Web Services - Right Here, Right Now

Delivering Web Services Today with IBM Solutions

By Lawrence Wilkes
Principal Analyst, CBDi Forum

1. Introduction

This report is aimed at IT management and senior professionals responsible for the design and architecture of business systems, from both a technical implementation and a business process perspective.

I should first qualify the term Web services as it is already used commonly to describe a number of diverse activities. Simply put, the Web services discussed in this report are the set of standard protocols and associated technologies that enable software applications to communicate with each other across the Internet.

From a technical perspective Web services are a small step, but one which I believe will have a profound impact on the way business and individuals interact with one another via the Internet. This is not to say that delivering Web services requires little effort. Rather, the effort is in the proper design of the business processes and the service architecture than in the programming of the Web services themselves.

The Need For The Next Generation e-Business

Web services are at the core of what is being termed the next generation e-business. The initial euphoria of Internet usage is now being replaced by more mature expectations of a quality of service that meets or exceeds that available from traditional channels. Users will no longer excuse poor service just because of the novelty factor. The next generation e-business must overcome the following challenges:

- **Consumer dissatisfaction** – e-commerce doesn't always work – often because of the following points – and customers become doubly dissatisfied when the retailer seems unable to resolve the problem;
- **Employee frustration** – They cannot access information needed to perform duties and respond to customers;
- **Reliance on self-service** – Access via browsers requires human operators and decision makers that introduce errors, delays and inefficiency;
- **Supply chain integrity** – There is increasing dependence on the supply chain as inventory is kept to a minimum, whilst at the same time there is a lack of visibility and integrity across it;
- **Un-integrated outsourcing and partnering** – Increasing reliance on external organizations but without seamless information flows.

A key reason for using Web services (rather than, say, HTML) is to enable applications to make intelligent use of information accessed across the Internet, rather than simply presenting it. The solution to the above is to improve the responsiveness of processes by automating the exchange of information between systems. To date, the inability to easily connect the diverse technologies in use has prevented this. The wide scale adoption of Web services will remove this major barrier.

Web Services Are Here Now

Though it will take time for the complete infrastructure to mature the basic technologies are already in place. Businesses are already using them to gain competitive advantage by improving the accuracy and timeliness of information that flows not only within their systems, but also between them, their customers, partners and employees to overcome the objections highlighted above. As such, those involved in the delivery of systems need to understand what capabilities they can use today, and the typical scenarios in which they can be applied.

This report looks at how Web services can be delivered today using IBM® solutions. IBM has been at the forefront of Web services: Having jointly proposed the first set of standards they have also led the way in delivering a broad set of products that turn the Web services vision into reality. IBM has demonstrated via it's support of Web services and Java™ that it is committed to open standards and the development of technologies and solutions that are platform-independent, enable interoperability, and importantly for their existing customer base, do not require any 'rip and replace' of existing systems.

WebSphere® Web Application Server 4.0	Provides full Web services support and Java™ 2 Enterprise Edition (J2EE™) certification, together with comprehensive integration capabilities required to expose existing applications as Web services.
WebSphere® Business Integrator	Enables businesses to use the reliability and transactional characteristics of MQSeries® to deliver messages from one Web services application to another.
Tivoli® Web Services Manager, and Web Services Analyzer	Monitors and analyzes the availability and performance of the infrastructure delivering Web services.
Lotus® Web Services Enablement Kit	Guidelines including demonstrations, samples, and source code, for enhancing Lotus products with Web services.
WebSphere® Studio Application Developer	Integrated development environment for building, testing, and deploying Web services and J2EE applications.
Web Services Development Environment (WSDE)	Enables software developers to create, test and deploy Web services applications and connect them to existing business processes. These tools also enable businesses to "publish" Web services applications for existing and potential trading partners
Web Services ToolKit (WSTK)	A package of design and run-time Web services tools, sample applications, and previews of the latest Web services technologies.
IBM jStart Program	A worldwide team of experts focused on helping customers and partners embrace new and emerging software technologies. The jStart team is assisting early adopters of Web services as they integrate this latest technology into their business models.

Table 1 –IBM Solutions for Web Services

Who’s Who?

It is useful to start by understanding who participates in Web services. The basic model for Web services is termed Web services Architecture[1] by IBM and describes the following three core roles in Web services:

- **Service Provider** - Provides a Web service that delivers a specific service;
- **Service Requestor** - Finds and invokes the Web Service in support of a particular solution. Also frequently referred to as the Consumer;
- **Service Broker** – Provides a directory or clearing house for Web services.

These roles can be viewed from either a run-time physical architecture or an organizational perspective. These roles may all be played by one or more organizations. From an organizational perspective it is also useful to consider the following roles:

- **Service Owner –** The entity that owns the logical service and with whom business contracts for its supply are made. The organization that acts as Service Provider at run-time, may not own the service from a logical business perspective. The Service Provider may be acting as host on behalf of the Service Owner;
- **Service Manager** – Manages and guarantees the integrity of the connection between Service Provider and Requestor. One or more organizations could provide a number of infrastructural services including reliable transport, authentication, identity and security, and billing.

Whilst there may be universal access to published Web services, it is obvious that they do not have universal applicability. Many will clearly only be specific to the industry or market they service. Often the Web services and the roles outlined above will take place within the boundaries of some closed ecosystem.

Some example ecosystems are:

- **A vertical or horizontal industry** - Such as Healthcare, Insurance, logistics, human resources, accounting etc. Particularly those that either already have wide scale information exchange between organizations, or those that have a predominantly information-based product;

[1] Web services Architecture Overview
http://www-106.ibm.com/developerworks/webservices/library/w-ovr/

- **A group of pre-qualified trading partners or marketplace** - To minimize security and trust concerns, a closed environment of trusted partners. Though use of the Web services remains dynamic, their participation in the ecosystem is pre-qualified;
- **Dominant organization** - It is possible that a dominant organization could insist on the use of Web services by its trading partner and dictate standards in a way similar to that in which Electronic Data Interchange (EDI) was sometimes introduced;
- **Enterprise** - Web services will be widely provided and consumed on an internal-only basis, taking place behind the security of the firewall.

Why Web Services?

There are numerous benefits to Web services that are presented throughout this report. However, sometimes these benefits can also be achieved through the use of other approaches and existing technologies. Some reasons why Web services are preferable to existing approaches are:

- **Loose technology coupling** – Remove the technology dependency between provider and requestor platforms;
- **Late binding** – Enable links between systems to be resolved at run-time;
- **Dynamic Inspection** – The availability and functionality of Web services can be discovered at run-time, rather than design time;
- **Programmable** – Enables systems to call functions across the Internet, not just pass or present data;
- **Use of standard protocols** - In place of multiple proprietary technologies;
- **Industry consensus** – There is genuine industry-wide agreement on the application of Web services. Previous interoperability approaches have often lacked certain key players.

Above all, Web services do not require the replacement of existing applications. Web services are a set of new protocols, not a new platform that requires systems to be re-implemented. On the contrary, Web services are the perfect way to enable existing systems to interoperate and evolve alongside new applications.

2. Vision – The Evolution of Web Services

Realizing the full vision of Web services requires a comprehensive set of standard protocols. IBM has been a major contributor to these standards and the following table lists some of Web services protocols that have been proposed by them and their partners, together with other related standards that will enable the delivery of robust and secure Web services.

Web Services Protocols	
SOAP	Simple Object Access Protocol. XML protocol to invoke Web services.
SOAP-DSIG	SOAP Digital Signatures. Proposal for securely signing SOAP messages.
UDDI	Universal Description, Discovery, Integration. Provides the basis for a registry of Web services.
WSCM	Web Services Component Model. OASIS (Organization for the Advancement of Structured Information Standards) initiative for composition and presentation of Web services
WSDL	Web Services Description Language. XML protocol to describe Web services.
WSFL	Web Services Flow Language. Proposal for workflow standard.
WSIL	Web Services Inspection Language. Enables the inspection of a site for available Web services.
WSXP	Web Services eXperience Language. Submission to WSCM.
Related Standards	
ebXML	United Nations and OASIS proposal for XML-based infrastructure for electronic business.
HTTPR	Proposal for reliable transport over HTTP.
SAML	Security Assertion Markup Language. OASIS initiative.
XML	eXtensible Markup Language. Describes structured data.
XML-Signature	Representing and verifying digital signatures.
XML-Encryption	Encrypting digital content.

Table 2 - Web Services and Related Standards Initiatives

Thankfully, those involved in proposing and deciding Web services standards, and providing implementations of them, are not attempting to cover the whole vision real in one go. The modular approach taken means each protocol can be introduced and evolve separately. As a result you do not have to wait for the whole vision to be implemented before starting out. Figure 1 illustrates how the vision will evolve in stages.

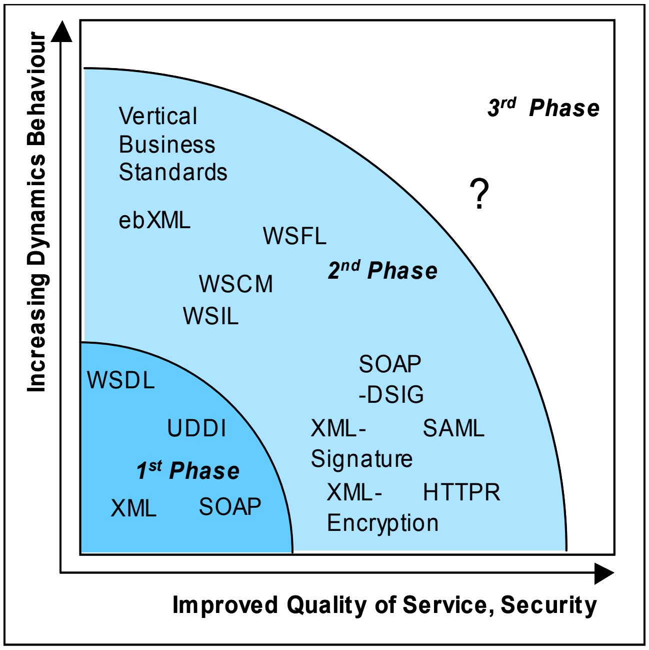

Figure 1 - Evolving Standard for Web services

Phase One (2001) – Dynamic Business Integration, Simple Transactions

We are already in the first phase. Though still going through the standards process, the core protocols shown in figure 1 are now in place, and have effectively become de facto standards. For many users, Web services will not require a significant investment in new technology as the protocols layer on top of existing technologies and access existing assets.

Early Scenarios

The technology available in phase one will enable organizations to enter into the following scenarios:

- Improve Enterprise Application Integration (EAI) – Web services are provided and requested within an organization. Efforts to connect diverse internal applications will be simplified via Web services. Availability of Web services on the source applications will remove some of the need for intermediate EAI solutions;
- Achieve Wide Area Business to Business (B2B) Integration between trusted partners - Web services will be used to provide real-time communications between applications of pre-qualified organizations, regardless of the technology of their underlying implementation;
- Deliver simple information services and transactions as Web services to wide audiences;
- Build Intelligent Endpoints - Requestors of Web services will use the structured information provided by Web services to deliver applications that take actions based on content, rather than just presenting it. This will replace the current reliance on user interaction through self-service to improve business processes automation.

The application of Web services is covered in greater depth later in this report.

Universal Support

Even though Web services are in their infancy, it is already evident that support for them is near universal. All major platforms already support Web services, major package vendors are providing Web Service interfaces and various connector/adaptor architectures are in place, which enable the majority of existing resources to be wrapped as Web services. It is difficult to find a previous technology that the whole industry has been so quick to endorse and adopt without major dissent.

Key Benefits Achievable Now

As with any new technology the question is always when should you start? Should you be an early adopter or wait until they are mature? Though Web services are in their infancy, many of the claimed benefits of Web services to both business and IT can be achieved using the technology available in phase one. For example:

- **Visibility** – Improve the visibility and access to remote information within or external to the organization;
- **Integrity** – Improve the integrity of data and processes;
- **Immediacy** - Enable real-time behavior when accessing and updating information, rather than replicating data on a periodic basis;
- **Automation** - Compared to providing self service access via a browser, the Web service provides programmable access to information, enabling it to be intelligently handled in the client;
- **Efficiency** - Remove manual overheads and delays with better business process automation;
- **Improve Developer productivity** - Spend less time on custom integration. Simplify access new and existing resources.

Phase Two (2002-4) Improved Infrastructure, More Complex Transactions

The next stage will introduce further protocols that deliver firstly a more robust infrastructure for Web services, and secondly will allow more complex business scenarios to be represented in a standard format. An important part of this is a variety of security standards and technologies that in combination should provide a secure environment for Web services.

Business Semantics Standards

Though Web services standards provide a more dynamic infrastructure for business to interoperate, this is of little use if the organizations do not share common meaning to the information they are trying to exchange. If the full vision of Web services is to be realized beyond tight knit communities then business standards must be agreed at an industry-specific level. Though not a pre-requisite, shared business vocabularies and processes will simplify the use of Web services. There should be deliverables from the ebXML project during this stage, and other vertically focused initiatives. Existing, or adapted standards will often be used, for example the ACORD[2] standards in the insurance industry are already being used as a basis for Web services by some organizations.

Phase Three (2004+) Adaptive Systems

Once comfortable with Web services, organizations will start to change not only their systems but also their business models to take advantage of the more dynamic behavior on offer. With a full range of standards in place, and a robust infrastructure, they will be able to take advantage of a more dynamic way to find business partners, use remote resources, and adapt to change. With standardized services, they will have the flexibility and freedom to change service providers, and implement self-discovering, self-healing systems that can constantly optimizing themselves to take advantage of new options.

Such is the rate of change, it is difficult to guess what additional standards or technologies might be proposed in the third stage. However, with businesses and individuals becoming increasing reliant on external Web services, it will place an added responsibility on IT vendors to ensure the reliability of the infrastructure that underpins them.

3. Implementing Web Services Today

If now is the time, then where is the place to start? As mentioned earlier, Web services are relatively small step in terms of technology. That said, in the coming year most organizations will find benefit from installing the latest versions of web servers that provide integrated support for Web services and hence a more efficient, productive and robust environment.

Where Do Web Services Come From? Implementation Approaches

One of the key concepts of Web services is that they separate the interface – the delivery of the service – from the actual implementation. Clearly any new requirements that are driven by the need to deliver Web services can be met by the development of new software. However, the expectation is that much of the implementation required to deliver a Web service will be found in existing portfolio of applications and databases.

The most likely scenario of course is that both new and existing software will be mixed together to deliver the Web service. Business and data access rules contained in existing applications will be combined with new software that are specific to the delivery of the Web service itself. In many cases there will be little option for the developer as the same back-end source must be used both for Web services and existing applications in order to ensure consistency of data presented though either.

The following outlines four scenarios for delivering Web services that demonstrate how the underlying architecture can evolve to support more complex requirements. IBM WebSphere® Application Server 4.0 Advanced Edition provides an ideal platform for the evolution of Web

[2] http://www.acord.com/

services, as it enables the delivery of increasingly sophisticated yet flexible scenarios without having to rebuild or re-engineer the infrastructure as needs change and usage grows.

1. From Website To Web Service

The websites of many organizations are totally separated from their core operational systems. It is used to publish information, and provide some straightforward self-service capability such as online ordering. Any information that must be passed between the two will be done via replication, file transfer, or simply manual re-input.

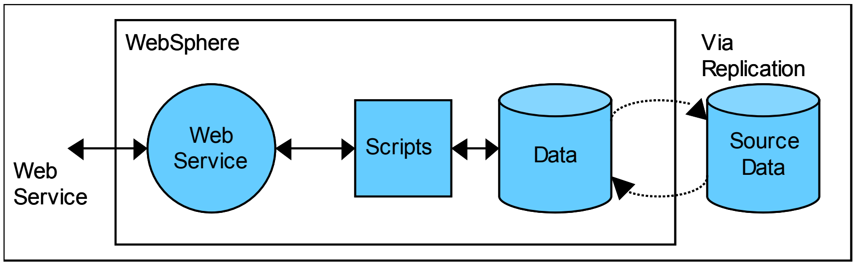

Figure 2 - From Website to Web Service

Exposing the information or the self-service function as a Web Service is theoretically a small task, as the existing capability is itself uncomplicated. However, the existing scripts are unlikely to have well formed interfaces, and new scripts may well have to be written due to the different behavior required by the Web service.

Website to Web service	
Objectives	Expose Web services off existing web site capability. Automation of self-service. Deliver information and service to multiple devices. Pilot and evaluation.
Development Effort	Delivery of Web service largely automated by tools. Likely to require new scripts to change behavior from presentation to service.
Platform	WebSphere Application Server Version 4.0, Advanced Single Server Edition.
New service Implementations	Primarily scripts on web server using variety of scripting languages, such as Java™ Server Pages (JSP), and XML.
Connection to other resources	No real time connections – *though other Web services might be consumed.*
Organization	Department, Small Enterprise, dot.com
Application	Simple information services. Very simple transactions.
Benefits of approach	Complexities of Web Service protocols, XML, etc hidden. Provides a managed environment on which to provision Web services off existing systems. Foundation for further expansion, scalability, etc.
Inhibitors	Service is not Integrated with core operational system. Likely duplication of scripts that present information via browser or via Web service.

2. Existing Systems and Platforms via Lightweight Web Server

Often it is necessary to deliver Web services that expose core operational systems. The obvious challenge is that these applications do not reside on the same Web application server.

This requirement can be met through use of a Web server whose prime role is to manage the delivery of the Web service itself and also provide a platform to connect them directly to the relevant existing system.

The suitability of existing systems to support Web services is of course varied. This is examined in more detail later.

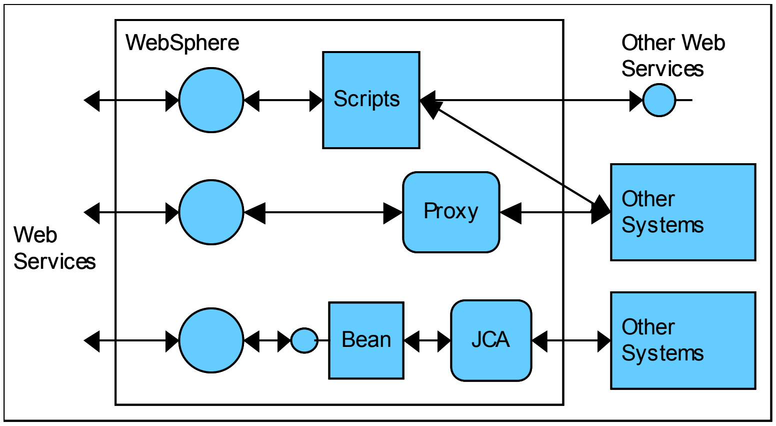

Figure 3 - Existing Systems as Web Services

Existing Systems and Platforms via Lightweight Web Server	
Objectives	Build new simple implementation. Wrap and expose existing assets directly as Web services. Pilot and evaluation.
Development Effort	Minimal. Delivery of Web Service largely automated by tools. No need to invest heavily in Object Oriented (OO) approaches, or Java™ 2 Enterprise Edition™ (J2EE™).
Platform	WebSphere Application Server Version 4.0, Advanced Single Server Edition.
New service Implementations	Extensions to existing systems, and use of existing platform. Applications on web server using JSP pages, Java™, servlets.
Connection to other resources	Scripts to existing applications via number of proprietary interfaces and technologies. Via Pluggable Service Providers and other Proxies. E.g. to DB2 via Stored Procedures and XML Extender.
Organization	Department, medium enterprise.
Application	Internal EAI, provide simple information services, simple B2B and supply chain automation.
Benefits of approach	Complexities of Web Service protocols, XML, etc., hidden. Provides a managed environment on which to provision Web services off existing systems. Platform provides core Web services and J2EE support with simplified administration. Foundation for further expansion, scalability, etc.
Inhibitors	Web services only reflect existing interfaces. Caution regarding proliferation of non standard integration technologies. Suitability of existing system to match Web service. Accessibility of existing system.

3. Full Object-Based Approach

Sophisticated and large-scale Web services scenarios will clearly require more complex implementations. Even where many of the business rules and data used reside in existing back-end systems, exposing them as Web service may require significant new functionality (see 'Some Key New Components' later in the report).

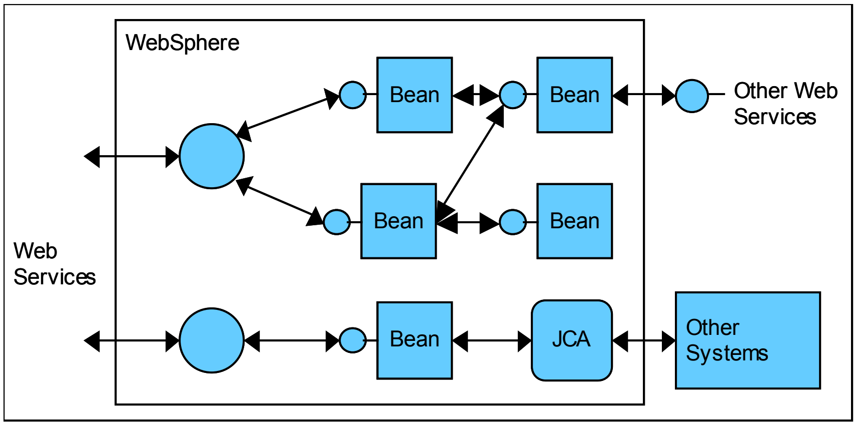

Figure 4 - Full Object Implementation

Organizations that use J2EE to deliver new applications will be will find the additional delivery of Web services straightforward from a technical perspective. Similarly in extending existing J2EE applications. The greater effort will be in ensuring the design of the system is suitable for the loosely coupled Web services approach in comparison to more tightly coupled component architectures.

Full Object-Based Approach	
Objectives	Deliver significant new functionality. New components to implement Web Service. Need to aggregate services or change behavior. Web services do not directly reflect existing services that were primarily designed for internal use or with specific endpoints in mind.
Development Effort	Delivery of actual Web services still automated.
Platform	WebSphere Application Server Version 4.0, Advanced Edition. Full use of J2EE.
New service Implementations	J2EE, Use of commercial components.
Connection to other resources	J2EE Connector Architecture (JCA), other Web services.
Organizations	Medium to Large enterprise. Committed to Java™ Platform.
Application	Complex transactions, aggregation of multiple resources.
Benefits of approach	Use of Web services transparent to Java developers. Capitalizes on existing Java skills. Flexibility via components, J2EE benefits.
Inhibitors	OO and J2EE skills required.

4. Business Process Automation (BPA) Approach

BPA tools are an excellent way to organize both the provision and consumption of Web services. If the premise of Web services is to deliver a more dynamic mechanism to assemble applications, then what better way to do that than via a BPA tool that is itself geared towards delivering more dynamic business process behavior?

Large enterprise will also have the widest diversity of existing assets. IBM WebSphere Application Server Version 4.0, Enterprise Edition includes Common Object Request Broker Architecture (CORBA) and IBM MQSeries support enabling investments in these technologies to be more easily used in support of Web services.

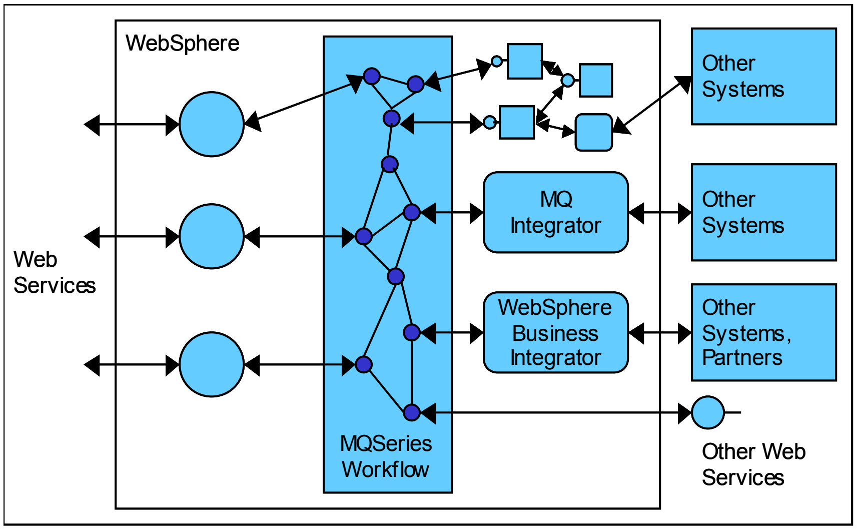

Figure 5 - Business Process Automation Approach

Whilst the BPA may appear to add another layer of complexity, it provides the most flexibility. Applications delivered in BPA tools are more responsive to business and process change, and behavior can be defined by the end-users themselves.

Business Process Automation Approach	
Objectives	Further extension, or alternative to, full object approach. Abstract business process from application code.
Development Effort	BPA skills.
Platform	WebSphere Application Server Version 4.0, Enterprise Edition. Use of J2EE, WebSphere® Business Integrator, MQSeries Workflow, MQSeries Integrator.
New service Implementations	J2EE. Extension to existing systems.
Connection to other resources	MQSeries, WebSphere Business Integrator, JCA, CORBA.
Organizations	Large enterprise.
Applicability	B2B, Long running processes, Complex multi-step processes.
Benefits of approach	BPA benefits. More business-oriented approach. Most comprehensive integration options.
Inhibitors	Web services workflow, orchestration and sequencing standards likely to take time to reach consensus and mature.

How Are They Made Available?

Though it is not a pre-requisite, Web services will be generally published via UDDI registries. Information on Web services such as the WSDL file could simply be emailed to the requestor. UDDI provides a standard structured approach to publishing services, and via the support provided in IBM's Web service development tools means that the process of publishing and finding Web services is largely automated.

There will be 2 basic models by which UDDI registries are used:

- **Public Registries** – As their names implies, public registries enable anyone to publish their services and for other, effectively anonymous users to locate them and understand how to use them. This is not to imply that anyone can use the actual service, as normal access restrictions will still apply from both a technical and business perspective. These are operated currently by IBM and Microsoft, and will be shortly joined by HP and SAP;
- **Private Registries** - Used within the enterprise or between a closed groups of trusted partners. In such an environment you could simply email others all the information required to use your Web services, but UDDI provides a dynamic mechanism by which they can be published and consumed.

IBM provides a UDDI implementation that organizations can install in-house to deliver a private registry. This is available via their alphaworks site, and is also distributed as part of their Web Services Toolkit.

Where Do They Go?

As mentioned earlier, a key reason for using Web services is to enable intelligent use of information rather than simply presenting it. From an abstract viewpoint we could say that the endpoint for a Web services is simple whatever consumes it, and that it matters little what that is or what it does next with it as long as they have permission. However, it aids understanding to identify a number of final endpoints and potential steps on the journey of a Web service

- A useful concept that we have promoted is the Business Service Bus[3], as has IBM. The Bus is primarily a logical concept that reflects a set of collaborating services that are shared within a specific domain. This may span the entire corporation, or a business unit, or a geography or a set of common characteristics such as a skill group, resource group, etc. All endpoints request their services from the Bus.
- Applications running on Web application server. The same applications that are providing Web services are consuming them too. For example this might be another server application participating in a B2B exchange of information between two organizations, or internal EAI use.

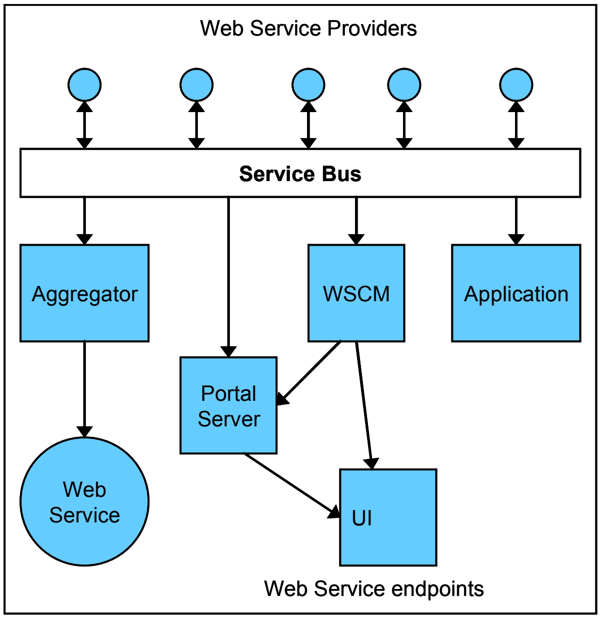

Figure 6 - Where Do Web Services Go?

- Applications running on client devices. For example office productivity software. One of the attractions of the independence of Web services is that they can be delivered directly to client applications, running on a wide variety of devices. Applications running on the increasing array of mobile devices such as PDAs and mobile phones will just as likely to be the endpoints for Web services as traditional PC hardware.
- Aggregated with other Web services and re-presented as a further Web services.
- Rendered directly as components that can be used in the presentation layer. The proposed WSXL (Web Services Experience Language) standard enables, amongst other things, Web services to be visualized directly in a suitable browser.
- Assembled and presented in a portal. Portal servers such as IBM aggregate, assemble and present information from the Web services via a browser. This approach may be standardized via WSXL.

Consuming Web services is of course much easier than providing one, as the requestor is only interested in the interface to the service, not the implementation. With the appropriate tools, such as those from IBM, Web services protocols and the complexities of XML are transparent to the developer consuming the service.

[3] The Adaptive E-Business, INTERACT, December 1999

How are Web Services Managed and Secured?

In some respects there is little difference in managing and securing Web services compared to any other Web-facing application. Actions on both service provider and requesters include:

- Monitoring the performance and availability of the Web services, and their implementation;
- Preventing unauthorized use via identity and authentication mechanisms;
- Encrypting incoming and outgoing messages where relevant;
- Logging activity;
- Prevention of viruses, hacking and other unwanted activities.

The tight integration of IBM's Tivoli™ products with their products that host the applications make them an obvious choice for managing and securing the implementation of Web services. Tivoli Web Services Manager also gives both service providers and requesters the capability to monitor Web services to ensure they are meeting service level agreements.

Though the wide use of existing HTTPS and Secure Sockets Layer (SSL) remains appropriate for Web services, the main activity now is for vendors to agree a standard security framework for Web service protocols that enables a variety of security technologies to be applied where appropriate. For example, IBM has proposed SOAP Digital Signatures[4] (SOAP-DSIG) as a way of defining the syntax and processing rules for digitally signing SOAP messages and validating signatures. This can be used together with SSL, to provide authentication and non-repudiation of SOAP messages[5]. Support for SOAP-SIG is included in their Web Services Toolkit. IBM also provides an XML Security Suite that provides a number of useful utilities for securing and monitoring XML documents. Proposals for XML security standards such as XML-Signature and XML-Encryption will play an essential role in Web services.

How Are They Developed?

There are various Web services toolkits on offer from each of the web server vendors. The best of these toolkits do an excellent job not only of handling the actual Web services, but also masking the complexity of XML-based protocols from the developer.

The use of WSDL is central to the development of both the implementations of Web services, and the applications that consume them. IBM's development tools do an excellent job of not only producing the WSDL when publishing a service, but also using it to generate proxies that make it easy to consume the service. If no implementation of the Web service exists, then a service implementation template can be generated to which the developer need only add the application specific code.

IBM has also proposed standards targeted at simplifying the use of Web services for the Java developer. These are, UDDI4J, and WSDL4J, with "4J" representing their use "for Java". These provide client APIs that mask the complexities of the UDDI and WSDL from the developer allowing them to effectively deal with them as standard Java class structures. Through the Java Community Process they should ultimately find their way into the J2EE specification.

IBM has also now consolidated their various Web service tools and technologies into their application development suite WebSphere® Studio. This will be available in two new versions:

- **WebSphere® Site Developer** - A tool for developing and managing Web sites that includes support for HTML, JSP pages, Java, servlets, rich media, and XML as well as Web services;
- **WebSphere® Studio Application Developer** - which adds full J2EE development support.

In addition, IBM also continues to make previews of new Web services technologies available via the alphaworks website.

Though such tooling is important for developer productivity and delivery of the actual Web services, it is also equally important to address the overall process and architecture for Web services. Whilst tools ensure that Web services are built correctly, architecture and process ensures that correct Web services are built. A subtle but important difference.

Organizations can minimize the risk of adopting new technologies by learning from the experience of others. IBM's jStart program assists organizations not only to implement new technologies, but also ensure that their delivery meets business need. The case studies on the jStart website demonstrate that with IBM's assistance, organizations are already achieving benefits from Web services.

[4] http://www.w3.org/TR/SOAP-dsig/

[5] http://www-106.ibm.com/developerworks/webservices/library/ws-soapsec/

Challenges

Delivering Service Level Agreements

I predicted some time ago a coming transaction explosion[6] as a result of the proliferation of increasingly intelligent and connected devices that would generate an order of magnitude higher number of transactions that organizations must deal with. I also argued that it required real-time behavior to deal with those transactions or get swamped in the backlog.

Web services provided the mechanism to do this, but as such are a two-edged sword. Yes they provide the benefits outlined earlier, but in return they will place high demands on the performance, scalability, and availability of the applications and underlying platform. Improvement in automation cannot be realized when the Web services are not available!

For the large part, this requirement will be met by the technologies that host the implementation of the service, i.e., the Web application server. Organizations must implement Web applications servers such as IBM WebSphere that are not only robust and reliable, but support the full component architecture necessary to deliver scalable Web services.

Phase one Web services protocols do not yet provide a standard for reliable delivery of messages. Organizations might choose to use early implementations of proposed standards that are already available in some environments. To minimize effort and ensure that they can be replaced by standards once agreed, this should be done via infrastructure components that can be used by any application.

Gaining Consensus of Trading Partners for Standards

The value of Web services will be increased in many domains when there is industry-wide agreement on the services that should be offered and the structure and meaning of the information provided. Fortunately, there seems no end of initiatives to define XML vocabularies and schemas for both vertical and horizontal industry domains, which can in turn be adapted for Web services.

Of course not all Web services need consensus between provider and requestor. To consume a Web service you do not necessarily have to agree it is the ideal structure, you just need to know how to use it. However, the ability to easily change service providers (especially automatically), or map them to your own applications is greatly simplified when both requestor and providers all conform to the same standard.

The need for consensus will be driven by the nature of a particular industry. Where the need is clear, organizations should consider working in relevant industry organizations to ensure that developed standards meet their needs.

Early Adopters of Phase Two Standards Proposals

Though candidates are already being proposed for additional standards that will delivered in phase two, it is likely that it will take longer to reach agreement than phase one due the more complex requirements.

In many cases implementations are already available for these proposals, primarily from the proposing vendor(s). Early adopters can take advantage of these technologies to build richer Web services in phase one, but should recognize that there is likelihood that compliance with final standards will require additional effort. Developers should make sure that any use of these early implementations is clearly isolated from applications to facilitate change.

Suitability of Existing Systems

Web services are likely to increase the attention on legacy rejuvenation projects to restructure and improve existing application code to make it easier to reuse and integrate. Existing systems that do not perform well will likely need a healthy dose of rejuvenation if they are to be used in any Web service scenario.

The suitability of existing systems to be used in the implementation of a Web services will vary greatly. Though the mechanics of implementing Web services are largely automatic, there is no guarantee that the existing systems align directly with the services required or can be wrapped without additional effort.

Part of this challenge may already have been addressed in the shift to client/server, or within EAI and/or various web access projects. However, two hurdles can still remain:

1. May not offer an interface suitable for real-time accessibility and integration that is a key driver for delivering Web services;
2. Existing interfaces to not map cleanly to the those exposed by the Web Service.

Some key determinants of the suitability of existing systems are shown in the following table:

[6] Transaction Explosion, Interact - CBDi Forum Journal, November 1999

	Ideal	**Has Potential**	**Constrained**
How well are concerns separated within the existing asset?	Clear separation of presentation, process, business rules, data access	Well separated client and server with clean server interfaces	Monolithic Tight dependences between client and server Business rules in client
Structure	Components Object-based	Client/Server	Monolithic
How accessible and usable are the existing interfaces?	Direct access to business rules Component Interfaces XML RPC	Direct access to data (care over bypassing relevant business rules)	Only through existing UI Batch
Reusability	Business Rules	Process (might not be suitable for style of Web Service) Data access	UI

Table 3- Suitability of Existing Assets

Some Key New Components

Whether accessing existing systems or building new software, there will be additional components that need to be implemented specifically to support the delivery of Web services. Even when delivering new software to support Web services there is a good case for isolating the code specific to delivering Web services from the rest of the functions of the systems. For example:

- **Conversation and state management** - The loosely coupled conversations across Web services, particularly in B2B scenarios, will require specific components to manage those conversations. This might be delegated to products specific to this purpose such as WebSphere Business Integrator;
- **Access Management** - Where Web services are exposed outside the organization there may be additional components required to control access, above the normal security capabilities. For example licensing control of charged Web services;
- **Aggregation and Transformation** - Web services might be an aggregation of information accessed from several existing interfaces. Web services might also typically expose single coarser-grained interfaces than the fine-grained interfaces of the component that implement them. The information will sometime need to undergo complex transformations, especially from older systems. This could be delegated to products such as MQSeries Integrator;
- **Isolation Web service protocols and technologies from change** - Web service standards will continue to evolve. Using early implementations of these standards is certain to require later upgrade.

4. Application - Common Scenarios

Now we know what they are, and how to provide them, lets look at some common scenarios widely applicable in any industry that are enhanced through the use of Web services, and are achievable today.

It would be convenient to match the types of implementation listed earlier to each of these scenarios except the complexity of Web services may have no relation to the complexity of the implementation required. The simplest Web service from the requestor's perspective may expose information that is nevertheless complex for the provider to deliver.

Internal Systems Interoperability – Web Services Based EAI

Scenario: The all-welcoming website invites customers to place orders for '24 hour delivery'. Yet days later customers find the orders still haven't arrived, there is no indication of the status of the order, and the etailers switchboard is jammed. Swamped by orders following the successful launch of one of the products they sell, the sales clerk has a backlog of orders that must be re-entered into the logistics' system.

Unfortunately, the website developer has no knowledge of the order processing, financial and other systems used in the back office. Extracts of information can be produced for replication each night but information presented to customers such as stock availability or order status remains 24-48 hours out of date.

Web services are often positioned in B2C or B2B scenarios. Certainly they are excellent solutions to requirements in these areas. However, many organizations still face more glaring challenges of interfacing their own diverse systems. Not only is efficiency improved, but improving the flow of information between internal systems can do so much more to deliver improved customer satisfaction - such as simply ensuring they deliver what customers want, when they wanted it - and yield more repeat business as a result.

The lack of compatibility between applications and platforms has led to a plethora of proprietary EAI solutions. However, they often lack the real-time behavior that is needed to provide instant commitment and feedback to customers who are one click away from going to the competition. Many organizations falsely believe they have solved the integration challenge because they have a mechanism that replicates data each day.

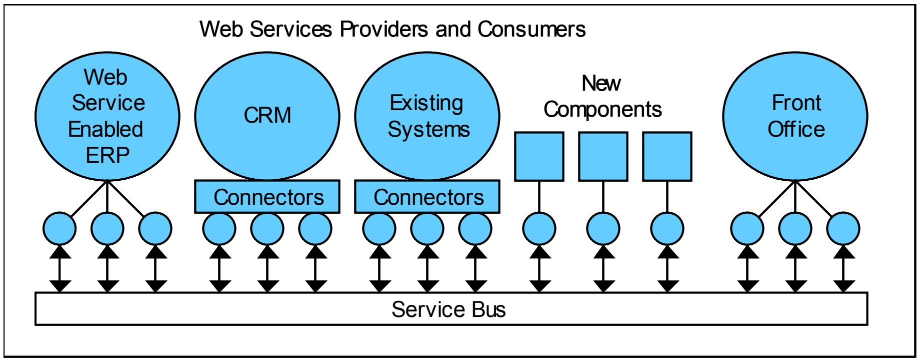

Figure 7 - Web Services Based EAI

The use of Web services will change the EAI marketplace. Whilst it is clear that the diversity of existing systems will remain for some time, the need for proprietary EAI solutions will reduce with the support of Web services on the all the platforms that host the applications, and with interfaces to common application packages now being made available as Web services. Current EAI solutions sometimes lack suitable interfaces to deliver real-time access to information, whereas using technologies such as COM, CORBA or RMI forces unnecessary technology dependences between provider and requestor.

One of the most common scenarios in many organizations is the connection of Windows-based front office systems, featuring both custom developed applications, packaged applications and elements of the Microsoft Office Suite, together with core back office systems running on a different platform. The back office systems might be new J2EE components running in IBM WebSphere, or existing applications running on a variety of platforms. Web services provide a mechanism by which n-tier client/server can be delivered using the same standard protocols across all tiers, and enable real-time access to back-office information into front-office systems.

Internal Systems Interoperability	
	Provider/Requestor
Motivation	Process automation and efficiency improvement Real-time access to remote information Reduced development effort via standardization
Web Service Implementation	Minimal to expose Web services providing suitable interfaces exist
Likely implementation Approach	Existing Systems as Web services Business Process Automation – for more complex scenarios. E.g. use of MQSeries Integrator
UDDI Registry usage	Private, Internal

Microsoft's support for Web services is therefore welcome, as many interoperability standard initiatives have previously lacked their participation. With both Microsoft and other platform vendors providing Web Service support, not only is there a standard mechanism for interoperability, but the implementation of Web services on the respective platforms enable developers on either side to continue to work in their familiar programming environment and request and provide services without having to learn new languages or tools.

Web services will not entirely remove the need for EAI products. Those that evolve to support Web services protocols can play a valuable role in converting the interfaces of older applications and packages to Web services.

Business to Business Integration

Scenario: Even when the e-commerce operation has improved integration to ensure that orders are dealt with efficiently, and the integrity of their internal processes are improved; they are still dependent upon the supply chain. They do not want to decline orders if there is stock in the supply chain, but neither do they want to accept orders for goods they cannot supply. How can they commit to orders in real-time without really knowing if their supplier has the goods? Responding to customer complaints about lack of fulfillment they increase warehousing space, logistics staff and inventory but as a consequence find that costs rise, reducing their ability to hold a competitive price advantage over traditional bricks and mortar stores.

The EDI solution in place has automated the transfer of orders to the supplier but doesn't provide a real-time 2-way flow of information. The supplier has provided self-service facilities through a web browser that allows the retailer to check availability and order status, but this manual process is expensive and inefficient and not customer facing.

Customers are also dissatisfied with delivery methods. When will the orders be delivered? Must they wait patiently at home till they arrive? And why must they wait for so long to receive goods that are readily available in the nearby store? A logistics company wants to offer a new service that enables the end-customer to negotiate a suitable delivery time but this requires integration of both the logistics and the e-tailers systems.

Instead of maintaining higher stock levels to ensure that orders are not declined, the retailers system can use Web services to check availability and place orders with the supplier in direct response to customer's purchases. Improving supply chain efficiency and the automation of other B2B processes should provide motivation to both provider and requestor of the Web Service. The scenario illustrated however, provides further motivation to the retailer in that it enables them to provide a better service to their end customers.

B2B shares fundamental integration issues with internal systems. That is, differences in technologies, protocols and information structures at each end of the conversation. In the case of B2B, the major effort is not the technical delivery of Web services, but firstly agreeing and aligning business processes, and secondly adapting operational systems to support the real time behavior required. B2B integration also faces a further set of challenges.

For example

- **Non-repudiation** - Non-repudiation of transactions is of greater importance in B2B than internal systems. Organizations should not be able to deny having sent or received information via Web services;
- **Long-running transactions** - Managing state and provide rollback and recovery mechanisms in multi-step transactions that make take place over extended periods between different organizations;
- **Security** - Issues of identity and security are greater when systems are exposed externally;
- **Quality of Service** - Increased reliance on 3rd party systems.

These issues are not specific to Web services, but to other B2B approaches too. However, attempting to implement real-time behavior via Web services challenges existing mechanisms. Although work on standards is taking place, organizations must be prepared to put extra effort in themselves to overcome these challenges during the first phase of Web services. Whilst developer effort can overcome the technical challenges, matters of trust will mean that B2B Web service activity will often be limited to pre-qualified trading partners.

It will not always be possible, or suitable to use an RPC approach to implement real-time transactions to distributed systems that have no common transaction model. An asynchronous message passing approach is often more suitable in loosely connected situations. B2B scenarios will also often involve long-running transactions where compensating transactions are more appropriate than rollback. In the near term many organizations will find that this behavior needs to be implemented by developers in their applications.

Business to Business Integration		
	Provider	**Requestor**
Motivation	Efficiency, timeliness of process	Efficiency, timeliness of process Improved service to end-customers
Web Service Implementation	Whilst it remains straightforward to deliver Web services from a developer perspective, significant effort is required to define collaborative Web services.	Whilst it remains straightforward to request Web services from a developer perspective, significant effort is required to define collaborative Web services.
Likely implementation Approach	Business Process Automation	Business Process Automation
UDDI Registry usage	Private – accessible only by partners Public – for simple B2B	

Plug-in Business Services

Scenario: In their e-commerce system, the international retailer wants to display prices in local currency but has to manually input exchange rates on a daily basis. They also want to check whether the customer has provided a valid shipment address, and if the formatting is correct for a foreign destination. They have a system that uses postcodes for domestic customers, but their national post office cannot provide information for other countries. They also want to implement a 'call me now' function on their website that puts customers in touch with a human operator, even from overseas, but realize the investment in telephony equipment would be high and not scalable.

Some business services consist of simple requests that lend themseblves readily to the current capabilities of Web services. Each of the scenarios identified above is already being solved by organizations offering them as Web services.

There are also many straightforward services that are provided today but use proprietary protocols, such as credit authorizations. Provision via Web services would greatly simplify their implementation.

Both providing and requesting these types of services is relatively straightforward. Easiest to implement is when the type of information being provided doesn't require access to back end systems, or vary in real time. Even where it does, as long as the information is just being retrieved but not updated, it is still straightforward to deliver using current Web services technology.

Even so, delivering simple Web services like these can still introduce new challenges. For example:

- **How are they paid for?** The adverts that funded the provision of overseas address look up when accessed via a browser are not displayed in Web services. Charging on a term rental basis is much easier than per execution, though Web services metering solutions do exist;
- **Access control** - If they are going to be charged, how do you control access? Identify and security issues surface again;
- **Quality of service** - The simpler and easier the service, the more frequently it will be used. Scalability and reliability needed.

There are an almost infinite number of Web services that might be provided as plug-ins that enable the service requestor to increase the functionality of their systems with minimal implementation effort. However, this does not necessarily mean there is an obvious motivation to provider them. They are most likely where they support some wider business service from the provider (e.g. the bank provides its customers the currency exchange rate service), or where the provider can see some obvious financial return by charging for the service.

Plug-in Business Services		
	Provider	**Requestor**
Motivation	Revenue – charged services Part of wider business service already provided to requestor	Increased functionality for minimum effort Improved service Reduction in manual process

Web Service Implementation	Minimal - providing source of information is straightforward to access Increasing to moderate with metering, access control, high transaction volume	Minimal - simple request response
Likely implementation Approach	Depends on source of information. - From Website to Web services - Existing Systems as Web services	
UDDI Registry	Public	

Enabling Employees

Scenario: The knowledge worker would like to interact with information from a variety of sources, including external ones, in order to take better-informed buying decisions. How is that new product selling, what are customers reactions so far, should they take up the supplier's end of quarter special offer? A colleague would like to use that same data to monitor sales and alert management to exceptional conditions. Meanwhile, the call center operator simply needs to get approval from another employee to exceed normal credit limits. Ideally it needs to be handled immediately but there is no reply from their phone.

The program-to-program capabilities of Web services will commonly be used to improve business process efficiency though greater levels of automation. Fortunately, humans are not fully removed from the process. Employees will still initiate processes, monitor and react to their status, interact with them on behalf of others using alternative channels (e.g. via call centre, bricks and mortar stores), and be presented with decisions and exceptions that are not resolvable by the automated process.

The applicability of Web services is not limited to applications pre-built by professional developers. Employees will interact with Web services via a number of mechanisms including,

- **Pre-built Applications** - The previous scenarios benefit employees by providing them with applications that can more accurately interpret and present information regardless of its source;
- **Portals** - Some employees will also want to assemble their own solutions in support of specific activities. For example IBM WebSphere® Portal Server enables them to bring together information from a variety of sources via Web services enabled portlets;
- **Desktop Productivity Software** - Emerging desktop productivity software can use Web services via a point and click interface enabling them easily integrate information via Web services into documents or spreadsheets, or to build their own desktop solutions;
- **Devices** – A new generation of mobile devices that support Web services that are now becoming available also means that these applications can be delivered to employees regardless of their location.

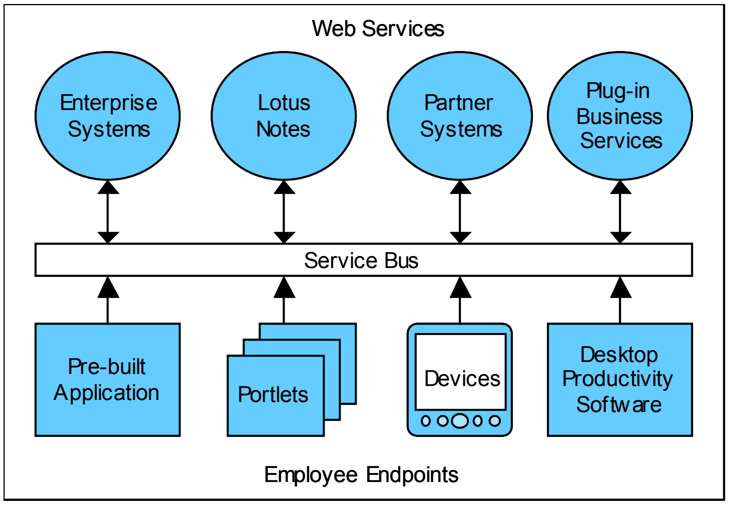

Figure 8 - Enabling Employees with Web Services

In addition, there are a variety of other ways that Web services can be applied to support employees such as:

- Calendaring. Sharing calendars via Web services enables applications to automatically schedule events;
- Notification. Employees can be alerted to events that require their action. Notification Web services will remove the need of the application to know what actual mechanism should be used at any given time. And vice-versa when the employee makes a response;
- Management processes. Not all workflows will be implemented within applications built by professional developers. Often managers will want to build workflows to automate office tasks that they can quickly change as circumstances dictate. However, getting them to work in collaboration with core operational systems is usually difficult because of the differences in technology.

Some of these scenarios will be familiar to users of Lotus Notes. Standard Web services will enables applications running today on Lotus to be more easily integrated with other applications, both internally and externally to the organization. To support this, Lotus is providing a Web Services Enablement Kit that provides Domino, Sametime, LearningSpace, Knowledge Discovery System, Domino Workflow, and Lotus Enterprise Integrator with Web services capability.

5. Conclusions

Why IBM?

With widespread industry endorsement of Web services it is increasingly difficult to find a vendor who does not claim some measure of support. However, we caution that this support can be very shallow and typically does not extend beyond basic SOAP, WSDL and UDDI support, if even that. Some vendors have done little beyond applying the same Apache SOAP web server as all their competitors (the implementation of which was contributed initially by IBM).

So why use IBM in preference to other vendors? I would recommend IBM as a leader in Web services for the following reasons.

- **Driving standards** – IBM has taken a leadership role in Web service standards. IBM's customers benefit not only from the earliest access to new Web services technologies that IBM are developing, but also confidence that the solutions they build will be based upon open standards.
- **Depth of Implementation** – The support of Web services in IBM products is second to none. This can be seen both with the range of different products that IBM are enabling with Web services, and also in the number of new Web services technologies they are implementing.
- **e-business experience** – IBM's vision of Web services is a result of their experience of working with many customers to implement e-business. IBM understands how to apply Web services and the opportunities they bring.
- **Professional Services** – The IBM jStart program enables customers to minimize the risk of introducing Web services.

Right Here, Right Now

New technology always brings risks. Will it cross the chasm and be widely adopted? Will systems built by early adopters need to be upgraded as the technology evolves to maturity?

There is always uncertainty, but those willing to take the risk stand to gain the greatest competitive advantage. And though it is relatively early in the Web services cycle it looks clear that adoption will be widespread. The breadth of industry endorsement is impressive. But more importantly organizations are already delivering commercial Web services, and in turn improving the way they interact with their customers and partners

This report was authored by Lawrence Wilkes, Principal Analyst of CBDi Forum and can be contacted via email at lawrence.wilkes@cbdiforum.com

Related IBM Resources

IBM General Web Services Information sites

alphaworks	http://www.alphaworks.ibm.com/
Web services site	http://www.ibm.com/webservices
developerWorks Web Services Zone	http://www.ibm.com/developerworks/webservices
jStart program (contact to begin Web services engagements)	http://www-4.ibm.com/software/ebusiness/jstart/

IBM Product Specific

DB2	http://www-4.ibm.com/software/data/webservices/
Lotus	http://www.lotus.com/home.nsf/welcome/webservices
Tivoli	http://www.tivoli.com/products/solutions/e-business/webservices.html
WebSphere	http://www.ibm.com/websphere

Insight for Web Service and Component Based Business

www.cbdiforum.com

CBDI Forum Limited, PO Box 536, Guildford, Surrey, GU4 7BX, United Kingdom