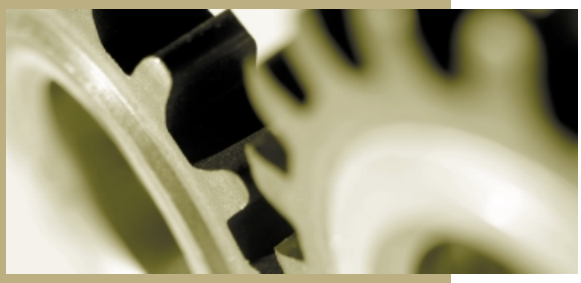


Web Services 2002

Market Milestone Report

Featuring a Delphi Group Assessment of: IBM

Contents

IBM Corporation
1133 Westchester Avenue
White Plains, NY 10604

www.ibm.com

Fever Pitch

The most passionate supporters of Web Services have proclaimed it as "bigger than the Internet" as well as the answer to every CIO's prayers and the one technology to finally bridge the gap between IT and business. Others have taken a more pragmatic view, casting Web Services as over-hyped, overblown, and perhaps just one more excuse to sell new software. The truth is that both may be right. Web Services offer nearly limitless potential to change the way software is deployed and managed, yet the near-term reality will undoubtedly fail to meet the lofty expectations set by its most feverent proponents.

Regardless of the steadfast positions held by both extremes, Web Services hold the promise to solve many, often recurring, business problems. From delivering information between applications within disparate environments to enabling the cost-effective reuse of exisiting technology assets, Web Services will play a large role in making organizations more efficient. Certainly, there are still many issues to be resolved including, but not limited to, interoperability, security, and revenue models, just to name a few—but the future of Web Services is bright.

What are Web Services?

Delphi defines Web Services as business assets that can be shared, combined, used, and reused by heterogeneous computing resources within an organization or between firms. Technically, a Web Service is an XML object comprised of content, application code, process logic, or any combination of these, that can be accessed over any TCP/IP network using the SOAP standard for integration, the WSDL standard for self-description, and the UDDI standard for registry and discovery within a public or private directory.

From a purely technological standpoint, a Web Service is a simple enough thing, but from a business perspective, Web Services get more confusing. The business asset could be an application or a component of an

application, a task within a process, or a whole process packaged together. The business analyst needs to know how to apply a Web Service to create a business benefit. In this report, Delphi will highlight not only the critical business issues surrounding Web Services, but also where Web Services will be applicable from a business benefit perspective.

Understanding Software as Services

In order to understand how Web Services will affect enterprises, it is imperative to understand the function of software as services. A services-based framework implies that every Web Service brings together a number of software components and manages their relationships without the need to know the specifics (i.e. underlying language) of each component. Since only the business-level services are searched for and used within a Web Services framework, a business service-based directory can be used.

If you are wondering exactly what a service-based directory is, don't fret. There are a number of issues that users must first understand before undertaking an evaluation of the Web Services market.

Standards

Conceptually, Web Services will deliver on the promise of interoperability—the ability for components created in different programming languages to work together as if they were created using the same language. This interoperability will be provided through industry standards that have been emerging throughout the past few years. The holy trinity of Web Services standards is SOAP (Simple Object Access Protocol), WSDL (Web Services Definition Language), and UDDI (Universal Description, Discovery, and Integration). There are several other emerging standards of which to be aware, among them XML, and all of which deal with transacting data from disparate systems over the Web. Please see the sidebar on pages 5 and 6 for a more detailed discussion of Web Services standards.

The standards effort in Web Services is led by a consortium of companies. There are several standards organizations out there that count among their members leading organizations that are trying to pave the way towards acceptance of common standards. Several of these standards organizations are doing critical work in the area of Web Services. Among them are:

- WS-I—promoting Web Services interoperability across platforms, operating systems, and programming languages
- OMG—producing and maintaining computer industry specifications for interoperable enterprise applications
- W3C—developing interoperable technologies to exploit the Web as a forum for information, commerce, communication, and collective understanding
- OASIS—designing and developing industry standard specifications for interoperability based on XML

The work of these organizations is not to be overlooked. The critics of Web Services point to true interoperability as the biggest issue holding back Web Services. For technology providers to provide real interoperability that will have a business impact, these standards organizations will have to gain traction with leading vendors and enterprises.

Required Services Beyond Standards

Standards are certainly important to the adoption of Web Services technology. There are, however, several other issues that decision-makers will need to pay keen attention to as they evolve Web Services initiatives for their enterprises.

Development

Before a complete discussion of Web Services development can begin, it is necessary to make the critical decision as to the framework for a developoment environment. Today, an enterprise can choose between the J2EE and .NET frameworks. Much has been written and debated about the "enterprise wars" between these competing framework. In Delphi's view, the major advantage to .NET is that Web Services are architected in from the ground up, whereas APIs are retrofitted to allow XML communication for the J2EE framework. Microsoft has no competitors with the .NET framework, though, while J2EE vendors are focused on keeping products interoperable and non-proprietary—this could be an important consideration down the road should Microsoft be seen as having a monopoly.

Integrated Development Environment

Before a Web Service can exist, it needs to be created. Each Web Service is created in a development environment, typically referred to as an integrated development environment, or IDE. For the .NET framework, the IDE is Visual Studio.NET. Several vendors provide IDEs for the J2EE framework, including Sun Microsystems with the Forte product.

What the IDE does is provide custom modules to create applications or services within a given platform. Thus, Visual Studio.NET is the tool for developers to create services within the .NET framework, while Forte is one of several choices for doing the same within the J2EE framework.

Web Services are created in an IDE like VisualStudio.Net or Forte and then deployed on an application server like BEA's WebLogic, IBM's Websphere or Fujitsu's INTERSTAGE. In order to use a Web Service once it has been created, a user needs to know that it exists and be able to find it.

Discovery

Web Services will enable the dynamic discovery and use of applications on the Web. For discover to occur, software will search UDDI directories examine specifications of Web Services, and construct applications from a composite of services that are found in these directories. No application will need to be hard-coded to use other applications, rather the software will understand the need to find and use Web Services dynamically—making for a more flexible software infrastructure.

How exactly will this discovery happen? One way will be to utilize USML (UDDI Search Markup Language), an XML-based protocol for carrying a search request, including multiple queries, key words, UDDI sources, and aggregation operators. Another method will be to use Web Services Inspection language, an XML format for assisting in the inspection of a site for available services and a set of rules for how inspection related information should be made available for consumption.

Dynamic Discovery

One of the major technological benefits of Web Services with respect to discovery will be the enabling of dynamic discovery. Dynamic discovery can refer to the exposure of a collection of services that live together under the same URL. Such service exposure is called dynamic because the number and types of services in the collection existing at the endpoint can change over time, but clients need only know the one URL in order to access any or all of them despite any changes to the collection. Alternatively, dynamic discovery can refer to systems in which clients search through registries to first discover and then invoke services supporting the capabilities they require. Such systems are described as dynamic because the clients supposedly have no prior knowledge of the services they're searching for.

Dynamic discovery is not as easy as it sounds. Clients and services must share common definitions of the concepts involved in the Web Service that might be shared. A lack of that shared information could mean that a Web Service would fail to interact with other services in the way it is

intended. As standards evolve, they will begin to define those common concepts.

Discovering Web Services will be one challenge; Guaranteeing the security of Web Services will be another major challenge.

Security

Web Services detractors often point to security as the gaping hole in the concept of Web Services. It is not a point to be taken lightly—enterprises still get hacked into over the Web at an alarming rate.

Authentication provides a guarantee that the sender of a message in a Web Service is who they claim to be and provides a guarantee that the message is what it is intended to be and has not been tampered with. It is imperative that Web Services provide authentication of both the sender and receiver of the message in a Web Service interaction.

Confidentiality provides a guarantee that information meant for a particular party is not visible to unauthorized parties. Typically, confidentiality has been achieved by encrypting messages with keys. In the Web Services world, automatic encryption and decryption will be a must-have in any solution.

Current Security Initiatives

Several initiatives to address security issues in the Web Services realm head-on are underway. Microsoft's Passport and Sun's Liberty Alliance are the most recognizable of these initiatives. Both initiatives deal with the concept of federated authentication—the process of utilizing a single-sign-on (SSO) to access multiple Internet Web sites instead of requiring the user to establish an account at each site.

Microsoft PassPort

Microsoft .NET Passport provides users with single-sign-on and purchasing capabilities at participating sites. Passport users can create a single sign-in name and password for use across participating .NET Passport sites. The SSO service provides a common Internet authentication mechanism across participating Web sites. The SSI service also allows users to avoid repetitive data entry by storing a limited set of basic demographic information that can be shared with participating .NET Passport sites when signing into those sites.

Passport users can also create a wallet that stores their billing and shipping information in a secured location. Consumers can then make online purchases at participating .NET Passport express purchase sites by signing in to their wallet and sending their purchase information to the merchant with one click and without the need to retype any information.

.NET Passport uses encryption technologies, such as Secure Sockets Layer (SSL) and the Triple Data Encryption Standard (3DES) algorithm, for data protection. The issue of privacy becomes key with this type of service, because users are essentially storing information online so that they don't have to remember it. .NET Passport addresses privacy by having all participating sites sign a contract in which they agree to post and follow a privacy policy that adheres to industry-accepted guidelines.

Passport will use the Kerberos standard to conduct open and federated authentication among organizations. This is intended to facilitate a trusted, interoperable authentication network across the Internet, bring universal single sign-in to all users and provide interoperability among different enterprise and service authentication systems.

The goal of the new Passport federated functionality is to make it easier for corporations to securely bridge existing authentication mechanisms with external systems and thus simplify the lives of constituents.

SOAP (Simple Object Access Protocol)

A protocol for exchange of information in a decentralized, distributed environment. It is an XML-based protocol that consists of three parts—an envelope that defines a framework for describing what is in a message and how to process it; a set of encoding rules for expressing instances of application-defined datatypes; and a convention for representing remote procedure calls and responses. SOAP is a key standard for delivering Web Services.

WSDL (Web Services Definition Language)

An XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

UDDI (Universal Description, Discovery, and Integration)

A platform-independent, open framework for describing services, discovering businesses, and integrating business services using the Internet.

XML (Extensible Markup Language)

A form of self-describing data that creates common information formats in order to share both the format and the data across the Internet, intranets, and other networks. XML frees Internet content from the browser, making it available to real applications.

WSFL (Web Services Flow Language)

An XML language for the description of Web Services compositions. WSFL considers two types of Web Services compositions. The first type specifies the appropriate usage pattern of a collection of Web Services, in such a way that the resulting composition describes how to achieve a particular business goal; typically, the result is a description of a business process. The second type specifies the interaction pattern of a collection of Web Services; in this case, the result is a description of the overall partner interactions.

ebXML

A modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet. Using ebXML, companies now have a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes.

WSRP (Web Services for Remote Portals)

Visual, user-facing web services-centric components that plug-n-play with portals or other intermediary web applications that aggregate content or applications from different sources. WSRP defines a web services interface description using WSDL and all the semantics and behavior that web services and consuming applications must comply with in order to be pluggable as well as the meta-information that has to be provided when publishing WSRP services into UDDI directories. The standard allows WSRP services to be implemented in very different ways, be it as a Java/J2EE based web service, a web service implemented on Microsoft's .NET platform or a portlet published as a WSRP Service by a portal. The standard enables use of generic adapter code to plug in any WSRP service into intermediary applications rather than requiring specific proxy code.

WSRP services are WSIA component services built on standard technologies including SOAP, UDDI, and WSDL. WSRP adds several context elements including user profile, information about the client device, locale and desired markup language passed to them in SOAP requests. A set of operations and contracts are defined that enable WSRP plug-n-play.

WSIA (Web Services for Interactive Applications)

Formerly known as WSCM (Web Services Component Model) The purpose of the OASIS Web Services for Interactive Applications (WSIA) TC is to create an XML and web services centric framework for interactive web applications; harmonize WSIA as far as practical with existing web application programming models, with the work of the W3C, emerging web services standards, and with the work of other appropriate business information bodies; ensure that WSIA applications can be deployed on any tier on the network and remain target device and output markup neutral; and promote WSIA to the status of an international standard for the conduct of XML and Web Services based web application development, deployment and management.

RosettaNet

RosettaNet is a non-profit consortium of more than 400 of the world's leading Information Technology (IT), Electronic Components (EC), Semiconductor Manufacturing (SM) and Solution Provider (SP) companies working to create, implement and promote open e-business process standards.

RosettaNet is named after the Rosetta Stone, which, carved with the same message in three languages, led to the understanding of hieroglyphics. RosettaNet, like the Stone, is breaking language barriers and making history.

By establishing a common language — or standard processes for the electronic sharing of business information — RosettaNet opens the lines of communication and a world of opportunities for everyone involved in the supplying and buying of today's technologies. Businesses that offer the tools and services to help implement RosettaNet processes gain exposure and business relationships. Companies that adopt RosettaNet standards engage in dynamic, flexible trading-partner relationships, reduce costs and raise productivity. End users enjoy speed and uniformity in purchasing practices.

RosettaNet Partner Interface Processes " PIPs " are specialized system-to-system XML-based dialogs that define business processes between trading partners. Each PIP specification includes a business document with the vocabulary, and a business process with the choreography of the message dialog. PIPs apply to the following core processes: Administration; Partner, Product and Service Review; Product Introduction; Order Management; Inventory Management; Marketing Information Management; Service and Support; and Manufacturing.

The RNIF Core Specification provides exchange protocols for quick and efficient implementation of RosettaNet standards. The RNIF specifies information exchange between trading-partner servers using XML, covering the transport, routing and packaging; security; signals; and trading partner agreement.

Web Services Choreography Interface (WSCI)

WSCI describes the flow of messages exchanged by a Web Service in a particular process, and also describes the collective message exchange among interacting Web Services. One of the key benefits of WSCI is that it bridges the gap between business process management and Web Services by describing how a Web service can be used as part of a larger, more complex business process. As of the publishing of this research paper, the WSCI standard was in a public review period. After the public review period is complete WSCI will be submitted, on a royalty-free basis, to an industry standards body.

Liberty Alliance

Sun Microsystem's Liberty Alliance was set up in response to Microsoft's Passport, with its goal being the same—to solve authentication problems for users logging on to Web Services. The technologies that make up the Liberty Alliance will have benefits on both the vendor/corporation side as well as the consumer side. For vendors and corporations, these technologies will assist in verifying that users are who they claim to be. For consumers, these technologies will ease the process of logging into secure public Web sites, applications, and any network resource created with Web Services authentication in mind.

The Liberty Alliance Project is a business alliance formed to deliver and support an identity solution for the Internet that enables single sign-on for consumers as well as business users in an open, federated way.

The view of Alliance members is that federated identity will facilitate federated commerce. Thus, a person's online identity, their personal profile, personalized online configurations, buying habits and history, and shopping preferences are administered by the users themselves, yet securely shared with the organizations that the user chooses. A federated identity model will allow every business or user to manage their own data, while ensuring that the use of critical personal information is managed and distributed by the appropriate parties, rather than a central authority.

The Liberty Alliance's role is to support the development, deployment, and evolution of an open, interoperable standard for network identity. The stated goals of the Liberty Alliance Project are:

- To allow individual consumers and businesses to maintain personal information securely
- To provide a universal open standard for single sign-on with decentralized authentication and open authorization from multiple providers
- To provide an open standard for network identity spanning all network devices

Other Initiatives

While the .NET Passport and Liberty Alliance are the most public of the Web Services security initiatives, there are others. AOL Time Warner has been developing its own authentication service code-named "Magic Carpet." Officially launched as the Screen Name Service, the service combines the screen name sign-ins of AOL's America Online, CompuServe 2000, AOL Instant Messenger, Netscape, and NetBusiness into one unified authentication system. It is interesting to note that AOL is a member of the Liberty Alliance; it is not yet known if the Screen Name Service will be compatible with the Liberty Alliance.

Another security initiative that exists in the Web Services realm is XKMS (XML Key Management Specification). This effort is led by Microsoft and VeriSign to integrate public key infrastructure (PKI) and digital certificates (which are used for securing Internet transactions) with XML applications. The concept is to delegate the signature processing to a "trust server" on the Web, so that thin clients don't have to have the innate ability to do this themselves. XKMS relies on the XML Signature specification already being worked on by W3C (one of the standards organizations discussed earlier) and on anticipated work at W3C on an XML encryption specification.

If a Web Service can be created, discovered, and secured, then the next step is to use it to complete a task or a process.

Orchestration

As with all technology segments, the Web Services segment has evolved its own vernacular. When it comes to utilizing services, the common term is orchestration. That term works well, as it brings up the context of a symphony, where different instruments must work together in harmony.

Relation to Process

Typically, Web Service orchestration occurs in the context of a process, whether that process involves

the simple exchange of data between two applications or is a complicated stock transaction deal. The act of orchestration involves defining and executing the logic and rules that assemble multiple synchronous and asynchronous web services into a long-lasting multi-step business process. Thus, once a service has been created, it needs to be consumed in some way. Orchestration defines how a service will be consumed.

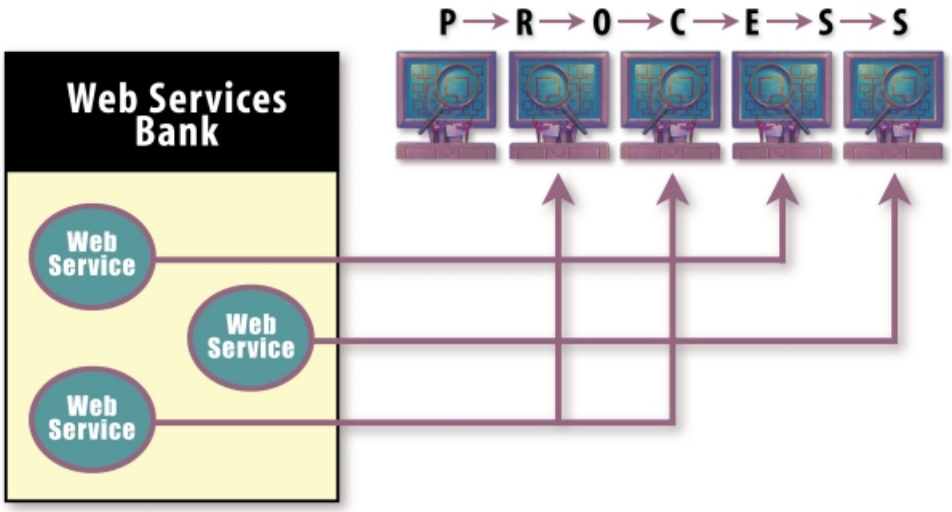
Business Versus Technology

As with every other piece of enterprise software, there is a technological side to orchestration and a business side, too. On the technology side, there are Web Services orchestration specifications that exist; two of the better known are Web Services Flow Language (WSFL) and eXtensible LANGuage (XLANG).

WSFL is an XML language for the description of Web Services compositions. It considers two types of compositions. First, flow models specify the appropriate usage pattern of a collection of Web Services, in a way that describes how to achieve a particular business goal. Second, global models specify the interaction patterns of a collection of Web Services, with the result being a description of the overall partner interactions.

XLANG is a notation for the specification of message exchange behavior among participating Web Services. It is widely thought that XLANG will serve as the basis for automated protocol engines that can track the state of process instances and help enforce standard protocol in message flows. At its core, XLANG is an XML business process language that provides a way to orchestrate applications and XML Web Services into larger-scale, federated applications by enabling developers to aggregate large applications as components in a long-lived business process. XLANG relates to WSDL in two ways. First, an XLANG service description is a WSDL service description with an extension element that describes the behavior of the service as a part of a business process. Second, XLANG service behavior may also rely on simple WSDL services as providers of basic functionality for the implementation of the business process.

On the business side, the concept of Business Services Orchestration (BSO) wraps the process context around the orchestration of services—BSO is a process-driven method for managing business services created both internally and externally. A BSO solution would make use of an orchestration specification, such as WSFL, XLANG or potentially through the recently announced standard WSCI (see glossary on pages 5-6).



As Illustrated within the diagram above, a typical instance of BSO would involve a steps in a process invoking specific Web services from a bank of available resources. The process engine would capture state information, and determine when to envoke or destroy short-lived services. Orchestrating Web Services does not end the challenge of managing them. It is imperative that organizations also make sure that their solutions cover transaction support.

Transaction Support

Any organization uses complex business processes to tie in with disparate applications from the many departments in the enterprise. These organizations are interested in learning how Web Services can work within a business process along with transaction support, routing of messages, etc. The concern is that most of the Web Services platforms available today provide a quick way of Web Service-enabling existing applications and invoking that functionality. Business processes however automate a set of operations that the organization performs. These operations could encompass several discrete Web Services, invoke other applications, and perform transactions all as part of the business process. Hence it is important for an organization to understand how Web Services would seamlessly fit into previously defined business processes.

Thus, applications will need to be able to transact with each other so that information can be changed by one and shared back to another. One example is a Web Service that does the function of credit card validation. After validating a credit card number, the balances of two accounts need to be changed. Dollars must be debited from one account and credited to another. Should something happen within the process that is not right, the process needs to be rolled back so that dollars are not created or lost along the way.

Operations Management

Like any other technology within an enterprise, Web Services must be both reliable and manageable. A Web Service is considered reliable if it works in the way that it is intended to work. Any messages passed back and forth during a Web Service lifecycle should be delivered to the correct recipients, in the correct amount of time, in the correct order, and the correct number of iterations. If a message were to be incorrectly delivered for any reason, then there must be functionality that alerts both the sender and recipient to the circumstances.

The important issues to consider with regard to reliability are guaranteed delivery, non-repudiation, and "once-and-once-only delivery." A guarantee that a message is sent over the Internet as part of a Web Services interaction and that it reaches the intended recipient is an absolute necessity. Time-outs and resends need to be handled automatically and transparently. Guaranteed delivery also means that an application can make a request to send a message and not have to write state-keeping routines that check whether a message has reached its intended destination with the intended results.

With guaranteed delivery in place (and security, of course), it is time of focus on non-repudiation. This means guaranteeing that a recipient cannot repudiate a message that the recipient received or a sender cannot repudiate a sent message. Non-repudiating adds a sense of accountability to Web Services. It is also necessary to ensure that a message that is intended to be delivered only once is not delivered in perpetuity, but rather only the one time it is intended to be delivered.

As Web Services are deployed throughout an enterprise, managing access to them, tracking their usage, logging the services performed, potentially billing the users for the service—these become challenges that will need to be solved. Real-time knowledge of the state of a Web Service is required. The importance of tracking increases if messages are queued asynchronously for delivery. Organizations will also need continuous visibility into the performance metrics of Web Services requests and replies, endpoint states, and message delays. Every event happening at the requester end, the provider end and in-between while invoking a Web Service has to be logged and easily visible to the managers of the Web Service. In addition, when deploying Web Services, organizations will need to pay close attention to user roles—giving different users various levels of access to different Web Services.

Commercial Interchange Models

The hype surrounding Web Services has been partially driven by the promise that Web Services will fundamentally change business models. In the short term, Web Services will serve as a tool to optimize business operations. The business models of financial services institutions and pharmaceutical firms are not going to change overnight due to Web Services. There are, however, interesting business models that will grow around the concept of Web Services.

Several niche vendors are addressing market needs that are evolving as Web Services gain traction. For example, there are small firms developing capabilities for monitoring services available in UDDI registries. Eventually, these firms will offer services for monitoring availability and inventory of services, usage monitoring, and billing services. Other firms are tracking services in order to provide a guaranteed delivery service, in essence becoming the Fedex of the Web Services world. And, in time, Web Services brokerages will open for business, creating a market for services that are developed by companies that don't have the resources to market and sell their own services.

Global Grid

As Web Services become more a reality over the next few years, computing power will gradually become more of an issue. Grid computing is a potential source of that computing power. The Global Grid is a worldwide project to organize distributed processing by creating a grid of networked computers using the Internet and Web Services. The goal is to build a processing grid that subscribers can plug into and use as needed, just as we do today with the electrical power grid.

The most prominent example of grid computing applicability is the SETI project, where a network of computers lend processing power during individual PC downtime to the search for extraterrestrial life. Delphi research shows that a significant amount of enterprises (42%) are not familiar with the Global Grid. In contrast, though, 37.5% of organizations view the Global Grid will be important or imperative to their Web Services strategy. The continuing globalization of business is one of the primary reasons the availability of a worldwide computing utility is viewed as being so important. Another key driver is the desire to minimize costs associated with the acquisition and use of computing power. Web Services and the Global Grid present a platform from which to respond to these market drivers.

Web Services Networks

Web Services networks (WSN) bring practicality to the deployment of Web Services. They provide the infrastructure and services that the requesters and providers of Web Services need to conduct business. These services include non-repudiation of messages, guaranteed delivery, "once-and-once only" delivery of messages, encryption of messages, and authentication. In addition, Web Services networks also provide a registry of categorized Web Services available to meet the needs of requesters and providers, thus providing a framework to search for available Web Services. WSNs also provide the ability to manage application-to-application interactions.

Web Services in Action

Web Services promise to make the lives of businesspeople easier. The best way to understand how Web Services will do this is to analyze an example. One of the best examples of the impact Web Services could have is on the corporate portal.

The portal is the desktop to the businessperson—a single, unified interface to the various pieces of information necessary to conduct business. A user can order supplies, make travel plans, check their calendar, and send off an email to a colleague all through the portal. What has typically made this possible is a "portlet," which allows information from different applications or Web sites to appear in windows within the portal. In the portlet model,

the user is interacting directly with an application or a Web site, but just happens to be doing so in the portal interface.

The Web Services paradigm profoundly affects how the user interacts with the information and Web sites, with minimal impact to the user. Take a businessperson using his/her portal interface to make a flight reservation. In the portal model of the last few years, the user might see the Travelocity interface in one window of the portal. The user would then use that interface to make flight plans. With Web Services, though, the portal is not forced to simply display the Travelocity content in a window, or portlet. The Travelocity capabilities can be delivered as a service. In this model, the user would simply see their portal interface, type in their travel preferences, and that information would travel as a service to Travelocity, which would then deliver back the reservations as a service. The user is not required to use Travelocity interfaces or even to understand that Travelocity is providing the service.

This is just one of myriad number of potential Web Services examples. In this example, a whole process (Travelocity reservation) is wrapped as a Web Service and invoked in another process (corporate travel reservation). As more and more services, both simple and composite, are developed and publicly registered, the business opportunities for leveraging those services will increase exponentially.

Where Web Services Can Impact the Enterprise

As with all emerging technology paradigms, Web Services must prove their benefit to the organization. Web Services will benefit both the business and IT sides of the organization—and in the process, help to bridge the communication gap that so often exists between business managers and IT managers.

Business Benefits

On the business side, Web Services will facilitate "on-the-fly" application development, process optimization, and "codeless" development. Business managers will have the components of applications and processes at their fingertips for the purposes of creating applications that will help them do their jobs more efficiently. Gains in process efficiency will become increasingly evident as managers become accustomed to using Web Services as a tool to improve process performance. All of this will be available to businesspeople without having to possess any programming knowledge. Suddenly, businesspeople will be able to perform functions that they used to have to rely on IT managers for. In many cases, this is where the communication breakdown between business and IT occurs. Web Services will play a large role in smoothing that communication.

As Web Services begin to gain traction in IT departments, enterprises can expect several business benefits:

- Improved operational efficiency through inter-application communication
- Lower operational costs due to the use of existing infrastructure and the minimal infrastructure needs of Web Services
- Improved integration of the value chain, both internally and externally, because of the ease of exposing internal processes to trusted business partners
- Improved organizational flexibility (and thus responsiveness and innovation) through the exposure of services to new and existing channels.

Technology Benefits

IT executives will also reap benefits from Web Services. Web Services will help enterprises to

capitalize on component architectures in a way that has heretofore not been possible. Most large organizations spent the 1990s transitioning from deep legacy software to some degree of adoption of enterprise applications. These packages promised to solve a variety of linked business problems in a single package, primarily in the area of transaction management and coordination. As organizations found out, however, deploying such software often involved expensive and time-consuming customization. Conversely, the component architecture of Web Services will help to eliminate some of the time and expense associated with application development.

Within any enterprise, it is increasingly necessary for application to "talk" with each other. Web Services will provide the IT executive with language-independent connectivity between applications. The IT executive will especially like the ability to wrap legacy applications in a standardized, consistent, and reusable format—doing so will allow for improved utilization of existing infrastructure investments.

Finally, Web Services will help to eliminate one level of integration by enabling applications to expose functionality directly from the machine on which that application exists, rather than having to add additional infrastructure in the form of servers required by middleware solutions. This will have positive budgetary implications for IT executives.

The confusion around Web Services will begin to dissipate as consumer learn more about issues such as discovery and orchestration and become more secure in the business benefits that can result. Already, organizations are beginning to test the waters of Web Services and move up the learning curve. Delphi's recent survey of over 250 organizations revealed an interesting picture of the Web Services market—from the end-user point of view.

The Software Buyers' Perspective

The Web Services market is evolving quickly—the pace of change can lead to confusion for software buyer. Delphi recently conducted a survey of software buyers from approximately 250 leading organizations. The perspective of these decision-makers points out some of the leading challenges and opportunities ahead in this market.

Clearly, confusion exists in the minds of software buyers as to exactly what Web Services are. There have been many definitions propagated in the marketplace, creating a obvious sense of confusion. There are signs, however, that the "fog" is beginning to lift. Many buyers recognize the standards involved with Web Services, with SOAP being the most recognizable, followed by UDDI and WSDL. This understanding of the importance of standards points to the fact that software buyers comprehend that Web Services need to deliver interoperability of diverse systems to provide real benefits.

Show and Tell

Many cynics point to a lack of case studies as a huge weakness of Web Services technology—they claim that Web Services are not truly being used in enterprises today. This is not entirely correct. A significant amount (72%) of survey respondents indicated that someone in their organization is working with Web Services technology now. In many cases, though, it can be an over statement to say that Web Services are truly being employed. XML is the Web Services technology with which most are working. Very few organizations have hands-on experience with UDDI or WSDL at this point—typically, companies are first using SOAP to experiment behind the firewall.

Of those organizations that are not yet using Web Services, 57% indicated that experimentation would begin once the benefits of Web Services had been made clear to them and once experience-based best practices were captured and shared by other organizations. Another 57% of respondents

noted that software tools facilitating the construction of Web Services must be generally available before they will begin to work with the technology. Interestingly, while respondents want to wait until tools have been on the market for some time, they are also willing to work with point tools offered by Web Services specialists and do not necessarily require a comprehensive platform offering. This attitude points to an opportunity for niche vendors to gain traction with products that are point solutions for users.

Web Services Strategies

Software buyers, while not necessarily knowing everything there is to know about Web Services, certainly understand the impact they can have on business—80% of respondents said that Web Services are important to imperative to their business strategy. This group understands that this technology will not only change the way that computing resources are built and used, but that it will also enable new business models.

One thing that the last decade has taught software buyers is to have a strategy for new technology paradigms. Over a third (35%) of respondents indicated that a Web Services strategy is currently in place within the enterprise. Another 26% will have a strategy formulated within six months. Only 13% foresee strategy development being more than a year down the road. Organizations are predicting that Web Services will light fire quickly, and that they need to be ready when that happens.

Within the enterprise, it is typically the CIO who will function as the Web Services strategy leader. The CIO is a logical choice considering that this is typically the individual best positioned to combine both the technical and business understanding of Web Services needed to set strategy and to oversee execution.

There still exists a group of organizations (15% of respondents) that view Web Services (and all technology, for that matter), as the domain of IT staff other than the CIO. These organizations are behind the learning curve, having failed to realize the importance of bringing together IT and business domain experience. The Web Services technology sector is one at the direct intersection of business and technology.

Standards Recognition

XML is clearly the foundational technology upon which Web Services will rest. The standard was nearly a unanimous choice, indicating that it is a "must have" component of Web Services. In fact, 53% of organizations are processing XML-based information with their business applications. Of those not yet working with XML, the majority (54%) expect to begin doing so within the next six months.

For many organizations, the conversion of legacy information, especially content, into XML is a daunting task that may slow their ability to shift to a Web Services computing model. While content stored in any format may be wrapped in XML to create an information object, there is a large performance difference between packaging content in XML and representing it natively in XML. The lesson to enterprises is that critical, frequently used content must be converted to XML as a first step in Web Services adoption, and all new content should be captured in XML format.

Typically, XML will be used for multiple purposes. Naturally, XML will be used to describe content and document structure with the language (its original purpose as intended by the W3C). Beyond just describing structure, though, early adopters of XML technologies have found additional ways to utilize the language: to control content presentation through the description of graphical templates, which enables a write-once, present-many content strategy (especially relevant to the concept of accessing Web Services on mobile devices); and for the integration of data between internal applications, and between internal and external systems (accomplished by wrapping information with XML code to create self-describing packets).

XML has been in the market's collective consciousness far longer than SOAP, WSDL, or UDDI. Thus, software buyers tend to be more comfortable with XML in concept than with the other Web Services specifications, though their understanding of these standards are increasing. SOAP is widely viewed (64% of respondents) as an important component of the Web Services infrastructure, whereas UDDI and WSDL were not seen as such a high priority. Most organizations will gain initial Web Services experience by deploying them internally to integrate applications, and this can be done using only XML and SOAP. UDDI and WSDL will be useful later within large, multidivisional companies, but do not become as essential until Web Services are shared outside the firewall.

In terms of organizations preparing for application interoperability, approximately 20% of respondents have begun to fit their applications with SOAP interfaces, while another 52% plan to begin this process within one year's time. While SOAP is gaining mindshare and momentum within organizations, UDDI is just at the beginning of the learning curve—only 8% of respondents are official members of the project. UDDI is a technology that is "on the horizon" for companies-with 23% of respondents definitely planning to leverage UDDI and 25% closely monitoring its development.

Grid Applicability

As discussed earlier in the section on grid computing and the Global Grid, one of the features made possible by the Web Services architecture is the ability to break large computing tasks into smaller components, each of which can be sent to another machine for processing, then returned to the original computer. The most public example of this model is the SETI@home project (Search for Extraterrestrial Intelligence). This effort to find extra-terrestrial life, coordinated by the University of California-Berkeley, uses the otherwise idle time of volunteers' desktop computers around the world to process cosmic data gathered by observatory equipment.

Of the organizations surveyed by Delphi, 14% are currently using Web Services for distributed data processing and nearly one-third plan to begin doing so within the next year—these numbers point to a recognition of the potential that distributed computing holds within the enterprise marketplace.

The Global Grid is a worldwide project to organize distributed processing by creating a grid of networked computers using the Internet and Web Services. The goal is to build a processing grid that subscribers can plug into and use as needed, in the same manner as we do today with the electrical power grid.

That 42% of respondents are not familiar with the Global Grid speaks to the fact that we are at the beginning of the adoption cycle for distributed computing; full-scale adoption is most likely three years away, at best. Consistent with a technology in its early stages of adoption, a significant number of respondents are aware of the Global Grid and its potential impact on their business—37.5% of respondents said that the Global Grid will be important to imperative to their Web Services strategy. The continuing globalization of business is one of the primary reasons the availability of a worldwide computing utility is viewed as being so important. Another key driver is the desire to minimize costs associated with the acquisition and use of computing power. Web Services and the Global Grid present a platform from which to respond to these market drivers.

Enterprise Deployment Environment

Much has been made of the enterprise wars: J2EE versus .NET. Software buyers will make the ultimate decision in that battle; so far, they are hedging their bets. The largest group of respondents (41%) is preparing for a mixed environment, meaning they will be able to utilize either J2EE or .NET. Microsoft's penetration into the enterprise can certainly be felt, though, with 36% of respondents indicating that they will prepare to deploy Web Services in a Microsoft-

centric environment (compared to 16% that have committed to J2EE). The enterprise wars are not over; software buyers realize that and are preparing for the future in a way that leverages their existing investments.

Applicability of Web Services

Web Services are widely viewed as a means to extend existing investments in information repositories, applications, and business processes throughout organizations and across extended value chains. One of the many questions not yet answered in the Web Services realm is exactly where Web Services will impact the enterprise.

Over two-thirds of respondents (70%) have a business portal project underway. It makes sense that these organizations will use Web Services to integrate information and applications into these portals, rather than building hard-wired portlets that must be recoded each time an application or information source changes. Additionally, portal deployments are reaching more frequently across the firewall. As this trend continues, the business portal will become a place where individuals, build, publish, access, and use Web Services.

The concept of dynamic object integration is not new to many individuals managing unstructured enterprise content. Most content management applications are adept at creating virtual documents on the fly by assembling disparate content chunks into a single framework. Web Services provide a standards-based means to broadcast, aggregate, and use content, which will replace the proprietary methods currently used by most content management vendors.

Another large group of respondents (39%) will concentrate their Web Services efforts on application-to-application integration. These respondents have struggled with proprietary EAI solutions and need to develop and implement standards-based integration services that will allow them to quickly bind applications internally and to share them with external constituents.

Nearly half of the survey respondents have a BPM project underway in their organization. Process integration is the next focal point for those companies that have successfully integrated content and applications in their computing environment (often through a business portal). As more organizations codify and modify their internal processes and those shared with external value chain members, they will use Web Services to extend XML representations of those processes and to integrate them with content and applications, providing best practice business context to corporate information. As noted in Delphi's 2002 BPM Market Milestone Report, Web Services will nicely complement BPM technology.

Inside-Out Approach

Technology projects have tended to start out as small pilot initiatives before gaining widespread acceptance within the enterprise. Web Services, as a technology sector, is following that same pattern. Two-thirds of respondents will deploy Web Services initially for internal application integration purposes. Half of these projects will be confined to one physical location, but an equal number will involve multiple locations. Organizations will use Web Services internally first, especially as they wait to learn how security issues evolve.

Applying Web Services internally first makes good sense, because these initiatives are simpler than projects involving external constituents. It is far easier to build and maintain consensus around internal efforts, which translates into improved ability to fund and control projects. It won't be long, though, before organizations will be hungry to reap the potential savings that can come from using Web Services outside the firewall.

Necessary Resources

In any technology deployment, an organization needs talented resources to optimize results. The majority of respondents (77%) indicated that they will leverage internal resources to deploy Web

Services. This is appropriate if they have the requisite skills in-house, since the cost of internal resources is generally less than that of contracted skills. Fifty-fice percent of organizations realize that they need more experience with service-oriented architectures, XML, and other Web Services technologies and thus will rely on business partners or third-party service providers to implement projects. For the most part, organizations will use a combination of internal resources and third-party experts—this allows the organization to get its people educated on all the technological issues that go along with Web Services.

Perceived Obstacles

Software buyers note four hurdles to successful Web Services deployments. While no one obstacle is seen as prohibitive by a large number of respondents, there were four hurdles that were cited more frequently than the others.

Foremost among these is inexperience in architecting services. Few respondents (5%) see a lack of trained developers as the most significant problem, but developers cannot do their job well unless the Web Services they are building have a clear and efficient architecture. It is probable that organizations will seek external help in architecting Web Services while using internal development resources to engineer and build them.

Multiple standards implementation methods are also viewed as problematic. Standards are merely technical specifications that have been agreed upon by a governing organization or by the market itself. There is no mandate to implement the specification in a prescribed manner. Without a clear architecture that represents established and proven best practices in place, there are many choices that must be made by a developer of Web Services (or applications). As with any technology, early adopters of Web Services will learn how to architect them by traveling the hard path of trial and error, and followers will learn through the experiences of those that have gone before them.

Two of the largest obstacles to Web Services deployment efforts are issues that can derail any IT project. It is impossible to receive project funding in today's economy unless the initiative demonstrates a relatively significant and rapid return on investment. And it is far easier to build the technology than it is to get people to use it. Web Services are all about the collaborative sharing of resources to reach a business goal, but many corporate cultures are still attuned to hoarding and protecting information and business processes, especially when dealing with other organizations. Individuals and firms must indoctrinate themselves in this new collaborative mindset first if they are to leverage Web Services successfully and reap meaningful ROI.

Web Services Ownership

Over 40% of initial Web Services projects will be funded by senior executives. Most respondents (80%) had indicated previously that Web Services are important if not imperative to their organization's business strategy; therefore, it is only natural that sponsorship would come from senior management. Also, most respondents had indicated that their initial Web Services deployment will be an internal one and the level of intra-organizational cooperation necessary to develop and deploy enterprise scale Web Services requires high-level leadership. Visible, active leadership from senior executives will be even more critical for initial attempts at inter-organizational integration through Web Services.

In nearly half (48%) of respondents' organizations, the CIO, IT Manager, or Line of Business Manager will fund the first Web Services project. Sponsorship by IT and Line of Business managers is best suited to skunk-works projects and pilot deployments linking a few workgroups or departments. As Web Services become more pervasive within these organizations and deployments become mission critical, CIOs and other senior executives will play a more active role in the efforts. As noted previously, the CIO is the person tasked with bridging the business and

technical sides of the organization, so it is natural that they fund many strategic Web Services projects.

Spending Habits

The largest group of respondents (23%) expects to spend less than $100,000 on Web Services projects within the next three years, and the next largest group (18%) anticipates spending less than $250,000 during the same time period. Spending is generally proportional to the size of an IT project, so one must assume that these groups envision starting with small Web Services initiatives. However, one of the promises of Web Services is the extension of existing IT assets as opposed to wholesale replacement of systems. In many instances, deploying Web Services will not force new investments in hardware or applications and will require only limited spending on XML and SOAP coding.

There is a correlation between the size of respondents' companies, as expressed in terms of revenue, and their spending plans for Web Services. In general, the larger the company, the more they plan to invest in Web Services. On the other hand, there appears to be little correlation between the level of planned spending and the strategic importance of Web Services expressed by respondents.

IT history teaches us that most projects come in over budget, often because not all the costs were identified and planned for up front. In the case of Web Services development and deployment, it is easy to overlook necessary initial expenditures such as developer training and third-party architectural services, as well as reoccurring costs like the organization and maintenance of Web Services components. Many firms will find that their original projections for Web Services spending were overly optimistic or simply ill informed.

Metrics

Every technology is judged on its ability to deliver benefits, either through cost reductions or revenue generation.

Respondents are evenly divided as to their financial goals for Web Services. Equally sized groups believe that Web Services will be a necessary cost of doing business that provide no measurable revenue upside, that Web Services will be revenue generators that make a positive contribution to the firm's profitability, and that Web Services will be a break even proposition.

The equality of the split in expectations underscores the lack of familiarity that many organizations have with service delivery. Without similar experiences to rely on, many companies have no way of knowing what to expect in terms of financial payback from Web Services. That two-thirds of respondents are anticipating Web Services will be cost negative or neutral at best matches well with the perception that proving ROI will be a significant obstacle to Web Services implementation.

Twice as many respondents (31%) are looking for Web Services to reduce the cost of integrating internal applications than for any other benefit. Current, proprietary EAI methods are very expensive because of the amount of time required to write and modify connectors that are specific to one pair of applications. Web Services allow an application to integrate with any other by passing XML-based business objects between them using a standard transfer protocol (SOAP). The cost of building a standards-based one-to-many data relationship is much lower than building multiple proprietary one-to-one relationships.

Improved integration of applications between organizations is also a benefit that Web Services users expect to reap. The potential cost savings and revenue generation from external integration far outweighs the benefits of internal application integration. However, it will be a couple of years

before most organizations have finished their internal Web Services projects, so not as many respondents are looking for externally-oriented benefits at this time.

It is reasonable to expect that reduced total cost of ownership (TCO) would have been the most cited IT benefit, if only because all of the benefits provided as choices contribute to TCO. That TCO was only the third most mentioned benefit suggests that most respondents are more focused on initial IT benefits of Web Services deployment than on long term gains.

The internal integration of business functions is the most attractive business benefit of Web Services for over a quarter of the respondents, paralleling their choice of the improved ability to integrate applications within an organization as the primary IT benefit. Web Services enable the sharing of more than just applications between corporate functions; they also let various departments share information and, potentially, business processes. Web Services promise to obliterate the silos that information systems have created over the last five decades and to unify organizations. However, it will take more than application integration to realign corporate value chains. Organizations implementing Web Services must evaluate and, potentially, redesign the business processes that cross-functional and departmental boundaries.

As with the external application integration IT benefits expected of Web Services, respondents' next highest business benefit goal is improved integration of external value chains. When information, application, and processes can be shared easily and quickly as Web Services with business partners, suppliers, and even customers, then organizations will operate more efficiently and profitably. Integration enables collaboration, and collaboration drives revenue and cuts costs. Again, inter-organizational process evaluation and design will be necessary to achieve complete and effective integration.

The expectation of these two benefits helps explain why some respondents viewed Web Services projects as a cost center and others said that they would contribute to corporate profit. The majority of those that named internal integration as the primary anticipated benefit also said that Web Services would be a cost center in their company. On the other hand, respondents that viewed external value chain integration as the key benefit expected Web Services to drive profitability and also sought the creation of new revenue streams through Web Services application. CIOs and IT managers tended to be in the first camp, whereas CEOs and line-of-business managers were most often in the latter.

As stated previously, businesses are looking to Web Services as a cost effective means of extending existing computing assets across the organization and to external constituents. Indeed, 43% of respondents indicated that this is how they will measure the potential impact of Web Services when they justify related projects.

Respondents are looking to leverage more than just internal infrastructure and software; they will argue the merits of Web Services based on utilization of internal development resources as well. The expectation that respondents will be able to use internal IT staff as their primary Web Services development resource was clearly articulated. The business case for adopting a Web Services architecture hinges on cost efficiencies resulting from improved use (and reuse) of existing internal IT resources, both human and machine.

Buyers' View of the Future

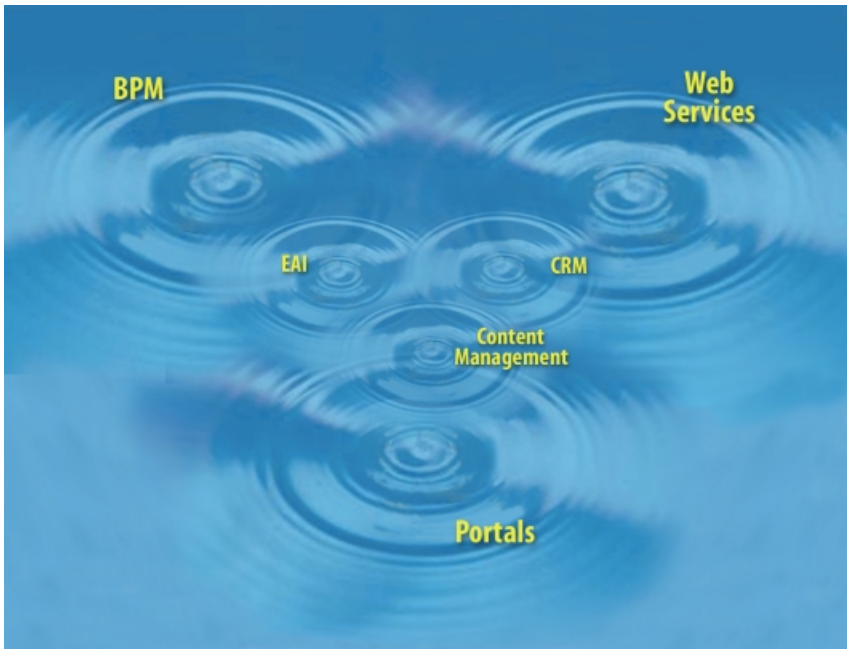
The majority of respondents (55%) believe that Web Services will become the foundation for most organizations' on-line activities. Only 12% said that Web Services would not become the pervasive operational model for Internet-based business. One third of respondents felt that it was too early to make a call either way.

These numbers suggest that there is considerable momentum behind the Web Services movement, but just under half of the market still needs to be persuaded through conceptual education and practical demonstration.

What's Missing?

A majority of respondents (52%) said that Business Process Management (BPM) standards are noticeably absent from first generation Web Services offerings. Many companies are looking to codify and share business processes as Web Services, both internally and with external value chain members. However, none of the widely supported Web Services standards (XML, SOAP, WSDL, and UDDI) address process management.

The Business Process Management Initiative (BPMI) has proposed the Business Process Markup Language (BPML) as a standard schema with which to describe business processes in XML. Similarly, IBM has put forth its Web Services Flow Language (WSFL). In time, it is likely that BPML, WSFL or another XML schema will become the accepted standard for management of Web Services in the context of a business process, so this shortcoming of current Web Services offerings should be viewed as a temporary void.



Distributed authentication methods were seen as lacking in Web Services by 46% of respondents. Most initial Web Services deployments will not be hampered by the current requirement for centralized authentication, as they will be internal projects that can leverage corporate profile directories to authenticate users and groups. When more companies begin sharing Web Services externally, distributed authentication will be critical. Without it, there is little security and no trust unless it has been established previously. Web Services or any other enabler of on-line collaboration cannot succeed with the security and trust that are provided by rigorous identity validation and authentication mechanisms.

Applicability of Web Services

Over the past year, Web Services has grown into its own technology sector. The value of Web Services, however, will be had through other applications and systems. Enterprises will use Web Services to enhance the functionality of existing technology infrastructure.

To truly understand where Web Services will impact the enterprise, it is necessary to recognize the convergence of diverse technologies into Web Services. An important evolution in the software world has led to the need for Web Services. A number of technology sectors are complementary to Web Services, and in turn complemented by Web Services functionality. What is striking is the fact that all of these sectors are touched by Web Services and made more whole by Web Services, yet none of them will be subsumed by Web Services.

The graphic to the left helps add perspective. Think of technology sectors as existing in and of themselves. At their beginnings, they are like a small tree, with just one ring showing their young age. As they evolve, they grow concentric rings as their base widens. In the

technology world, as those rings widen, they are bound to overlap. Thus, portal technology overlaps with BPM technology now, which makes perfect sense considering that users interface with their process tasks through a corporate portal.

Likewise, as Web Services evolve, this sector too, will complement many other technologies. The areas of great immediate impact will be:

- Portals: as mentioned previously, business users will access Web Services through their portal interface. In this way, portals will be the delivery mechanism of the future for Web Services and the destination for employees to execute their business tasks.
- BPM: because Web Services will be used in the context of a process, it will the job of BPM tools to orchestrate services and to automate their usage. Web Services will also be processes in and of themselves (as a process can be wrapped up as a service to be delivered to another process or application).
- EAI: Web Services could play the role of "lightweight" integration, helping to provide a layer of separation between applications and processes.
- Content Management: Web Services are important for content management applications both because the expense and difficulty involved in provisioning Web content for business applications today, and the ability of content management suppliers, most of whom already utilize XML in their applications, to develop the standards-oriented interfaces for Web Services operations. The challenge will be to get existing content that is not in XML form converted at a low cost and in a timely fashion.
- CRM: information about customers will play an increasingly large role in business processes, especially those that are customer facing. Integrating this information seamlessly into these processes will be critical—Web Services can be that method of integration.

Market Analysis and Definition

Many would describe the Web Services market as immature. Certainly, there are many vendors vying for mindshare and plenty of debate over the importance of standards and the state of interoperability. A look at the supplier landscape, however, does show that the market is maturing and focus is sharpening.

Web Services Supplier Landscape

Categorizing vendors is always a difficult task; it is even more difficult in the realm of Web Services because any vendor can lay claim to being a provider of Web Services. In fact, this situation has led to much of the confusion surrounding Web Services. Many software vendors simply SOAP-enable their applications in order to claim Web Services functionality. While it is true that SOAP-enabled applications are important, it is also true that they are not enough—more sophisticated Web Services tools are necessary to provide real value.

Platform Vendors

Vendors such as IBM, Microsoft, BEA, and Sun Microsystems are considered platform providers, supplying products and services across the realm of Web Services functionality. These vendors provide both infrastructure for building and deploying Web Services, as well as tools for utilizing Web Services within business operations.

Infrastructure Providers

In addition to the platform vendors, there are those that provide back-end technology, such as applications servers, to ensure performance and scalability. This back-end is infrastructure both for the creation of Web Services and the connection/integration to necessary systems within an enterprise. Fujitsu is a good example of a company that provides infrastructure in the form of its INTERSTAGE product.

Service Development & Design

While the application server provides a technological foundation for Web Service creation, the integrated development environment is where services are actually designed. Sun Microsystem's Forte and BEA's "Cajun" are examples of tools used as an IDE for the creation of Web Services. These development environments are typically used for the creation of simple services, whereas applications (such as portals) can be development environments for composite services (where several Web Services are combined to create a "meta" service).

Deployment Services

Deployment servies are offered by Web Services networks, as described earlier in the report. These vendors provide support for Web Services lifecycle management, complete with security services and delivery mechanisms.

Orchestration Services

Orchestration is where process and Web Services intersect—these vendors facilitate usage of Web Services by managing how services function within processes. Examples of orchestration vendors include Fuego, HNC Software, and Nobilis.

Application Development

There are several software vendors (among them portals), that produce applications that both consume Web Services and use simple services to create composite applications and services.

Collaborative Services

One of the driving forces behind Web Services acceptance is the need to better communicate with and work with partners. Vendors such as divine are offering collaborative services—secured messaging, team interaction, group feedback, telephony webinars—that allow businesses to work together in ways that were not possible five years ago. Deliver these services as Web Services allows them to be seamlessly integrated to processes and to have immediate impact on business execution.

Content Services

Content management vendors are focused on turning content processes into Web Services. By doing so, they are changing the game. Content management used to be focused on getting content to the Web browser so users could interact with it. With Web Services, these vendors are now simply driving content to the application that needs it. In this way, the content is serving the employee, customer, or partner that needs it.

Vendor Business Models

Mirroring the dot.com phenomenon in many ways, there are pure play Web Services tools that are taking the lead in niche functionality and peripheral movers that are developing Web Services capabilities on top of or within their existing products. As can be expected, these vendors are coexisting for the moment, though significant merger and acquisition activiey has already begun to take place.

How Platform Vendors Shape This Emerging Market

The term "platform" has different connotations depending on who is interpreting it and their context. In the software world, the word platform conjures up images of Java, .Net, Perl, Open Source, and MacOS. The major platforms (or frameworks) in the context of Web Services are J2EE and .NET. Vendors that are considered "platform providers" tend to offer products that offer a range of Web Services functionality for either or both of the two major platforms. Microsoft has several products that offer Web Services capabilities for the .NET platform, while IBM, BEA, and Sun Microsystems do the same for the J2EE platform. What distinguishes a platform vendor from other providers is that platform vendors provide both infrastructure products, in the form of application servers, as well as tools for orchestration and/or composite application development.

The platform vendors are essentially selling technologies to create and use Web Services, not Web Services themselves. The actual sale of Web

Services will not occur until enterprises create services that other companies may want to leverage. These platform vendors, though, shape how the market will evolve—they are driving many of the standards efforts and they provide much of the infrastructure that will be the foundation for Web Services going forward.

By no means is this a deathknell for non-platform vendors. In fact, there will be ample opportunities to piggyback on the efforts and initiatives of platform providers. The added-value services that niche vendors begin to provide down the road will shape the stage in the evolution of Web Services.

Closing Point

Why is there so much confusion surrounding Web Services—after all, a Web Service itself is a relatively simple thing? The reason is that this simple thing, the Web Service, could play multiple roles within enterprises. The possibilities are literally endless, and it is difficult to forecast beyond the near-term. For now, though, it is best to remember the complementary nature of Web Services—this technology will help many other technologies fulfill their promises in a more efficient manner.

Vendor Report: IBM

Overview

IBM's strategy for Web Services is at once both holistic and modular—items from across IBM's product line are available to take advantage of Web Services, yet customers can also buy only those products they need. An IBM "Web Services product" does not exist; rather, support for Web Services has been built into many of IBM's product families. This distinction is what makes IBM a true platform provider for Web Services.

IBM's strategy for Web Services is not just about products—it is about driving the industry as a whole forward. In addition to adding Web Services support to its family of products, IBM is helping to lead and support standards initiatives, such as the W3C, OASIS, and WS-I. The company is also using its alphaWorks site and professional services teams (jStart and IBM Global Services) to get code to developers quickly and to support open-source efforts.

The advantage to the customer is twofold. If the customer already has IBM products in its infrastructure, it can build Web Services capabilities into the existing systems. If a customer has an infrastructure that is non-IBM, it can still use IBM products to leverage its existing systems and take advantage of Web Services.

Through Web Services capabilities, IBM seeks to deliver to its customers the ability to conduct dynamic e-business. This means allowing customers to be flexible in responding to market needs while leveraging existing technology investments, a requirement of virtually every organization. IBM's Web Services support will help to break down the barriers of incompatible technologies and platforms by utilizing open standards, such as XML, SOAP, and UDDI.

Perhaps most importantly, IBM realizes that Web Services are delivered through a network of products and IBM is therefore not creating a "Web Services product." IBM's goal is to deliver Web Services functionality throughout its family of enterprise software products (WebSphere, DB, Lotus, Tivoli), in addition to driving Web Services evolution from lessons learned in IBM Global Services engagements.

A Platform Approach

There are four pillars of IBM's approach to Web Services: the IBM WebSphere, a J2EE-based Web application server for execution of Web Services; IBM DB2 for managing data and supporting stored procedures; Tivoli for management and security of Web Services environments; and Lotus products for collaboration (Domino, LearningSpace, Discovery Server, K-Station, Quickplace, and SameTime). While these IBM infrastructure software families have Web Services capabilities incorporated into them, it all begins with the execution environment for Web Services—this is why IBM has extended the IBM WebSphere brand to provide this execution environment for Web Services.

How WebSphere Delivers Web Services

The WebSphere Application Server is designed to be production-ready—companies can create solutions for dynamic e-business by deploying Web Services through WebSphere. By leveraging the SOAP, UDDI, and WSDL standards, WebSphere can deliver interoperability between Web Services and J2EE applications. Security issues are addressed via HTTPS support and implementations of XML Signature and Encryption.

Tightly integrated to the WebSphere Application Server is the WebSphere Studio Application Developer. It is a development environment for creating, building, testing, publishing, discovering, and integrating of Web Services-based applications that support UDDI, SOAP, WSDL, and XML. The WebSphere Studio Application Developer is built

on Eclipse open-source development platform, allowing customers to add third-party tools or to create their own tools. IBM's goal with the WebSphere Studio Application Developer is to provide a set of tools that speeds both the deployment of Web Services, as well as a company's ability to find and integrate a Web Service.

Eclipse

In an effort to address the lack of interoperability among tools that has plagued developers, came the creation of Eclipse, which is the foundation of IBM's newest WebSphere Studio products.

The Eclipse Platform is a functional development environment consisting of a framework that is extendable by plug-ins. It includes a fully functional IDE written in Java, meaning that Eclipse is more than just a collection of APIs. Eclipse has tools that plug into the platform are, providing users with a set of common services that make tool-to-tool integration possible. Eclipse has many extension points that a plug-in can interact with—the plug-ins can interact with each other or with the platform itself.

For IBM customers, the benefit provided by the Eclipse Platform is the ability to create custom IDEs depending on their specific requirements.

MQSeries

Also important to the Web Services equation is the IBM WebSphere MQSeries transaction messaging system and integration server, which leverage the IBM WebSphere product line. The MQSeries products allow enterprises to automatically transmit, receive, filter, and personalize digital information in real-time across platforms and across applications. The messaging system and integration server complement each other, with the messaging system providing a secure, cross-platform transport channel and the integration server linking business operations both internally and externally.

One of the areas where Web Services will have a large impact is in application integration. The MQSeries integration software products allow businesses to connect internal operations to business partners and/or customers in real-time. The products connect multiple applications, Web site, databases, or other information sources within a common framework. The result for IBM customers is an ability to extend their business through the Internet and optimize relationships with partners and customers. At the core of the integration server are integration brokers, which are supported by adapters, connectors, XML, transformation engines, rules engines, workflow, and business process engines. Together, the brokers and supporting mechanisms link operations internal to the enterprise, as well as with operations of partners or customers.

Portal Server

The WebSphere Portal Server builds upon the capabilities of the WebSphere Application Server, enabling enterprises to build next generation portals—personalized, secure, single points of interaction with people, content, applications and processes. The portal server gives customers the dual ability to publish WebSphere Portal Server portlets to a Web Services directory and also to search the directory and add a Web Service as a new portlet for use in the Portal Server.

This report described earlier the increasing overlap between Web Services and portal technologies. Delphi sees portals as a major delivery mechanism for Web Services. IBM addresses this evolution with the Portal Server. Customers will be able to use WebSphere Portal Server to set up distributed enterprise portals that will allow dispersed employees and partners to share portlets of all types using Web Services.

IBM Web Services Toolkit

For those customers who want to develop Web Service solutions that execute atop this IBM infrastructure software IBM provides the IBM Web Services Toolkit (WSTK). This software

development kit includes a run-time environment, a demo, and examples to aid in designing and executing Web Services applications that can automatically find one another and collaborate in business transactions without additional programming or human intervention. For example, there is a demo of how technology standards, such as SOAP, UDDI, and WSDL, work together.

The basic software components needed to create a Web Services environment are provided with WSTK. Included within the kit are an architectural blueprint, sample programs, utility services, and tools for developing and deploying Web Services. It also includes a Web Services client API that can be used to directly access a UDDI registry, thereby allowing firms to publish directly to public registries or to access external services that already exist in those registries. WSTK can be used with the IBM family of software products to begin creating Web Services solutions.

The IBM Web Services Gateway reduces development costs to make selected services available to different divisions within an enterprise or customer and partners either outside the firewall or who use different protocols. With the Web Services Gateway developers and IT Managers can safely "externalize" a Web Service so that it can be invoked by users from outside the firewall in a managed, controlled and secure environment.

Delivering Added Value

IBM's strategy for Web Services is to provide more than just the infrastructure that WebSphere brings to the game—providing management services and tools is an important part of the equation, as well.

Tivoli

The IBM WebSphere Application Server program provides a solid base for Web Services solution creation. IBM's commitment to Web Services does not end with the IBM WebSphere family, as several IBM products come into play in delivering Web Services functionality. The market overview section of the paper illuminates the need for security and availability in an environment that uses Web Services. Tivoli has products available today to help ensure that Web Services have a high level of security, availability, and performance management.

Web Services standards help to simplify development and integration of management software. Tivoli is a product that interprets and acts upon management data from all sources—including Web Services applications. It is possible for IBM to help service providers gain value in the Web Services environment. With that in place, they must ensure that their Web Services can be delivered via existing standards, WSDL for describing services, UDDI for publishing and finding services, and SOAP for communication between the service provider and the service requestor.

Services providers will then have to deliver on Service Level Agreements that meet the needs of their customers. Doing so will mean that service providers have the infrastructure management products that go beyond the standards of today and deliver an environment for true interoperability. This is where Tivoli comes in—providing the security, availability, and performance products that will allow Service Providers to compete in a world of Web Services.

Tivoli manages the IBM infrastructure software that runs Web Services. The core set of Tivoli availability and performance management products, in conjunction with Tivoli Managers for the WebSphere Application Server, Lotus, and DB2, give customers management of the applications that execute their Web Services. The integration between the Tivoli SecureWay products and IBM middleware products e.g. WebSphere, also provides policy-based security for the Web Services environment.

Tivoli Web Services Manager gives service providers and requesters the ability to monitor Web Services and to ensure that services are performing within set thresholds. Customers can

monitor transactions in real-time or via synthetic transactions.

Tivoli is adapting many of its existing products or building new products to take advantage of the emerging standards that promise to improve interoperability and integration capabilities. Also, Tivoli is aiming to allow groups of customers to provide management across organizational boundaries by allowing service providers to utilize standards to publish vital management information about their Web Services.

There are also emerging standards for managing and securing of Web Services—Tivoli is working to help define those, as well. Tivoli provides the non-repudiation functionality that is a part of SOAP by leveraging the standards on XML Digital Signatures and XML Encryption. Tivoli has also published a reference architecture, Tivoli Management Extensions for Java, for management of emerging Web Services applications built in Java.

DB2

DB2 is a relational database that supports the open Internet standards, such as XML, WSDL, and SOAP, required by Web Services infrastructure. DB2, through the XML Extender, allows Web Services applications to access data stored in DB2 as XML-structured documents. The advantage to business being more efficiency in developing new dynamic e-business applications using XML interfaces only (this isn't a sentence -- doesn't have a verb). With DB2 using the Web Services infrastructure, IBM aims to help customers integrate their current business processes by sharing information with their partners and improving efficiencies and reducing costs.

Web Services access to procedures and data stored in DB2 is provided through tools and runtime software. There are three types of Web Service operations that are supported. The first is XML-based—retrieving and storing XML. The second is SQL-based—query and update. The third operation is based around stored procedures invocations.

The DB2 XML Extender provides the support for the XML-based operations by allowing XML documents to be stored or retrieved from a DB2 database. This means that stored procedures can be exposed as Web Services, and they can also serve as an integrating technology because they can call out to other Web Services.

Lotus

As part of the IBM family of products, Lotus will play a large role in IBM's Web Services capabilities. With Web Services, the applications running on Lotus platforms today can be integrated with other applications, both within an organization and beyond. Through Web Services Lotus can help to enable person-to-person interaction in context with enterprise information discovery and e-business transactions.

A Tool For Lotus Products

Lotus provides a Web Services Enablement Kit, which will include instructions for using the IBM Web Services tools to configure custom applications running on Lotus servers to be Web Services consumers and/or providers. The Enablement Kit is not a product in and of itself, but rather a for enhancing Lotus software with Web Services. The larger Web Services strategy for Lotus is to add on Web Services modules to the collaboration, knowledge management, and e-learning products.

The value proposition is that applications, such as Domino, can be utilized by customers as Web Services. Domino, as a workflow tool, can orchestrate Web Services created in the Websphere environment or a trading partner or supplier can invoke a complete Domino process as a service. Again, the benefit will be the flexibility to do business in the most efficient manner possible.

Several Lotus products play a role today with respect to Web Services. :

- Lotus Domino Application Server—Domino collaboration, workflow, and messaging capabilities are accessible as Web Services by adding SOAP interfaces and WSDL descriptions to new or existing Domino applications, using XML, with the native XML capabilities in Domino.

- Lotus Domino Workflow—works on top of Domino, providing the ability to develop, manage, and monitor enterprise-scale, human-interactive business processes. By adding SOAP interfaces and WSDL descriptions to existing and new workflows, developers can create Web Services that allow external applications to use Domino Workflow-based applications over the Web.

- Lotus Knowledge Discovery System—developers can add SOAP/WSDL interfaces over the expertise database of the Knowledge Discovery System, enabling the information to be served up as a Web Service. The Knowledge Discovery System has two components. First is the Lotus K-station, a knowledge portal with collaborative capabilities that uses a browser-like user interface to access information sources. Second is the Lotus Discovery Server, a knowledge server enabling search and expertise location.

- Lotus Sametime—a real-time collaboration solution combining chat, whiteboarding, audio, video, and application-sharing capabilities that enables immediate between people anywhere. Sametime Everyplace extends Sametime's real-time capabilities to mobile devices. Using Java APIs and Beans, developers can build Web Services that incorporate online presence awareness and instant messaging as part of any Web Services-enabled system.

- Lotus LearningSpace—a distance learning platform enabling integration of live, asynchronous, and self-paced course content delivery. Using Java APIs and Beans, developers can build Web Services that incorporate LearningSpace capabilities, such as course lists or schedules, as part of any Web Services-enabled system.

- The Lotus Web Services Enablement kit is designed to further demonstrate the development of Lotus-based Web Service applications. The kit includes instructions for using the IBM Web Services Toolkit to configure custom applications running on Lotus servers to be Web Service providers and/or consumers; it also includes sample Web Services applications.

Partnering Program—Web Services on WebSphere (WoW)

Through WoW, IBM is developing a partner community that will focus on building Web Services solutions with the goal of enabling customer development and deployment of dynamic Web Services solutions. At the core of this initiative is IBM's application server, WebSphere and its three partner constituencies. The first partner constituency is the application software vendors. IBM will provide the technical, business, and marketing ability to help these vendors rapidly adopt Web Services on WebSphere then quickly announce and deliver WoW-enabled products and services.

The second partner constituency is systems integrators. Web Services content, certification, and support have been added to current programs in order to give systems integrators the ability to add Web Services to their existing WebSphere practices.

The third constituency is complementary technology software vendors. The WoW initiative will provide complementary technology software vendors with the products and skills they need to

develop the highest quality support for WebSphere. In this way, customers can (strikethrough: will be able to) leverage investments they may have already made in complementary technologies.

Additionally, there is a WoW Advisory Council that guides and directs development efforts. The advisory council consists of a small number of key members of the three partner constituencies. The functions of the advisory council are multifold. The council represents the broader community's technical and marketing needs and acts as the primary feedback conduit to IBM. It also provides business enablement guidance to the broader community in an effort to smooth the transition to Web Services business models.

The Advisory Council also provides thought leadership for the WoW community, driving constant innovation. Finally, the council develops a set of best practices for use by the greater WoW community to model, design, build, develop, deploy, change and manage Web Services solutions.

IBM and Grid Computing

IBM is one of the first companies to begin acting on the synergy that exists between Web Services and grid computing. The company realizes the effect that grid computing could have on eUtility—the utility model for delivery of software will allow businesses to use simple front-ends to access the software they need, and pay for it on usage or contract basis rather than buying it outright. This model is also expected to provide different levels of quality of service (QoS) with corresponding fee structure in their service-level agreements (SLAs).

IBM and the Globus Project (a multi-institutional research and development effort creating fundamental technologies for computational grids) have developed a set of new specifications that allow businesses to share both applications and computing resources over the Internet, making grid computing applicable to real business situations. The specifications are referred to as the Open Grid Services Architecture (OGSA)—a set standards that combine the benefits of grid computing and Web Services.

Using the OGSA, customers can access and share computing resources on demand over the Internet, relying on an infrastructure that is resilient, self managing and always available. The goal of this initiative is to allow customers to integrate applications and share data and processing power while capitalizing on potential cost and efficiency savings.

This set of specifications builds on standards such as XML,WSDL, and SOAP for grid computing, which are used to locate, schedule and secure computing resources. IBM intends to leverage OGSA as a key foundation in its Project eLiza initiative. Project eLiza is IBM's Autonomic Computing initiative to build an open, heterogeneous, self-managing server infrastructure for e-business and commercial grid implementations.

IBM And Standards Development

For Web Services to have a real impact on businesses, standards must be developed and adopted. IBM has taken a leadership role in developing the specifications needed to create Web Services standards. The company created WSFL and HTTPR, and has worked with other vendors to make SOAP vendor-neutral, to create WSDL and UDDI.org, and to propose changes to HTTP that would make it a reliable protocol as an underpinning to SOAP messaging.

IBM Professional Services Teams

IBM's Web Services capabilities as represented through its family of products and solutions are driven, in many cases, by advances in the professional services groups. At IBM, two groups are capitalizing on Web Services. IBM Global Services (IGS) is working with numerous clients on

heterogeneous systems and seeing the need for cross-platform computing. IBM jStart is a group that is focused on helping customers and partners embrace new and emerging software technologies. The jStart team helps customers design and deploy Web Services applications for both EAI and B2B purposes, focusing on pilot projects.

Case Study: Storebrand

Elimination of data entry errors results in more efficient pension plan management for both customers and payroll vendor partners.

Storebrand is Norway's largest provider of pension plan, insurance, and other financial services—more than 390,000 employees of 6,500 companies in Norway alone are members of Storebrand pension plans. With that many individuals involved in the pension program, keeping information up-to-date becomes a major challenge that affects everything from process performance to customer service.

Already, Storebrand kept the data for all its plan members in an IBM DB2 Universal Database on an IBM S/390 Parallel Enterprise Server. The process for changing a member's information within the database was cumbersome. The employer would have to manually revise its records followed by sending updates to Storebrand through FTP (file transfer protocol), snail mail, or fax. Storebrand customer service representatives were then tasked with manually entering the changes into the DB2 database.

This process left ample room for human error, both by employers that are manually changing information and by Storebrand CSRs that are manually reentering information into databases. In addition, two sets of manual entry constitute a sign of redundancy within the process. Storebrand set out to solve this problem by automation the capture of this data, with the goal of reducing costs and errors, improving customer service, and enhancing the company's leading-edge image.

Storebrand decided that the best way to automate the capturing and updating of customer information changes was to create a Web Service that its clients' payroll applications could access over the Internet. Storebrand was already using WebSphere software in other lines of business; therefore, it made sense to choose IBM to build the infrastructure for creating and deploying Web Services.

IBM offered Storebrand the right mix of products and professional services necessary to build a Web Services that could accomplish Storebrand's goals. The IBM jStart program helped Storebrand to leverage its WebSphere application server, and then to use Lotus Domino, IBM MQSeries, and MQSeries Integrator to create a secure channel for its applications to interact with those of its partners. Using this channel, the pension plan updates can travel directly over the Web through the WebSphere application server and into the DB2 pension plan database.

There are three main components to this project: a Common Object Model (COM) object that runs within the employer's payroll application; the Java-based business logic residing in the application server (WebSphere); and an integration layer based on MQSeries that transforms incoming data to make it DB2-ready.

IBM's Web Services Tool Kit (WSTK) was critical to the project. Storebrand used IBM VisualAge for Java to create the business object for the payroll application. Still, though, it would have to be wrapped in WSDL for it to be accessed as a Web Service. WSTK gave Storebrand the ability to quickly create the WSDL wrapper and to publish and deploy the Web Service without having to do any reprogramming.

About This Document:

The product-specific information contained in this document is intended to provide an overview of a specific product and vendor at the date of publishing. Facts presented have been verified to the best of our ability with the vendor and actual users of the product where indicated, however, Delphi cannot insure the accuracy of this information since products, vendors, and market conditions change rapidly. Delphi Group makes no implied or explicit warranties, endorsements, or recommendations in this report nor should such warranties be inferred from its contents. A complete assessment of your specific application, the method of implementation for a given product or technology, and the current state of that product must be considered in order for a recommendation to be made on any product's suitability for your purpose, needs and requirements.

Delphi Group is a leading provider of business and technology advisory services to Global 2000 organizations. With offices established around the world, Delphi has assisted professionals across disciplines and industries at nearly every major national and global organization and branch of government. Its clients and subscribers include more than half of the Global 2000.

Delphi Group
Ten Post Office Square
Boston, MA 02109-4603
v (617) 247.1511
f (617) 247.4957

Now, when employers make changes to the information in their payroll applications, the changes can be automatically reconciled with Storebrand's records. How does this happen? The payroll application extracts the data and translates it to XML. Storebrand's COM object then generates a SOAP request that access the Web Service through the Internet. Back at Storebrand, the Lotus Domino system authenticates the employer sending the information, then sends the XML data to a business object in the WebSphere application server, which routes it to a queue in MQSeries, thereby triggering a message flow in MQSeries Integrator. The Integrator then passes the information to the server.

The benefits of this Web Service are multifold for Storebrand. Very quickly, there have been savings in customer support costs through the elimination of data entry errors. Both customers and payroll vendor partners are pleased with this technology advancement—customers have an easier time managing their Storebrand pension plans and payroll partners can enhance their products with this Web Service.

This project also positions Storebrand well to take advantage of Web Services technology in the longer-term. It is possible for Storebrand to extent the architecture it created for the pension plan application to serve its other lines of business. This is the flexibility and ability to respond quickly that enterprises have been chasing. Furthermore, Storebrand can publish its Web Services for others to access by utilizing WebSphere's ability to integrate with UDDI registries. This ability is but one of the several revenue-generating opportunities that exist due to this project.

Summary

The benefits derived from Web Services solutions can drive businesses to new levels of efficiency. The combination of IBM's products and services in the Web Services realm make it possible to begin realizing this efficiency now, without the need to learn new programming languages. IBM's offerings allow enterprises to leverage their existing software systems and to integrate those systems together, both behind the firewall and with partners.

IBM is pushing for the universal adoption of standards in an effort to bring true interoperability and platform independence to the market. That leadership, the experience of professional services teams such as IGS and jStart, and a platform offering consisting of Web Services infrastructure and tools, positions IBM as a leader in the Web Services marketplace.