, and most businesses that use Web services today are deploying them over firewall-protected intranets or with trusted business partners. Early adopters—businesses in the financial services, insurance, and health care industries—deal in some very sensitive data.

Seeing the future

A Web service can be a simple one-way exchange of information, in which a Web

services consumer requests information and receives it. Today, Web services already are sending us stock quotes, weather reports, and sports scores. In the future, a taxi company might ask an airport system for information about arriving flights to help the dispatcher send enough taxis.

A Web service also can be a more complex two-way exchange of information, such as a transaction with multiple steps. For example, a health club could implement a scheduling Web service that allows customers to access their personal trainers' calendars and reserve appointments, or receive an e-mail or pager notification if a slot opens up in Wednesday's tai chi class. Or, when you buy a plane ticket from a travel Web site, you might request that you be notified of all schedule changes, but that dear old mom only receives an e-mail one hour before your flight arrives.

As Web services become more sophisticated, they can ask for information, process it, and deliver it directly to users' desktops. “Unlocking the data isn't enough,” says Neil Charney, director of the Platform Strategy and Partner Group at Microsoft. “You must be able to do something with it.” For example, a major U.S. stock market developed a Web service application that sorts, retrieves, and analyzes financial data before displaying it in Microsoft Excel spreadsheets. Consequently, traders and analysts get faster, streamlined, custom answers to queries that previously required traders to search through paper reports.

How to succeed in Web services

That's how Web services work and how businesses can apply them, but it is only half of the picture. The real story is about the incredible benefits that Web services can deliver—especially if you know the best way to approach them.

Leverage existing resources. One of Web services' greatest strengths is their ability to exploit and connect existing disparate systems, which saves time and money. Because so many systems have the basic components of Web services—a Web server, XML parsing, and SQL database access—it is easy to construct Web services applications that are portable across platforms. Companies also can reuse Web services interfaces to shorten the development time of projects.

Form faster partnerships. An upfront standards-based agreement about how to exchange data in an automated way can eliminate the time-consuming meetings, decisions, and development cycles of past approaches.

Increase flexibility. Web services allow ad hoc connections that companies can build and change quickly in response to changing business needs. Because Web services utilize an existing connection—the Internet—between your company and other companies, new integration projects are significantly less expensive compared with traditional application integration.

Setting the standard:

How conformity helps you compete

Thanks to software vendors playing nicely together and supporting Web services standards, businesses gain the ability to interoperate. This means that when you use industry-standard tools to build your Web services, you can interact with other standards-compliant Web services. Because the method of exchanging data is predefined, you can establish partnerships quickly and easily, boosting your business agility.

Although standards bodies such as the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS) define Web services standards, the backing of industry heavy hitters such as Microsoft, IBM, and Sun Microsystems is crucial to their acceptance. In fact, a group of vendors has formed the Web Services Interoperability Organization (WS-I) expressly to promote Web services interoperability.

"Web services are strategic and important enough that these companies are committing time and energy around building a common understanding and interpretation of them," says Neil Charney, director of the Platform Strategy and Partner Group, at Microsoft.

Web services utilize XML, a general-purpose document format language used for Web documents. Here's a rundown on the alphabet soup of XML components that Web services use:

- SOAP: The Simple Object Access Protocol allows a Web services consumer to communicate with a Web services producer using XML messages
- WSDL: The Web Services Description Language is used to describe a Web service's capabilities
- UDDI: Universal Discovery Description and Integration, the "yellow pages" of Web services, lets you register the availability of your Web services

Simplify development. Development tools, such as Microsoft Visual Studio® .NET 2003, speed development by automating much of the coding typically involved in creating Web services. Also, most developers already know the ins and outs of the Internet markup languages—such as XML and HTML—that they will use to create Web services.

Capitalize on standards. Agreed-upon standards and industry support for those standards are key to the success of Web services. Standards enable Web services from different companies to interoperate, which helps businesses to form and evolve partnerships quickly and easily.

Join industry efforts. Many industries have developed standard XML dictionaries and templates to encourage application compatibility. Industries that have published Web services standards include insurance, human resources, and printing and publishing.

Enjoy a competitive edge. Faster, more cost-effective development means your IT department can better respond to business needs. "The business can now drive the technology instead of vice versa," Bloomberg says.

Getting in the game

Ready to dive into Web services? Experts suggest that you develop a careful, gradual plan.

Invest in training. Whether you are just starting to dabble in Web services or expanding your efforts, educate your IT staff about the Web services capabilities of your current development platform and language. Online training classes can shorten the learning curve.

Investigate what's happening in the IT trenches. Is an informal Web services implementation operating under the corporate radar, for example, within a different department? If so, talk to the developers about what they have learned and brainstorm about other potential applications.

Create a pilot project. If you are new to Web services, get started by creating a one-way Web service that is not mission critical but has some business function. For example, publicize something about your company. If your company is a manufacturer, create a Web service that allows

.NET makes gains

Microsoft's monster commitment to standards gets results

If you have looked online for a job, you have most likely browsed Monster, the number 1 career Web portal. Behind the scenes, Monster relies on Microsoft software to build its giant network, which runs on Dell servers. Like many Microsoft customers, Monster is using .NET, Microsoft's software for implementing Web services, to increase its business agility.

Microsoft has united with partners and competitors, including rivals IBM and Sun Microsystems, to support Web services standards. "They're committed to interoperability, which is a new thing for Microsoft," says Jason Bloomberg, a senior analyst at ZapThink, a market research firm specializing in XML and Web services. "To penetrate the enterprise space, Microsoft realizes it must interoperate. The company is serious about this."

When announcing the .NET strategy several years ago, Microsoft committed to building XML support for Web services across its product lines. In Visual Studio .NET, released February 2002, Microsoft delivered the only development environment built for Web services from the ground up. Since that time, thousands of customers have experienced the benefits of .NET-connected solutions using Microsoft tools, client software, servers, and services.

Beginning with the launch of Windows Server 2003 and the latest edition to the Microsoft Visual Studio family, Visual Studio .NET 2003, Microsoft products that support Web services will begin to display a ".NET Connected" logo. "The .NET Connected logo promotes products and services with Web service-based connectivity that people build on our platform," says Neil Charney, director of the Platform Strategy and Partner Group at Microsoft.

Monster expects the new Visual Studio .NET 2003 to boost developer productivity for Windows Server-based applications. "We're shooting for more than a 30 percent improvement," says Brian Farrey, president of TMP Technologies, a division of Monster's parent company, TMP Worldwide.

Although Monster has been using XML for a couple of years, integration of XML with the operating system will make it easier to build Web services to interact with its larger enterprise customers. "We can more seamlessly integrate our offerings with them," Farrey says.

Faster development times and lower integration are compelling reasons to consider implementing Web services. Staying agile in the marketplace is another. "You may not be thinking of it," Charney says. "But your competitor is."

customers to access your product catalog. This process can give your IT staff confidence and provide a working example upon which to build.

Evangelize. Educate employees outside the IT realm about the capabilities of Web services and enlist their help in finding the most profitable business applications for Web services. Check with major suppliers and customers to see if they offer Web services that you can use to streamline your business relationships. The more people who understand what the technology can do, the easier it is to use Web services as a business tool.

Tackle the next challenge. By being proactive with your Web services strategy, you will not only be ready to react to change, but you will proactively move your company's IT infrastructure forward and help improve its bottom line.

Building on success

As your company puts a Web services strategy in motion, keep in touch with your current software vendors. Also keep an eye on some of the new companies that specialize in Web services. Watch for software vendors to improve development offerings and add management tools as the technology and the standards mature and evolve.

For example, companies are refining security—an important next step to making Web services truly transparent. Automated authentication of

both parties is an aspect of Web services that still needs to be standardized. Web services currently use HTTP over Secure Sockets Layer (SSL) encryption, which ensures private communications but does not guarantee that communicating parties are indeed who they say they are.

The initial deployment of Web services targeted point-to-point integration of existing systems, because that is where businesses can get a measurable return on their investments. This area will continue to grow because it saves time and money by allowing companies to leverage existing resources.

As larger companies compel their partners to support Web services standards, these requirements will drive Web services deployment between companies. Don't expect Web services to be open to public consumption in the short-term future, but that possibility may eventually take shape after more companies implement Web services and become confident in their security.

Change may be good or change may be bad, but change in the business world is certainly inevitable. That's why—whether you're just getting acquainted with Web services or expanding your initiatives—the most important point to remember is that Web services can help your business perform well in the face of inevitable change. Take advantage of Web services technology and enjoy the rewards of being agile. Then, when the heat is on, you can relax knowing that your business can leap higher than the hurdles it faces. **D**

For more information:
www.microsoft.com/net
www.webservices.org
www.ws-i.org