

Minder Chen

# An analysis of the driving forces for Web services adoption

Published online: 10 August 2005  
© Springer-Verlag 2005

**Abstract** Web services are supported by major IT vendors and have been adopted by some enterprises in various applications. However, due to the hype surrounding Web services, information technology (IT) personnel and business managers often have difficulty assessing the potential uses, impacts, and benefits of Web services. Based on literature review and technical information, as well as field and Web-based case studies, we have developed a framework for analyzing the driving forces for Web services adoption. The framework and detailed benefits analysis model can be used by IT and business strategy planners to identify technical options and business opportunities, as well as to formulate Web services implementation strategies.

**Keywords** Web services · Technology adoption

## 1 Introduction

Business managers and information technology (IT) professionals often have different ideas and interpretations regarding Web services. Some technical IT professionals view Web services as a technology breakthrough while others view them as a natural step in the evolution of distributed computing technologies (Hanna 2003). Few business managers understand the implications and impacts of Web services on their business strategies.

---

M. Chen  
Decision Sciences and Management Information Systems,  
School of Management George Mason University,  
Fairfax, VA, 22030, USA  
E-mail: mchen@gmu.edu

Based on literature review, technical information analysis, and field and Web-based case studies, we have developed a framework for analyzing the driving forces for Web services adoption. The framework, as depicted in Fig. 1, classifies macro-level driving forces of Web services adoption into three major categories: (1) technical factors, (2) stakeholders of Web services standards and technologies, and (3) perceived benefits. Web services standards are considered an important technical factor. The results of Web services adoption and implementation decisions may include targeted applications, technical architectures, development strategies, and deployment strategies. This research does not study factors that affect an individual firm's adoption decision, such as firm size, existing IT infrastructure, and IT personnel's current skill set.

These driving forces interact with each other. For instance, technical merits have contributed to the perceived benefits of Web services, motivating various stakeholders to adopt Web services in their products or applications. This paper examines each category of these major driving forces and discusses the interactions among them. Section 2 describes the Web services in the context of a service-oriented architecture (SOA). Section 3 analyzes the technical factors that are driving Web services adoption. Web services are described in the context of a three-tier architecture providing support for loosely coupled applications integration. The major stakeholders of Web services standards and technologies, as well as their interactions in the adoption process of Web services, are discussed in Sect. 4. Section 5 analyzes the business benefits of Web services based on literature research and analysis of the technical merits of Web services. Linkages between technical-

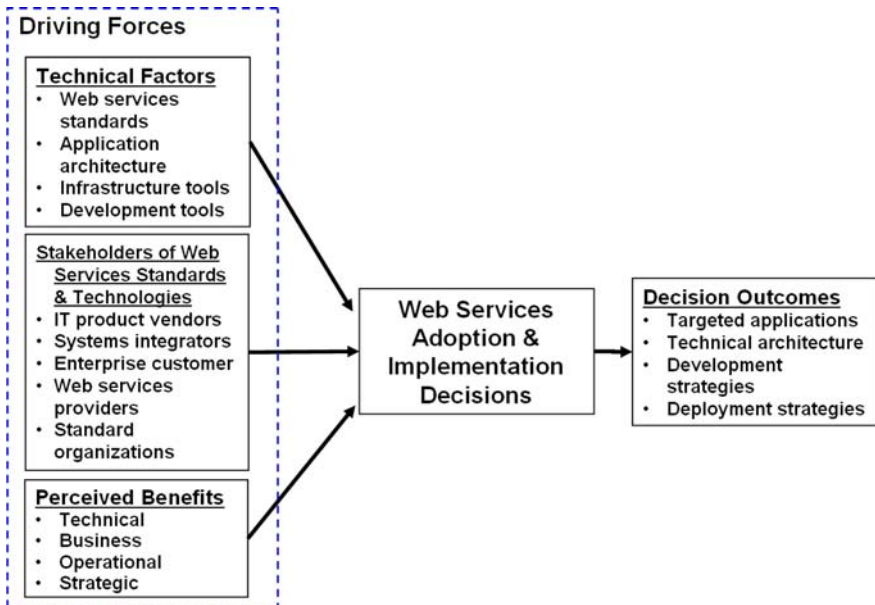


Fig. 1 Driving forces of Web services adoption

oriented benefits and business benefits due to Web services adoption are identified. Section 6 concludes the paper with a discussion of future trends of Web services adoption and implementation.

## 2 Service-oriented architecture and Web services standards

The core idea of Web services, using SOAP messaging protocol to invoke software methods in remote systems (W3C 2000), is often described by some technologists as remote procedure calls (RPC) over Internet protocols (e.g., HTTP). A more detailed definition by the W3C Web Services Architecture Group stated that (Champion et al. 2002):

*“A Web service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML based messages via Internet-based protocols.”*

This W3C definition tends to be narrow and technical and does not point out the benefits of Web services. The Stencil Group, an IT research firm, proposed a broader definition of Web services (Stencil Group 2002):

*“Loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols.”*

The term “loosely coupled” implies that Web services are independent of any programming languages, platforms, and object models. Web services can be easily deployed to provide interoperable software functions over the

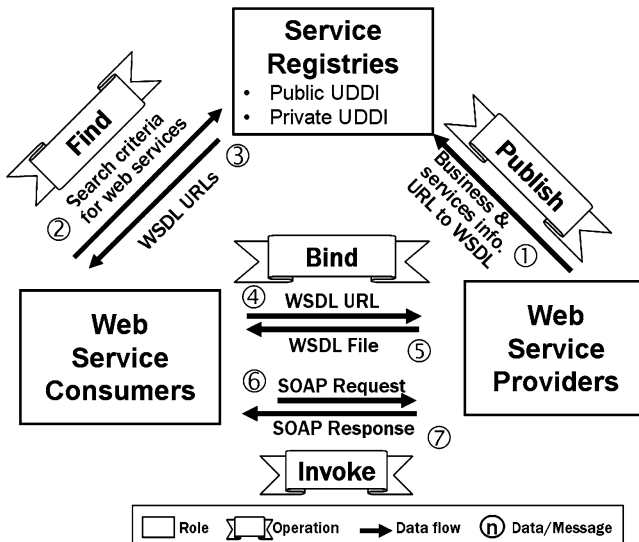


Fig. 2 Service-oriented Web services architecture

ubiquitous and low-cost intranet and the Internet. From a distributed computing architecture viewpoint, Web services are emerging enabling technologies for implementing service-oriented and component-based application architectures. From a business-oriented and conceptual viewpoint, Web services can be used as basic building blocks for developing dynamic e-business processes that are composed of discrete tasks implemented as Web services. These services can be distributed widely throughout a network of value-added Web services providers.

Web services provide a standard-based approach to implementing distributed software components. Via Web services, data and business logic services can be offered over standard Internet protocols to applications programs inside a firm or across enterprises.

Web services are a set of standards to form a SOA, as depicted in Fig. 2. This architecture models the interactions among Web service providers, Web service consumers, and service registries (Champion et al. 2001). The interactions involve publishing, finding, binding, and invoking Web services. Interfaces to a Web service implementation can be described by Web Services Description Language (WSDL). A Web service provider can publish a Web service defined by WSDL to a service registry such as UDDI (Universal Description, Discovery, and Integration). WSDL is an interface definition language specified in XML format for “describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information (W3C 2001).” WSDL is similar to the Interface Definition Language (IDL) in CORBA. WSDL is usually generated automatically by Web services development tools.

A Web service requester (consumer) can search and find WSDL access points of appropriate Web services provided by a Web service publisher from a UDDI directory. The service consumer can retrieve the WSDL file from the Web service publisher and use the service description to bind with the Web service by generating a client-side Web service proxy. Then, a programmer can use the Web service by invoking specific Web methods/operations of the Web service as if it is an imported class. Behind the scenes, at run time, SOAP messages will be sent via the Web service client proxy to the service provider to invoke these Web operations and optionally receive SOAP responses. By conforming to Web services standards, software components can be accessed by applications from customers and business partners independent of hardware, operating system, and programming language.

The UDDI depicted in the SOA architecture is a standard that enables companies and applications to easily and dynamically find and use Web services over the Internet or intranets (UDDI.com 2003). There are public UDDI registries where Web services and other business services can be registered. Few companies are actually using UDDI dynamic services discovery and integration in real business applications. A browser-based interface to the UDDI registry can be used to publish and search a Web service while programming interfaces, implemented as Web services, are available to automate these activities. Private UDDI registries are often created to support enterprise application integration (EAI) efforts. We can also build industry-specific UDDI registries to support vertical markets for

Web services. Business related standards such as ebXML and RosettaNet (2003) are also critical to the adoption of Web services in a B2B scenario.

### 3 Technical factors driving Web services adoption

Web services technologies emerged from the lineage of the distributed computing evolutionary path. RPC, DCOM from Microsoft, CORBA, and Remote Java Method Invocation (RMI) all use binary messaging formats; therefore these messages cannot easily pass corporate firewalls. Web services often use HTTP protocol that is widely deployed in enterprises. The standard HTTP port 80 is usually not blocked by firewalls. Web services are independent from any programming languages and operating systems, while RMI is limited to the Java language and DCOM is limited to the Windows platform. XML based messaging is used in Web services and it adds extra overhead in bandwidth and processing time; however, XML messages are much easier to process from the software development viewpoint.

Web technologies have evolved from static Web sites using HTML for document format. Dynamic Web sites use server side programming to create interactive Web applications to be used by users directly. Web services currently are designed to be consumed by other programs remotely, i.e., program-to-program communication. The integrated programmable Web enabled by Web services can automate many business processes inside or across corporate borders. It may reshape the business landscape by allowing enterprises to reconfigure their business relationships with trading partners dynamically once the adoption of Web services has reached a critical mass. This section analyzes the technical factors that are driving Web services adoption by examining Web services as a new middle-tier technology (Lea

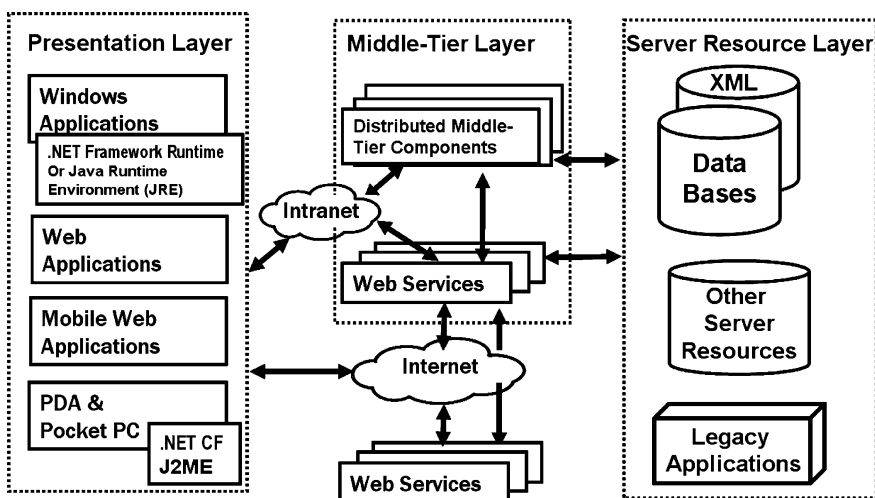


Fig. 3 Web services in the context of three-tier software architecture

and Vinoski 2003). Web services usage is discussed in the context of a three-tier technical architecture framework.

Web services can be used as a wrapper to expose functions of legacy systems to applications inside and outside enterprises. They can be accessed by a variety of client devices and applications including mobile applications. Web services can also be viewed as a new middleware technology for building middle-tier software components implementing business computation and data access logic. A Web services-centric three-tier architecture as depicted in Fig. 3 is discussed in detail in the following:

(a) *Server resource layer* The server-side resources in a three-tier architecture can be exemplified by database management systems and the data sources they manage. Data sources stored on servers may include XML files and documents stored in a variety of formats. We label this layer as the server resource layer to indicate that it may include server-side resources such as email servers, directory services, legacy applications, etc. Some architects use the term “data access layer” to refer to the software components that wrap around the data stored in the database. This type of software component is classified as the middle-tier layer in this architecture. Web services are often used to wrap around resources at this layer. Some Web services need to access server-layer resources to render their services. For example, a credit rating Web service needs to have access to the customer credit history database in order to implement its credit rating services.

(b) *The presentation layer* Since Web services provide only programmable interfaces, they can be consumed by Web services clients that may provide a user interface to end users. The presentation layer (i.e., Web services clients) can be implemented in the following ways: (1) GUI client applications: GUI-based client/server applications provide rich user interface elements and interactions. These are often used for the internal applications of an enterprise. (2) Browser-based Web applications: Web presentation layer components implemented by server-side scripting languages (e.g., ColdFusion, ASP, JSP, or ASP.NET) are running on the server side. HTML or other appropriate presentation formats will be generated dynamically by these server-side programs to the requesting user’s Web browser for rendering the returned result. (3) Mobile clients: Mobile devices such as cell phones and personal digital assistants (PDAs) can also consume Web services. There are two approaches for mobile clients to consume Web services. First, mobile micro-browsers are used to access Web-based applications that often use server-side programs to render results in a format (e.g., WML) appropriate for the requesting mobile device. Secondly, a native mobile application is run as a Web services client. Java 2 Micro Edition (J2ME) is needed to run a Java application on the mobile device while .NET Compact Framework is needed to run a .NET mobile client application. Both J2ME and NET CF support the SOAP messaging protocols on the client side. T-Motion online, one of the largest implementations of Web services today, is an example of a Web services-driven mobile Web portal site for wireless subscribers to access content offered by 100 providers (McDaniel and Kelly 2002). All three

presentation layer approaches can consume Web services to implement business functions encapsulated in these Web services.

(c) *Middle-tier components* In the traditional three-tier architecture design, the middle-tier components are usually implemented as Enterprise Java Bean or Microsoft COM components running on distributed servers. However, such an approach will limit presentation-tier to Java implementation or Windows platforms. Using Web services to implement the middle-tier components allows them to be accessed by any client applications. Web services technologies are geared towards building loosely-coupled component solutions that span various computing platforms. Web services are non-visual objects often consumed by front-end applications. Web services running on the middle tier can invoke software components implemented as DCOM or CORBA objects deployed on the same local area network. They can also invoke Web services hosted elsewhere on the Internet. Understanding Web services' role as a middle-tier software component technology in a distributed computing environment is critical for enterprises to evaluate the potential development strategies (e.g., build, purchase, or subscribe) and limitations of Web services, such as reliability.

You can use any language and application environment to implement Web services. J2EE and .NET are the two most common environments for Web services implementation. In the J2EE environment, the Java API for XML-based RPC (JAX-RPC) is the Java API for developing and using Web services (Sun Microsystems Inc. 2003). A Web service is a server application that implements as a Java class that is available for clients to call. It is deployed on a server-side Java servlet container such as Tomcat or a J2EE Enterprise JavaBeans container supported by many J2EE application servers such as IBM's WebSphere, Oracle's iAS 9i, or BEA System's WebLogics.

Application frameworks for building enterprise applications can be classified broadly into two major development platforms: Microsoft .NET and Java 2 Platform Enterprise Edition (J2EE). .NET is supported by Microsoft's .NET Server products and J2EE platform is supported by various application servers from several vendors. Web services can easily be deployed on these two platforms that have additional features supporting the development of enterprise-level Web services applications. Security, state management, auditing, reliability, transaction management, and message queue services are common features supported by the underlying platforms (Coyle 2002). Currently, developers mostly rely on the functionality provided by these development tools and deployment platforms to implement these features. Standards such as WS-ReliableMessaging (Ferris and Langworthy 2003), specifically designed to address issues related to the use of Web services in mission critical enterprise-level applications, are emerging (Arkin et al. 2003).

Web services are loosely coupled, i.e., platform-, language-, and vendor-independent. Therefore they can be developed by a firm to be used internally for enterprise application integration (EAI) or to be used by trading partners for business-to-business (B2B) integration. An increasing number of ERP and CRM systems have started to expose certain functions of their appli-

cations as Web services (Mello 2002). Third party Web services developers may also build Web services such as currency exchange rate, news feed, short message services, and language translation. Third-party Web services may be hosted by Web services brokers such as Grand Central Communications at GrandCentral.com (2003).

#### 4 The stakeholders of Web services standards and technologies

The basic components of a Web services “standard stack” consists of XML, SOAP, WSDL, and UDDI described earlier. There are additional standards that address issues related to security, reliability, document attachment, transaction processing, etc. The development of these additional standards is still in the early stages. Adoption of Web services for enterprise-level mission critical applications depends on further development of these standards and vendor implementation of existing and emerging standards in their infrastructure platforms, products, and development tools.

The major stakeholders (i.e., IT user community, IT product vendors, and standard setting organizations) form an IT value chain that is driving the development and adoption of Web services. As depicted in Fig. 4, these stakeholders are interacting and supporting each other in Web services development and adoption. The interdependency and driving forces from the perspective of these stakeholders are discussed in the following:

1. IT user community: This community consists of enterprise IT users, as well as systems integration firms and internal IT units that support IT. At the end of the value chain are customers who demand integrated services, thereby forcing enterprises to build integrated and multi-channel applications that can interact with customers at various touch points. Enterprise IT users who spend a lot of resources on these integrated projects view Web services as an important enabler for systems integration. The standard-based approach to interoperability is appealing to IT users such that it has become a main driving force for Web services adoption. Enterprise IT users who use IT to support their business operations are interested in Web services standards and technologies to ensure that various applications that they have developed based on different vendors’ products are interoperable with each other.

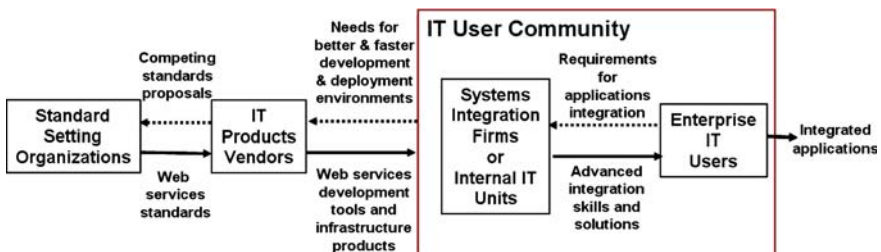


Fig. 4 The IT value chain driving Web services standards development and adoption



By adopting Web services standards and technologies, IT users can reduce their systems integration costs and development time. The primary outcomes of IT users' Web services adoption include the implementation of Web services in their applications and the deployment of IT infrastructures to support Web services. Network externality (effects) may explain the slow adoption rate in the earlier stage of Web services and the possible acceleration after a certain critical mass of Web services-enabled applications has been developed (Liebowitz and Margolis 1998). Adopting standard-based solutions and software prevents IT users from being locked into any particular vendor's solutions, therefore increasing IT users' bargaining power with IT vendors.

2. IT product vendors: Most IT product vendors prefer proprietary solutions so that they can lock in customers. However, since IT users are demanding interoperability via Web services, IT software vendors have a vested interest in participating in standards development and implementing standard-support features in their software products (Holloway 2003). For example, IBM and Microsoft have been working together aggressively along with other vendors to develop several critical Web services standards. The open standards strategy of IBM and many IT vendors can be described as "cooperate on standards, compete on implementation (Bowles 2000)." The primary outcomes for IT vendors are the introduction of development tools or infrastructure solutions that provide direct support for Web services development and deployment. IT vendors often encourage their employees to participate in standards bodies. Some proprietary Web services specifications developed by vendors have been submitted to standards bodies to become formal standards. IT vendors sometimes even give up their intellectual property claims on these specifications to accelerate the standard development and adoption process (Boulton 2002).
3. Standard setting organizations: Standard setting organizations play an important role in setting and promoting standards. Members of these organizations often come from both IT vendors and IT users. IT vendors have a major presence in Web services standards setting bodies. Standards development speed is affecting the adoption rate for Web services (Holloway 2003). Two major Web services standard setting bodies are the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS). W3C developed XML and SOAP while OASIS developed UDDI. The Web Services Interoperability Organization (WS-I) is "an open industry effort chartered to promote Web Services interoperability across platforms, applications, and programming languages." WS-I focuses on creating a set of Web services specifications profiles that work together to support specific types of solutions and implementation guidelines for how these specifications should be implemented. It intends to play a critical role in *standard harmonization* by working with other standard bodies such as IETF, OAGI, OASIS, OMG, and W3C. Standards are a means to achieving interoperability among heterogeneous systems. Vendor adoption of standards often results in building

features supporting these Web services standards in their development tools (e.g., Microsoft's Visual Studio .NET) and infrastructure platforms [e.g., BEA Systems' (2003) WebLogic]. Support and promotion of standard-based tools by IT vendors in turn accelerates the adoption of these Web services standards by IT users.

## 5 Perceived benefits for Web services

Web services rely on standards to solve interpretability problems among disparate systems within and across organizations. Web services are based on widely adopted XML for wire formats and Internet protocols such as HTTP for transport. The cost of implementing Web services is marginal while the benefits of Web services can be tremendous. For example, a study conducted by IDC with seven IBM customers in the early stages of implementing Web services solutions found that major benefits projected over three years include a cost reduction of 39.7 million on an average investment of 1.8 million and 22 percent faster time to develop key new applications (Hailstone and Perry 2002).

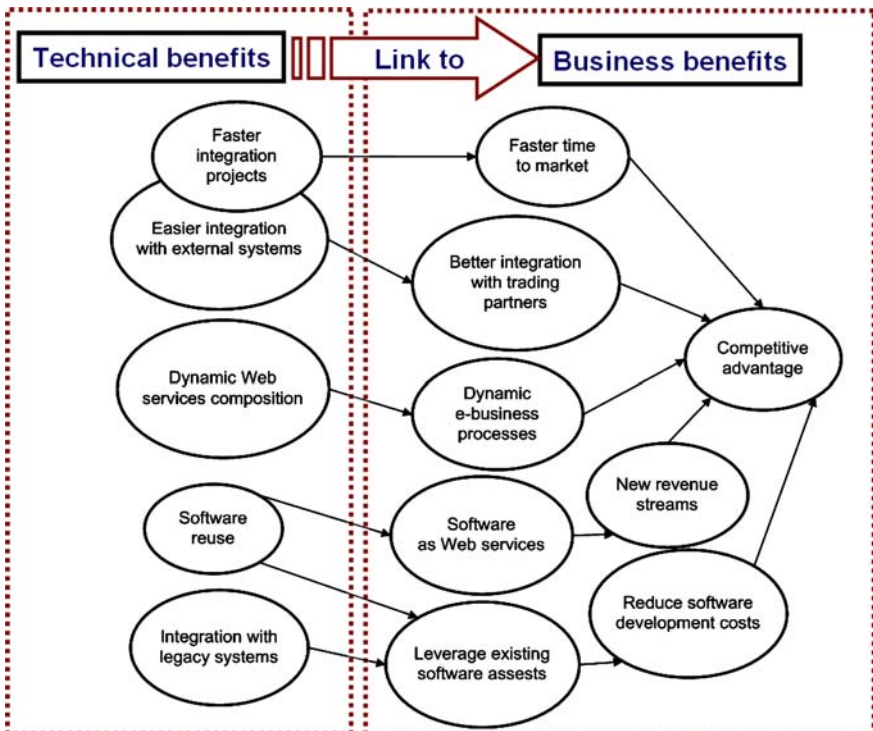


Fig. 5 A Web services benefit analysis model

The cost of implementing Web services includes IT infrastructures, development tools, and training. However, since Web services from a technical perspective are really evolutionary technologies, the costs tend to be incremental (CapeClear 2003). This approach is similar to other IT benefits research (Seddon et al. 1999). The development and deployment of Web services in various business contexts to take advantage of the technical merits of Web services as discussed in Sect. 3 are driving the business and strategic benefits of Web services. This section focuses on analyzing Web services benefits from the IT user community's perspective. Linking technical and business benefits and understanding the potential impacts of Web services are critical to the user organizations' Web services return-on-investment (ROI) analysis and implementation strategies.

A Web services benefit analysis model was constructed based on literature research regarding Web services adoption and implementation, including about 200 case studies reported by Microsoft's customers (Microsoft 2003). We have identified major technology benefits as well as their derived business benefits using an approach similar to that taken by Shang and Seddon (2000) in their study of ERP systems. This Web services benefits analysis model, depicted in Fig. 5, connects technical benefits to business benefits.

In the last few years, companies have invested a lot in IT assets and during the current economic downturn, many companies are focusing on integrating various existing applications and maximizing the return on investment of existing IT assets. Some companies have also adopted Web services for enhancing supply chain integration. Web services have become a key enabling technology for these integration projects. Several benefits of Web services depicted in Fig. 5 are summarized in the following:

1. Software reuse and software as a service. Software reuse can be achieved simply by developing common subroutines to be shared among developers. Using object-oriented programming languages, class libraries can be developed and reused. When an extension of a class library is needed, a programmer can create a derived subclass inherited from the original class without knowing its implementation. The increasing reuse of software implemented as Web services is due to the flexibility and broad applicability of Web services (Bloomberg 2002). Developers of Web services need not know exactly how these services are used in terms of user interfaces. They should focus on implementing business logic or server resources accessing functions with clearly defined program interfaces. Front-end applications may call Web services to access these important business functions and resources remotely. Companies may offer Web services to their business partners to increase supply chain integration and efficiency. By publishing their Web services via UDDI or other registries, companies can offer "software as a service." They can provide their Web services on a subscription or per-usage charge basis to generate a new revenue stream.
2. Integration with legacy systems and leverage software assets. Web services can be used to wrap around the functionality of legacy systems and expose these functions as service interfaces; therefore, they can be used to allow disparate systems to communicate with each other. Using Web

services means that all wrapper code should follow the same standards, and application integration becomes simpler, and hence, less expensive. These functions in legacy systems can be easily integrated into a new system to enhance its functionality. Internal business operations are streamlined because of better integrated enterprise applications. Since 75 percent of the expense in many large-scale software projects often goes toward integrating new systems with old systems, the most immediate promise of Web services is to simplify and accelerate the integration process of enterprise applications. The life of legacy applications will be extended, allowing companies to invest the resources saved in other new initiatives.

3. **Faster and more flexible integration with trading partners.** Web services are loosely coupled so that they can be used by applications written in various programming languages and running on different operating systems. Web services can wrap around legacy applications to render some of their functionalities and speed up integration projects. Therefore, businesses can create applications and services integrated with their business partners' systems at a deeper level to build a more streamlined supply chain. The flexibility in creating new connections with new business channel partners allows businesses to expand their market reach (Hagel 2002). Faster development time enables companies to quickly bring new services online to gain competitive advantages. Dollar Rent a Car's implementation of Web services is a case in point and illustrates such strategic benefits (Microsoft 2002).

Many of these benefits are related and interact with each other; therefore, there are certain degrees of overlapping among these benefits. The linkages between the technical and business benefits identified in our research may help IT/business strategic planners and architects to determine applications in which Web services should be deployed.

## 6 The future of Web services

Web services are not a revolutionary technology (Chappell 2003). They augment other technologies to extend existing applications. An incremental approach to Web services adoption is most likely in many user organizations. One survey of 758 executives found that 21 percent of companies used Web services in enterprise-wide or multi-enterprise integrations; 8 percent were in business unit-wide implementations; 41 percent were in incremental rollouts or pilots; 20 percent of companies were still thinking about Web services; and just 10 percent showed no interest (Line56 and Kearney 2002). The adoption of Web services looks promising based on various case studies and analysts' forecasts. The relatively rapid Web services adoption rate can be attributed to the following factors as elaborated in this paper:

1. IT vendors are strongly endorsing Web services standards. IT vendors are adding support of Web services to their development tools and

infrastructure platforms as part of normal software upgrades. They also offer new tools and platforms to hide some of the Web services implementation complexity from the developers.

2. The cost of investing in Web services infrastructure is marginal. This is particularly true for enterprises that already have an e-business infrastructure in place. Therefore, the adoption may even occur from the bottom-up. Once an application developed using Web services has proven beneficial, companies may launch an enterprise-wide effort to implement Web services.
3. Web services have a relatively smooth learning curve comparing to other distributed computing technology. This is particularly true for developers who have three-tier architecture design and programming experience. New development tools supporting Web services are also very helpful in shortening the learning curve.

However, several major barriers hinder Web services adoption and need to be addressed by potential adopters. These barriers include:

1. Business-level standards need to be developed to provide some consistency of the Web services that have been registered in the UDDI. Industry-specific Web services standards for business transactions or documentation (e.g., OASIS Universal Business Language) may emerge. Research in semantic Web services and ontology may also enhance current UDDI usage to facilitate intelligent Web services search (McIlraith et al. 2001).
2. Security is a major concern of many potential Web services adopters. There are no widely adopted Web services security protocols and standards. Many organizations use existing security measures such as HTTPS or encryption of entire SOAP messages to secure Web services communications. Dedicated standards such as Security Assertion Markup Language (SAML) are in development.
3. The quality of Web services is still in its infancy, so that the use of third party hosted shared Web services is not ready for prime time (Hagel 2002). Measurements of Web services quality may include evaluation of reliability, performance, and availability. Deploying Web services on a computer grid is a promising solution to improve Web services reliability and availability (Tan et al. 2003). The vision of software as services and the emergence of marketplaces for Web services rely on useful Web services reaching a critical mass and the maturity of Web services hosting infrastructures that are capable of providing reliable and secured Web services.

Web services will have a widespread impact on the IT industry and user organizations (SIIA 2002). More in-depth studies are needed to understand Web services adoption and the diffusion processes for different types of applications and industries. Additional theoretical and empirical research on Web services benefits and their impact on businesses may help enterprises evaluate Web services from a strategic viewpoint (Hoffman 2002).

## References

- Arkin A et al. (2003) Web service choreography interface 1.0 specification. <http://dev2-dev.bea.com/techtrack/wsci.jsp>
- A.T. Kearney and The Stencil Group (2002), The emerging web services market. <http://www.atkearney.com/main.taf?p=5,3,1,40>
- BEA (2003) Web services: introduction. Sept. 10, <http://www.bea.com/products/webservices/index.shtml>
- Bloomberg J (2002) The paradox of reusability, 30 January. <http://www.zapthink.com/flashes/01302002rhodes.html>
- Boulton C (2002) SOAP 1.2 Passage snagged by IP issues. 8 November, <http://www.internetnews.com/dev-news/article.php/1497091>
- Bowles A (2002) The open standards imperative: IBM's open-for-business strategy. IBM e-Business Software, March, [http://www-3.ibm.com/software/solutions/webservices/pdf/OpenStandards\\_wp.pdf](http://www-3.ibm.com/software/solutions/webservices/pdf/OpenStandards_wp.pdf)
- CapeClear (2003) EAI and web services lowering the total cost of integration. [http://www.capeclear.com/products/whitepapers/eai\\_ws\\_whitepaper.pdf](http://www.capeclear.com/products/whitepapers/eai_ws_whitepaper.pdf)
- Champion M, Ferris C, Newcomer E, Orchard D (2002) Web services architecture. W3C Working Draft, W3C, 14 November, <http://www.w3.org/TR/2002/WD-ws-arch-20021114/>
- Chappell D (2003) Lifting the fog on web services. <http://www.adtmag.com/article.asp?id=6395>
- Coyle FP (2002) XML, Web services, and the data revolution, Addison-Wesley, Reading
- Ferris C, Langworthy D (2003) Web services reliable messaging protocol (WS-Reliable-Messaging). 13 March, <http://msdn.microsoft.com/ws/2003/03/ws-reliablemessaging/>
- GrandCentral.com (2003) Grand central communications: features and benefits. September 2, <http://www.grandcentral.com/solutions/benefits.html>
- Hagel J (2002) *Out of the Box: Strategies for Achieving Profits Today & Growth Tomorrow Through Web Services*. Harvard Business School Press.
- Hailstone R, Perry R (2002) IBM and the strategic potential of web services: assessing the customer experience. IDC
- Hanna J (2003) Web services. 3 February, [http://hbsworkingknowledge.hbs.edu/pub-item.jhtml?id=3285&sid=-1&t=special\\_reports\\_cyber2003](http://hbsworkingknowledge.hbs.edu/pub-item.jhtml?id=3285&sid=-1&t=special_reports_cyber2003)
- Hoffman T (2002) Web services ROI remains tough to prove. *Computer World*, 28 October, <http://www.computerworld.com/developmenttopics/development/webservices/story/0,10801,75417,00.html>
- Holloway R (2003) Web service standards set the stage. *.Net Magazine*, June, [http://www.fawcette.com/dotnetmag/2003\\_07/magazine/columns/trends/default\\_pf.asp](http://www.fawcette.com/dotnetmag/2003_07/magazine/columns/trends/default_pf.asp)
- Lea D, Vinoski S (2003) Middleware for web services. *IEEE Internet Comput* 7(1):28–29
- Liebowitz SJ, Margolis SE (1998) Network externalities (effects). 2 November, <http://www.utdallas.edu/~liebowit/palgrave/network.html>
- Line56 and A.T. Kearney (2002) E-Business outlook: 2003. 22 August, [http://www.line56.com/download/ebiz\\_outlook-2003.pdf](http://www.line56.com/download/ebiz_outlook-2003.pdf)
- McDaniel T, Kelly D (2002) Case study: T-motion powered by Systinet, Hurwitz Group
- McIlraith S, Son TC, Zeng H (2001). Semantic web services. Special issue on the semantic web. *IEEE Intell Syst* 16(2): 46–53
- Microsoft (2002) Dollar rent a car breathes new life into legacy systems using .NET connected software. Microsoft .NET in real world, March
- Microsoft (2003) Microsoft case studies. <http://www.microsoft.com/resources/casestudies/>
- Mello A (2002) When CRM and Web services collide. Tech Update, ZDNet, 3 April, <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2859862,00.html>
- RosettaNet (2003) RosettaNet overview. <http://www.rosettaNet.org/>
- Seddon PB, Staples S, Patnayakuni R, Bowtell M (1999) Dimensions of information systems success. *Communications of AIS* 2
- Shang S, Seddon PB (2000) A comprehensive framework for classifying benefits of ERP systems. Proceedings of the 2000 Americas conference on information systems, Long Beach California, 10–13 August, pp 1005–1014

- 
- SIIA (2002) Web services today. <http://www.siia.net/divisions/software/webcasts/11-5-02/survey.ppt>
- Stencil Group (2002) Defining Web Services. [http://www.stencilgroup.com/ideas\\_scope\\_200106wsdefined.pdf](http://www.stencilgroup.com/ideas_scope_200106wsdefined.pdf)
- Sun microsystems inc. (2003) Java web services tutorial. <http://java.sun.com/webservices/docs/1.0/tutorial/index.html>
- Tan YS, Topol B, Vellanki V, Xing J (2003) Business service grid Part 1: introduction, <http://www-106.ibm.com/developerworks/ibm/library/i-servicegrid/>
- UDDI.com (2003) About UDDI, <http://www.uddi.org/about.html>
- W3C (2000) Simple object access protocol (SOAP) 1.1. W3C Note, 08 May, <http://www.w3.org/TR/SOAP/>
- W3C (2001) Web services description language (WSDL) 1.1. W3C Note, 15 March, <http://www.w3.org/TR/wsdl>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.