THE IMPLICATIONS AND IMPACTS OF WEB SERVICES TO ELECTRONIC COMMERCE RESEARCH AND PRACTICES

Minder Chen
School of Management
George Mason University
mchen@gmu.edu

Andrew N. K. Chen
W. P. Carey School of Business
Arizona State University
andrew.chen@asu.edu

Benjamin B. M. Shao
W. P. Carey School of Business
Arizona State University
benjamin.shao@asu.edu

ABSTRACT

Web services refer to a family of technologies that can universally standardize the communication of applications in order to connect systems, business partners, and customers cost-effectively through the World Wide Web. Major software vendors such as IBM, Microsoft, SAP, SUN, and Oracle are all embracing Web services standards and are releasing new products or tools that are Web services enabled. Web services will ease the constraints of time, cost, and space for discovering, negotiating, and conducting e-business transactions. As a result, Web services will change the way businesses design their applications as services, integrate with other business entities, manage business process workflows, and conduct e-business transactions. The early adopters of Web services are showing promising results such as greater development productivity gains and easier and faster integration with trading partners. However, there are many issues worth studying regarding Web services in the context of e-commerce. This special issue of the JECR aims to encourage awareness and discussion of important issues and applications of Web Services that are related to electronic commerce from the organizational, economics, and technical perspectives. Research opportunities of Web services and e-commerce area are fruitful and important for both academics and practitioners. We wish that this introductory article can shed some light for researchers and practitioners to better understand important issues and future trends of Web services and e-business.

Keywords: Web services, e-commerce, e-business

1. Introduction

E-Commerce or E-business generally refers to the conduct of business transactional or managerial activities using Internet technologies via the Web. As e-commerce models (e.g., B2C, B2B) and applications have been widely employed in today's business environment, a new movement called *dynamic e-business* has been urged to advance e-commerce applications to the next level by simplifying business interaction over the Web through effective and widely accepted messaging and data encapsulation standards. Gisolfi (2001) defined *dynamic e-business* as the next generation of e-business focusing on the integration and infrastructure complexities of B2B by leveraging the benefits of Internet standards and common infrastructure to produce optimal efficiencies for intra- and inter-enterprise computing. Maruyama (2002) referred to *dynamic e-business* as the third wave of e-business evolution (after B2C and B2B). Coupled with the technical needs of being able to externalize a company's business processes in a standard way, it is now possible to select business processes provided by outside companies and integrate these processes to dynamically form new applications (or business flows). Web services that consist of a set of standards and supporting technologies are the enablers for implementing dynamic e-businesses. Essentially, Web services propose a service-oriented paradigm for computing in which distributed, loosely coupled services collaboratively implement business processes and can be accessed via the Internet by end users. The paper by Chen

and Meixell (2003) published in this special issue addresses several important design and implementation issues of supporting a dynamic e-business process in the context of an e-procurement application.

The use of Web services over the intranet and Internet has increased rapidly. Web services are used to support application-to-application communication and to address interoperability issue for systems integration project, particularly in the context of electronic commerce and e-business. These Web services provide a standard-based approach for different software applications or components involved in supporting real-time information retrieval or presenting dynamic context-driven information to the user. The early adopters of Web services are showing promising results such as greater development productivity gains and easier and faster integration with trading partners. However, there are many issues worth studying regarding Web services in the context of e-commerce. This special issue of the JECR aims to encourage awareness and discussion of important issues and applications of Web services that are related to electronic commerce from the organizational, economics, and technical perspectives.

2. Web Services Standards and Infrastructure

Web services rely on a set of standards to support interoperability among applications developed in different languages and running on different platforms or operating systems. One way to understand Web services is to understand Web services standards. Core Web services standards including SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration), and other emerging ones will be discussed in this section.

2.1 Web Services Definitions

The basic idea of Web services is the use of SOAP messaging protocol to invoke software method in remote systems. This is often described by some technologists as Remote Procedure Calls (RPC) over the Internet protocols (e.g., HTTP). A SOAP message consists of an "Envelope", an optional "Header", and a mandatory "Body". The SOAP "Body" carries application-specific contents including the method name and the serialized values of the methods' input or output parameters (Scribner and Stiver, 2002). Parameters of a Web services method can be a simple value or a compound value (structure or array). Serializing a Web services message in (pure text) XML format allows the SOAP XML to pass through Internet firewall.

The Web services can be considered as a set of callable interfaces to software programs or components, regardless of their implementations. They can be invoked remotely via SOAP messaging. Therefore, these programs can provide services to other applications using Internet protocols. W3C's Web Services Architecture Working Group refers to the services provided by those programs as Web services (W3C, 2002). A detailed definition of Web services by the W3C Web Service Architecture Group stated that (Austin et al., 2002):

> **"***A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols*."

Additional standards, WSDL and UDDI, were developed to support the description and discovery aspect of the Web services. WSDL is the equivalence of IDL (Interface Definition Language) in CORBA. A WSDL file contains service definitions for distributed systems to support the automatic creation of client-side stubs or proxies, and the binding to the Web services (Christensen, et al., 2001). WSDL is specified in XML format. It describes the interfaces to a Web services implementation in terms of format of the messages, binding of the abstract messages to a concrete protocol, and address of the endpoint. It is a "take-it-or-leave-it" technical contract offered by a Web services provider to Web services consumers.

A Web services developer usually uses a software tool to generate the WSDL file from the source code of a Web services program in which some public methods are marked as Web services operations to be accessed by Web services consumers. A Web services consumer needs to generate a Web services proxy (client) to handle the encoding and messaging of an actual Web services call. A Web services generation software tool that uses a WSDL file as the input can generate the Web services proxy. Once the proxy is generated and referenced, the client application can use the Web services program via a set of the published Web operations as if they were local procedures or objects. The Web services dient proxy will handle the serialization and de-serialization between internal data types and SOAP data representation.

UDDI is a registry standard for Web services providers to publish their Web services. It may be used by a Web services consumer to discover (search) Web services developed by Web services providers. The role of UDDI as a service registry along with its relationships with service consumers and providers in the context of service-oriented architecture is depicted in Figure 1.

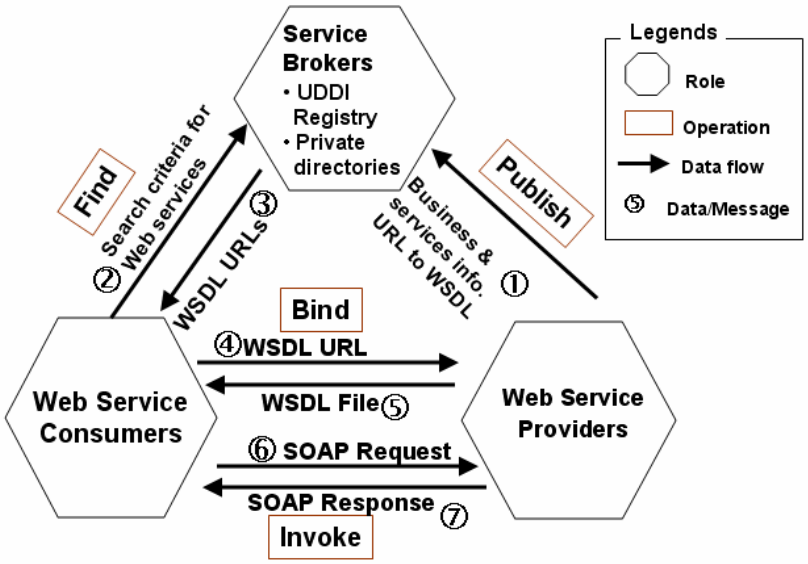

Figure 1. Web Services-Based Architecture

UDDI can store company information, services provided by a company, and the specific technical information for binding with a specific service. The technical binding information for using a Web service will be the URL reference to the WSDL file of the Web service. The structure of UDDI repository is defined in XML Schemas containing four entity types (highlighted in boldface in the following descriptions): (1) **Business entity** contains information about a company; (2) **Business services** are provided by a business entity; (3) **Binding templates** implement business services; and (4) "**tModels**" contain references to technical specifications for services.

The services registered in a UDDI repository are not limited to Web Services. However, when registering Web services, it is recommended to list the WSDL file of a Web service at the <overviewURL> entry of a tModel (Curbera et al., 2001). There are several public UDDI-based directories and they are referred to as "UDDI Business Registry". The access to the UDDI is via Web services, such that its functionality can be embedded in an application. Browser-based tools have also been developed to allow users to submit their Web services entries or search for Web services.

From a system architecture viewpoint, UDDI is a critical component in the service-oriented architecture to support dynamic usages of Web services; i.e., a Web services consumer can search for and bind to a Web service at runtime. However, the dynamic searching and binding of a Web service rarely happens today for the following reasons:

- Currently public UDDI Business Registries contain many Web services that are developed as experiments and their quality and reliability are doubtful.
- The search of a Web service in the registry can be a hit and miss because UDDI does not provide domain-specific classification information to assist the search process.
- There is no active effort of building Web services standards for specific domains or industries. Many industry specific standards are related to XML-based document standards. The lack of such domain specific Web services standards makes it very difficult to write a Web services that can be consumed in a dynamic plug-and-play fashion.

Companies may use public UDDI directories to register their businesses and services, or set up private UDDI systems for internal integration projects. Currently most entries at public UDDI are still at an experimental stage. Many traditional B2B marketplace operators may further refine UDDI standards to set up vertical market registries for Web services. We envision that companies may set up private UDDI-based or non-UDDI-based Web services registries to support their own Web services-based applications. Industry-specific registry may be established to store domain-specific Web services. Classification schemes as well as standards on various Web services interfaces that are common in various industries need to be developed to support domain-specific service registries.

The Service-Oriented Architecture (SOA) depicted in Figure 1 is a model for building loosely coupled software services in distributed environments (Serviceoriented.org, 2003). Web Services are considered an instance of an SOA. The Web services network is an application level network involving a number of participants: service

providers, service consumers, and service registry operators. Further research is needed to identify additional standards, enabling technologies, and business models for achieving Web services-centric application development to deliver software as service (Hagel, 2002; Sprott et al., 2003).

2.2 Service-Oriented Standard Stacks

When SOAP was first developed, it was intended as a simple messaging protocol to provide remote procedure calls over the HTTP protocols. However, Web services have been gradually used to support critical business applications; therefore, additional standards, such as security and composition, have been developed or should be developed to support different aspects of Web services-based applications. A Web services standards or technologies stack (Booth et al., 2003; Gottschalk et al., 2002) is shown in Figure 2 to illustrate the relationships and dependencies among various Web services standards. It can be used by standards organizations in guiding the standards development efforts and used by IT user organizations to assess deployment strategies for Web services technologies.

The bottom layer of the stack is the basic communication protocol layer for Web services including TCP/IP, HTTP, Simple Mail Transfer Protocol (SMTP), etc. XML 1.0 Specification (W3C, 2000) and XML Schema (W3C, 2001) are the definition languages used to define all the other Web services standards except the communication protocols. The basic messaging protocol is SOAP. There are several efforts, such as WS-ReliableMessaging (Ferris and Langworthy, 2003), to enhance the functionality of SOAP via processing information placed in the SOAP header. The grand vision of Web services-oriented architecture is that Web services can be composed and invoked dynamically to support business processes within and across enterprises. Hence, the highest layer of the Web services standards stack, the Service Composition layer, has emerged. This layer consists of standards that specify how individual Web services can be composed to support business processes. A number of new standards have been introduced to address this Web services composition issue, including BPEL4WS (Business Process Execution Language for Web Services), WSCI (Web Services Choreography Interface), and BPML (Business Process Modeling Language) (Andrews et al., 2003; Arkin et al., 2003; Arkin, 2002).

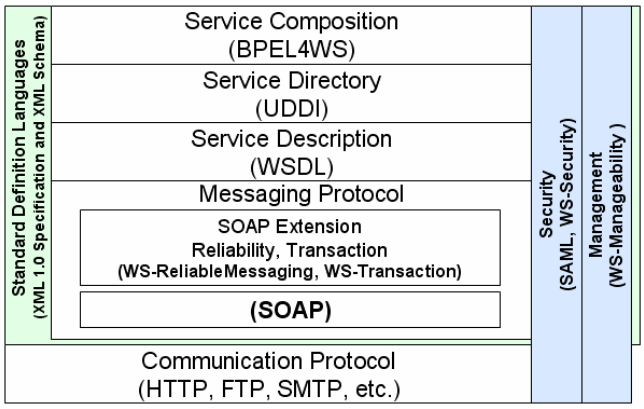

Figure 2. Web Services Standards Stack

BPEL4WS is a specification that enables a business process to be performed using a number of Web services, possibly services provided by other companies. As a joint effort by BEA Systems, IBM, and Microsoft, BPEL4WS combines and replaces IBM's Web Services Flow Language (WSFL) and Microsoft's XLANG specification. It describes executable business processes that rely on the import and export of Web services, and provides a formal standard for specifying business processes and business interaction protocols. It aims to specify the mutually visible message exchange behavior of each of the parties (e.g., trading partner) involved in the process, without revealing their internal behavior. BPEL4WS is supported by commercial products, such as Collaxa and BizTalk 2004. The advancement in Web services composition will have a major impact on research and practice in business processes and workflow applications (Leymann et al., 2002).

Some of the major concerns among companies considering adoption of Web services are security, scalability, reliability, and manageability issues (Orchard, 2002). Existing standards can be deployed or new standards need to be developed to address these issues at various layers of the Web services architecture. For example, WS-Security is one of the emerging "standards" proposed by IBM, Microsoft, and VeriSign to enhance SOAP messaging to provide Web services security through message integrity, message confidentiality, and single message authentication (Atkinson et al., 2002).

Web services do not have user interface elements. They cannot be accessed by end users directly and have to be consumed by programs. Therefore, the awareness of Web services by application users is very limited. New standards, such as Web Services for Remote Portals (WSRP), are visual Web services centric components that can be plugged-and-played with portals or other intermediary Web applications. WSRP supports the aggregation of content or applications from different sources into a portal environment without programming (Luis Diaz et al., 2002). The development of end user-oriented Web services is still rare.

There is a stream of research in Semantic Web that enables greater access to services on the Web (Berners-Lee et al., 2001; McIlraith et al., 2001). The purposes for the development of ontology of services, DAML-S (The SAML Services Coalition, 2003) under the DARPA Agent Markup Language (DAML) program, are closely related to various Web services standards as indicated in Table 1. The Semantic Web research community is mainly grounded in artificial intelligence discipline while Web services standards are developed mainly by practitioners and sponsored by software vendors such as IBM and Microsoft. There are very limited interactions between Web services standards and Semantic Web research. Adding ontology support to UDDI to facilitate Web services search is a promising direction for future research.

Table 1. Comparison of Web Services and Semantic Webs

Purposes	**Web Services Standards**	**Semantic Webs**
Advertising and discovering services	UDDI	SERVICEPROFILE: service *profile*
A detailed description of a service's operation	BPEL4WS	SERVICEMODEL: the *process model*
How to interoperate with a service, via messages	WSDL and SOAP	SERVICEGROUNDING: the *grounding*

3. Web Services Adoption

With evolving standards and improving infrastructure, Web services are currently in the early adoption phase. While most organizations are currently experimenting with Web services for application integration, some have already developed innovative implementations. So far, their deployments have been mainly found in two areas: (1) EAI – enterprise application integration that integrates disparate components of an enterprise wide system; and (2) B2B – business partner integration and secure exchange of data with dedicated partners without using a proprietary approach.

Web services, like the Internet, is less about technology but more about creating more efficient ways to do business and identifying new business opportunities. Web services will lead to major changes to business processes. Such changes will require IT organizations to evaluate their system architectures and determine how they will deliver these new business services. The experiences of early Web services adopters reflect the fundamental shift towards service-oriented architecture. Web services technology has been adopted as a mechanism for a wide variety of uses within the enterprise. The technology can be exploited as a remote-procedure-call mechanism, a component interface mechanism, an EAI-style integration mechanism, a means to standardize portal technologies, and an approach to linking businesses over the Internet.

The adoption process of Web services can be discussed in several stages. The initial Web services utilization is driven by a need to remove the binding problem on interfaces. Web services (specifically SOAP, UDDI, and WSDL) enable any compliant application to invoke any interface function. This is a unique proposition that eliminates various technology concerns for users. If any technology has the ability to invoke a Web service and expose interfaces as services, it becomes quite simple to use the technology as a means to hide platform differences.

The second stage that most organizations will undertake in their adoption of Web services will be the enabling of existing applications. Application integration requirements will foster the use of Web services interfaces on back-office and legacy applications, and many existing browser-based applications will need to be enhanced to provide information in a structured form.

The third stage of the Web services adoption is the creation of generally available business services like business components that use the Web services' technical infrastructure. These will be collections of Web services interfaces that are available to the organization. These services will be automated business functions that the organization can delegate. These services will have programmatic interfaces to execute the primary transactions and functions of the services. However, they will also likely have various services that are presented as browser-based applications as well.

Finally, the last stage of Web services development will arrive as the number and types of Web services increase in the organization. Early on, when there are few Web services, the technology that provides those services

will be derived from existing application capability, or add-on solutions. As the number of services grows, the need for a central point to administer these services and to create common solutions to various technical characteristics will increase.

3.1 Drivers and Motivations

Web services represent a profound change in the way applications are developed and can derive productivity and efficiency gains throughout organizations and across industries. Rather than developing static applications, Web services will allow companies to develop functions that can be called on demand and used on their own or in conjunction with other Web services to provide a business service. In addition, when business conditions change, business professionals can adapt the Web services application to new circumstances quickly and easily. By encapsulating business relevant functionality, Web services make it possible to rapidly create and modify applications that could never have been created through a traditional IT development model.

Today's IT imperative is to reduce technical complexity so as to increase business flexibility, and Web services provide the first widely accepted, standards-based framework for the new agile enterprise. Open standards and interoperability are major drivers of Web services' acceptance. Such technical benefits as flexibility, visibility, and leverage of Web services can also enable bottom-line returns. By adopting Web services, businesses can realize lower costs, application sharing, flexibility, streamlined business processes, additional new revenue streams, and new business models.

The business implications of Web services are significant and there are many opportunities that these new standards present for reducing costs and creating new business opportunities. Businesses are motivated to adopt Web services for several reasons: (1) lower development costs - through the use of standard interfaces (XML/SOAP) integration is simplified, resulting in both time and cost savings; (2) application reuse - each application can become a reusable object allowing a business to leverage their existing applications and shorten the total development cycle; (3) flexibility - using Web services it becomes easier to incorporate changes to support new product introduction; (4) streamlined processes - Web services can help improve business processes by eliminating manual steps through the use of standard application interface; and (5) smooth integration - introducing Web services-enabled applications to either work with business partners in a limited manner or fully in the marketplace allows other Web services applications to find each other and interact without requiring permanent links or integration development.

3.2 Adoption Issues

However, despite all of the potential benefits of Web services, a number of impediments still hinder their widespread adoption. Web services have to overcome these obstacles before its full-scale adoption becomes a reality. Among them, security, availability, reliability, and performance are the critical issues that have to be addressed first.

Security is oftentimes cited as the most important concern in Web services adoption. When Web services introduce enterprises to a new level of interoperability and openness, new security challenges are arising. This is due to several factors: the human-readable nature of the SOAP payload, use of the relatively open HTTP port 80, and the Web services philosophy that promotes interoperability across the public Internet.

Furthermore, because Web services involve passing information among multiple vendors, they also pose particular challenges to identity management, authorization, and authentication. These issues are particularly problematic for Web services because applications are connecting to multiple other applications across corporate boundaries. As such, when Web services transactions span multiple applications and organizations, it is difficult to trace and respond to an attack across different Web servers, applications, and lines of business.

Currently anyone who wants to implement highly secure and routable Web services must use their own proprietary extensions, manually modify the WSDL generated by their toolkits, and customize the SOAP messages at a very low level. Just as with standard Web-based traffic, authentication, access control, encryption and data integrity play an important role in providing basic levels of security for communication with Web services. Hopefully, standards efforts such as WS-I (Web services Interoperability) and WS-Security can help address these authentication and authorization issues. WS-Security also provides a general-purpose mechanism for associating security tokens with messages.

The Web in its current state can be unreliable sometimes, and applications that rely on several Web services are susceptible to the failure of any one of them. When external vendors are the Web services providers, the consuming company has no control over the server load or reliability. Moreover, as multiple applications may rely on one Web service, any change to that service can cause several applications to fail at once and result in a propagation of errors. As such, higher reliability for the Internet infrastructure has to be improved in the future for widespread Web services adoption.

The next concern is the performance of Web services. Since XML is text-based and entails more data to be transmitted and processed, Web services run relatively slower over HTTP, which in itself is not a high-performance

protocol either. Adding a security protocol like Secure Socket Layer (SSL) would slow performance even more. Therefore, Web services currently are not considered as an appropriate framework for mission-critical applications.

In addition to these technical challenges, careful diligence is required to identify how Web services can run business more efficiently and create new opportunities. Managing the costs and complexity of change is the biggest obstacle to increasing business flexibility. Monitoring, control, and billing of costs are among the primary business requirements for adopting Web services.

Trust has also become another managerial issue in adopting Web services. It's not only a matter of how well a company can trust the security of the Web services, but also of how well customers, partners, and IT managers can trust Web services to perform as advertised. Companies that are undertaking Web services initiatives and building service-oriented architectures are doing so with extreme caution. One of the two papers accepted for this issue specifically examines this trust issue from the perspective of customers. In their study, Luarn and Lin (2003) find trust, customer satisfaction, perceived value and commitment all influence customer loyalty, either directly or indirectly. They offer some important implications from their results for IT practitioners and future IS research.

3.3 An "Inside-out Incremental" Strategy for Adoption

As the standards and technologies for Web services keep evolving, organizations will need a roadmap that can guide them to adopt Web services in a structured manner to deliver its business benefits without unnecessary risks. An "inside-out" strategy can minimize risks and ensure that potential pitfalls are avoided.

Web services should be implemented incrementally, first within an organization and then expanding outward as standards and technologies mature. As with many new technologies, organizations will first deploy Web services internally to improve operational efficiencies and gain a unified view of complex business processes. One of the advantages of starting internally is that it is relatively easy to ensure that the chosen technologies and standards are available and compatible. Familiarity with Web services inside the organization is clearly desirable before exposing them to trading partners.

Starting internally, Web services also provide a simple and powerful way of making processes and data assets of a business accessible and reusable throughout that organization. Because disconnected proprietary systems can be accessed through standard Web services interfaces, it is relatively easy to create applications that bring together data from multiple, possibly remote locations. Similarly, new functionality can be exposed using standard SOAP interfaces to make it available across the organization. The standards that surround Web services allow companies to leverage the power of reuse like never before.

After Web services are successfully utilized internally, organizations can extend these services to customers, partners, and suppliers with which they have already negotiated agreements. Organizations now migrate existing applications to a Web services platform to automate the processes that implement an ongoing contract or supply chain relationship. During this phase, transaction protocols and security standards will become critical to the success of Web services-enabled applications.

As the breadth of companies deploying Web services on commonly used standards grows, so will the potential for more sophisticated interactions. During this phase, organizations could use Web services to facilitate the negotiation of business agreements within a set group of partners, suppliers, and customers. This phase is far more dynamic than the earlier phases; therefore, some business logic may still required human intervention.

The last part of the strategy involves improving on previous phases by enabling not only B2B process execution and negotiation with existing partners, but also the dynamic search for and identification of new business partners. Once potential partners are identified, Web services-enabled applications will negotiate the terms of the operating procedures between organizations that may never have worked together before, execute the resulting agreements, and even enable billing and payment for services and products. By allowing organizations to easily participate in unbounded communities of customers, partners, and suppliers, Web services will pave the way for unprecedented B2B productivity. Achieving this level of services will require widespread agreement on a wide range of standards, as well as global directories that enable anonymous but reputable and legitimate participants to easily and reliably locate one another for collaboration.

4. Web Services Impact

Web services are self-contained and self-describing modular applications that can be published, located, and invoked across the Web. This nature of Web services could facilitate a new trend in B2C e-commerce towards micro-commodity and micro-consumption; that is, products and services can be designed and delivered in finer granularity and customers can pick and choose only the desired ones. On the one hand, businesses might want to break the existing applications in much smaller pieces and make it available as Web services. On the other hand, new applications under development should be designed as granular Web services. For consumers, Web services can provide greater flexibility and refine cohesion. Theoretically, customers can buy different pieces from multiple

product or service providers, put these pieces together the way they want, and do not worry about the communication and interfacing problem. That is, customers can choose "best of breed" modules and put them together in any way desired. At the same time, under Web services, the cost of switching product or service providers will be reduced to minimal.

Briefly defined, business-to-business (B2B) e-commerce refers to e-commerce activities that are conducted exclusively between business entities. Electronic data Interchange (EDI) is an example of the early form of B2B systems. Maruyama (2002) suggested that the primary motivation of Web services comes from a desire to dynamically integrate business processes. This desire meets with the now ubiquitous Internet technologies and the modern software engineering concepts (e.g., software components and distributed objects) and produces the standard set of protocols. Existing EDI and XML-based B2B systems improve the efficiency of communication between business parties, but there are no fundamental changes in the business workflows. Now because business processes that are implemented as enterprise application are programmatically accessible from the Internet, it is possible to dynamically integrate these businesses to create new business workflows. As Web services have been increasingly used to build and integrate applications in a standardized way, the barriers to business-to-business integration will be greatly reduced. As a result, Web services will advance traditional B2B e-commerce to *dynamic e-business* by dynamically connecting systems, business partners, and customers cost-effectively through the Web.

4.1 Impact on Business Applications

There has been an ongoing effort on enterprise application integration (EAI) for many companies to improve interoperability as systems can speak the same language or use the same protocols. Accenture (2003a) described that EAI is a vital strategic enabler that provides a common integration framework, enabling companies to integrate program-to-program business processes, workflows, and data across disparate applications. EAI increases a company's ability to respond, adapt and collaborate with innovations quickly and relatively inexpensively. However, EAI might be too focused on the current implementation and result in locking companies into a proprietary EAI product.

Web services promote easy integration within and outside a company's boundary. Web services also increase interoperability – especially as they are added to middleware and integration products. Businesses should view the Web services evolution as a complimentary integration solution to EAI and beyond. Corporate leaders should understand and review both technologies for implementation into overall integration architecture. EAI will inexpensively support the addition of Web services into the architecture. Kreger (2003) described that Web services technologies are being developed as the foundation of a new generation of B2B and EAI architectures, as well as important parts of such "on demand" components as grid, ubiquitous, and autonomic computing.

Internet expands the geographic possibilities, making it feasible for services to be solicited and delivered from almost anywhere. Web services add another dimension where applications rather than individuals find and engage services. The interaction no longer needs to be local and no longer even needs to be performed by a human – automatic computing. For Grid computing, Gannon (2002) defined a computational Grid as a set of resources, such as computers, networks, on-line instruments, data servers or sensors that are tied together by a set of common services which allow the users of the resources to view the collection as a seamless computing/information environment. There are several obvious ways that the Web services model can be used in Grid systems. Many more of the standard Grid services can and will be given WS interface. Standards like UDDI and WSDL provide a discovery mechanism for services that is very similar to the Grid information system model, but it is applied at the level of application services instead of Grid hardware resources. WS applications can also be enhanced by adding Grid security protocol.

Accenture (2003b) depicted that the Web services architecture – with connectivity and integration at its core – will help to spur growth through the take up of a whole range of new computing forms that should fuel mobility in the short term and ubiquitous connection in the long term. That is, Web services will enable the movement from (1) application integration to (2) the integration of processes across the enterprise and then (3) ubiquitous integration as devices, people and organizations are continuously and seamlessly connected to each other.

4.2 Impact on workflows and business processes

Papazoglou (2003) described that traditional enterprise workflow and business process management systems support the definition, execution and monitoring of long running processes that coordinate the activities of multiple business applications. However, traditional workflow models rely on message-based computing methods that in nature are tightly coupled to protocols for e-business or application integration. These protocols assume a tightly linked and controllable environment, which is not the nature of the Web. As a result, internal application may not be readily configured for Web deployments. Further, when moving to the Web environment, participants in workflows won't be pre-determined and may be dynamically chosen to fulfill certain roles.

At the message exchange level, Web service technology can solve the traditional problem by replacing proprietary interfaces and data formats with a standard Web-messaging infrastructure equipped with SOAP and XML. Further, Web services promote communication standards and common protocols for loosely coupled applications and can assist dynamic workflow and business process management in the Web environment. Box (2003) stated that Web services can help automate and orchestrate business processes, and Web services can contribute to the drive for standardization of workflow and business processes across the enterprise to engender global economies of scale.

Three standards that have been developed to handle business-level coordination on workflow and business process as Web services technology advances: Business Process Execution Language for Web Services (BPEL4WS or BPEL), WS-Coordination (WS-C), and WS-Transaction (WS-T). BPEL4WS is a workflow-like definition language that describes sophisticated business processes that can orchestrate Web services. WS-C and WS-T complement BPEL to provide mechanisms for defining specific standard protocols for use by transaction processing systems, workflow systems, or other applications that wish to coordinate multiple Web services. These three specifications work together to address the business workflow issues implicated in connecting and executing a number of Web services that may run on disparate platforms across organizations involved in e-business scenarios. As a result, Web services can be viewed as Internet-based applications fulfilling a specific task or a set of tasks that can be combined with other Web services to maintain workflow or business transactions.

4.3 Impact on business practices

Web services can extend the life of their existing systems rather than rebuilding and incurring greater costs. Kreger (2003) reported that Web services deployments have shown that existing assets used within a company can readily become revenue-generating assets. Web services should be leveraged today for application integration and programming model unification. New product plans and business connections being developed should consider Web services as part of the solution. Companies should carefully examine their current status and future needs, and plan implementation of Web services realistically to achieve the company's goal. Accenture (2003a, 2003b) argued that companies can use standard Internet technologies (independent of any particular platform) offered by Web services for three types of integration – internal, external, and multi-channel. For internal implementation (application-to-application interaction), integration of heterogeneous systems can be done at lower cost and enterprise systems can be exposed as a set of reusable Web services which can be consumed by composite business applications. For external implementation (inter-enterprise connectivity), business partners will communicate transactions based on agreed standard documents for each step in a business process, significantly reducing custom business processes. For multi-channel implementation (global access by extended users over the Web), this type of integration can retool existing applications across these different channels to adapt to business innovations and extend the reach of systems across value and supply chains.

Web services can remove the technical barriers to inter-organizational processes. However, this shifts the focus to the two remaining business barriers: shared semantics and contractual obligations. The reason most Web services Web services implementations are internal today that both of these factors are within the control of the organization. Kreger (2003) pointed out that the vision of fully dynamic, *ad hoc* business partnerships is not yet viable for a number of reasons. First, the infrastructure standards outlined here must be finished, productized, and widely deployed. Second, industry standard Web services interfaces, or portTypes, must be defined for the various aspects of B2B relationships. Finally, XML languages that can describe legally binding businesses and service-level agreements must be defined and standardized. This is more than a technical challenge; it may be a cultural challenge as well because business relationships often span legal, cultural, language, and national boundaries. Dignan (2002) suggested that for Web services to work as imagined, technology hurdles must be the first challenges overcome, but businesses also will have to change the way they view software and intellectual property rights. Proponents of the Web services vision also face work in the areas of security, standards, and privacy. Aoyama et al. (2002) also identified that non-ownership of software and service outsourcing over the Internet will challenge the software industry as well as the whole community of software users.

5. Web Services and Outsourcing

IT outsourcing has been a pervasive concept and practice. Basically, IT outsourcing describes the situation where a company contract with outside vendors to fulfill the company's IT needs. Richmond et al. (1992) defined IS outsourcing as the subcontracting of some or all the information systems functions by one firm to another. Willcocks and Fitzgerald (1993) described that IS outsourcing is the commissioning of third party management of IT assets, people, and /or activities to required results. As companies started to baffle the idea of maintaining full-scale applications internally, IT outsourcing initiatives started to emerge in the 1980s. Jurison (1998) summarized many reasons for considering IT outsourcing such as cost savings through economies of scale, cash infusion, reduced

capital spending, faster applications development, improved service and quality, access to IT expertise and competence, access to new technologies, flexibility in managing IT resources, and elimination of a troublesome function.

Originated from the concept of IT outsourcing, application service providers (ASP) came in to existence in 1997. Basically, ASP is a business model of offering software as a service on a subscription bases. ASP industry Consortium (2000) defined that application service providers are service organizations that deploy, host, manage, and enhance software applications for customers at a centrally managed facility, offering application availability, performance and security. End-uses access these applications remotely using Internet or leased lines. Terdiman (2000) defined that application service providers are organizations that deliver the application functionality and associated services across a network to multiple customers using a pay-as-you-go payment model. One-to-many relationship (economies of scale) is the basic premise of an ASP striving for economies of scale by offering higher quality services at lower costs.

ASP business model has suffered from the inability to customize and integrate applications. In order to maximize the advantages of one-to-many, an ASP could try to only offer software in its pre-packaged form. However, ASP customers do demand customization and require integration, and they do not want to pay huge premiums for these services. Traditional ASPs usually absorb the costs for customization and integration in order to gain and keep customers. As a result, since early 2000, many ASPs have filed for bankruptcy including ones that were considered major players. In order for ASPs to offer their services efficiently and gain profitability, they must provide services in a standard way and deliver easy integration of provided services and customers' systems.

Web services promote integration and ensure interoperability and compatibility between applications on multiple platforms in seamless communication by using standards. Therefore, integration of the ASP-provided services with customers' systems becomes much easier with implementation of Web services. As a result, Web services allow clients to connect directly to the applications making the process more efficient. It allows ASPs to tightly integrate with their customers, creating tighter bond between both entities, and no longer have to worry about making these systems compatible with each other. The services that the ASP will be able to offer to the clients will be exceedingly better than what can be offered today. Specialization will become easier due to the platform independency and the ability for easy integration by using Web services.

Web services promote customization and flexibility by preparing applications as self-contained, self-describing, and modular applications. Therefore, ASPs can provide IT outsourcing services in finer granularity and customers can pick and assembly only necessary components; therefore, customization problem of ASPs can be eased. In order to achieve this goal, ASP must transfer existing hosted applications into the Web services model. That is, ASPs might need to break the existing application packages in much smaller pieces and make it available as Web services. For example, in 2000, Corio and USinternetworking, two early leaders of the ASP market, introduced their "one-to-several" approach. This approach is to offer customization solutions in a standardized efficient way. The reusable components are modular solutions that can be offered to customer with similar needs. According to Corio, their product, Orion which facilitates the one-to-several approach, can satisfy about 95% of request customization.

For ASP customers, Web services can provide greater flexibility and refine cohesion. Theoretically, customers can buy different pieces from multiple ASPs, put them together the way they want, and do not worry about the communication and interfacing problem. That is, customers can choose "best of breed" modules and put them together in any way desired. In addition, under Web services, the cost of switching ASPs will be reduced considerably. This is especially crucial benefit from Web services for ASPs to attract customers by alleviating their fear of instability of ASPs.

Despite the downfalls of economy and dot-com breakdown, IT outsourcing is still a valuable option for companies. Web services appear to be a potential solution for the critical problems of the ASP business model and could further enhance IT outsourcing future. ASPs must set up a strategy to enter the Web services market and increase the benefits to their customers. The underlying business model will not have to be changed, but new standards must be adopted, and a strategy must be formed in order to efficiently switch from hosting proprietary applications into providing platform-independent and Web-based applications.

6. Concluding Remarks

Web services refer to a family of technologies that can universally standardize the communication of applications in order to connect systems, business partners, and customers cost-effectively through the World Wide Web. The emerging Web services standards and technologies enable companies to provide software functions and business services over the Web to be integrated by internal business processes or with trading partners. Web services have been proclaimed as "bigger than the Internet" by many advocates. These supporters believe that Web services are "the" technology to bridge the gap between IT and business. Beyond the hype, Web services hold the promise to

handle and solve complex business problems in the foreseeable future of global competition. It generates a renewed interest and excitement in B2B electronic commerce and mobile commerce, as well as enterprise application integration (EAI). Major software vendors such as IBM, Microsoft, SAP, SUN, and Oracle are all embracing Web services standards and are releasing new products or tools that are Web services enabled. Web services will ease the constraints of time, cost, and space for discovering, negotiating, and conducting e-business transactions. As a result, Web services will change the way businesses design their applications as services, integrate with other business entities, manage business process workflows, and conduct e-business transactions. Research opportunities of Web services and e-commerce area are fruitful and important for both academics and practitioners. We wish that this introductory article can shed some light for researchers and practitioners to better understand important issues and future trends of Web services and e-business.

REFERENCES

Accenture, "Web Services and Enterprise Integration: Friends Not Foes." http://www.accenture.com/xdoc/en/services/web/insights/web_enterprise.pdf, 2003a.

Accenture, "Web Services: IT Efficiency Today … Powerful Business Solutions Tomorrow." http://www.accenture.com/xdoc/en/services/web/insights/web_survey.pdf, 2003b.

Andrews, T. et al., Business Process Execution Language for Web Services Version 1.1, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/, May 5, 2003.

Aoyama, M., et al., "Web Services Engineering: Promises and Challenges," *Proceedings of International Conference on Software Engineering (ICSE' 02)*, p. 19-25, 2002.

Arkin, A., Business Process Modeling Language, BPMI.ORG, November 13, 2002.

Arkin A., et al., Web Service Choreography Interface 1.0 Specification, http://dev2dev.bea.com/techtrack/wsci.jsp, 2003.

ASP Industry Consortium, Internet Survey, http://www.aspindustry.org/surveyresults.cfm, 2000.

Atkinson, B, etc., Web Service Security (WS-Security), Version 1.0, http://www-106.ibm.com/developerworks/library/ws-secure/, April 5, 2002

Austin, D., et al. (eds.) Web Services Architecture Requirements, W3C Working Draft 14, http://www.w3.org/TR/wsa-reqs#id2604831, November 2002.

Berners-Lee T., J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, 84(5):34-43, 2001.

Booth, D., et al. (eds.), Web Services Architecture, W3C Working Draft 8, http://www.w3.org/TR/2003/WD-ws-arch-20030808/, August 2003.

Box, B., "The Web Services Revolution," http://www.eds.com/thought/thought_leadership_web_revolution.pdf, 2003.

Chen, M. and M. J. Meixell, "Web-Services Enabled Procurement in the Extended Enterprise: An Architectural Design and Implementation," *Journal of Electronic Commerce Research*, 4(4), 2003.

Christensen, E., et al., Web Services Description Language (WSDL) 1.1, W3C Note, http://www.w3.org/TR/2001/NOTE-wsdl-20010315, March 15, 2001.

Collaxa, On BPEL4WS1.1, http://www.collaxa.com/product.bpel11.html, Sept. 2003.

Curbera, F., D., Ehnebuske, and D. Rogers, Using WSDL in a UDDI Registry 1.05, UDDI Working Draft Best Practices Document, http://www.uddi.org/pubs/wsdlbestpractices-V1.05-Open-20010625.pdf, June 25, 2001.

Dignan, L., "Report: Web Services a Decade Away," *ZDNet News*, http://zdnet.com.com/2100-1106-963182.html, 2002.

Ferris, C. and D. Langworthy, Web Services Reliable Messaging Protocol (WS-ReliableMessaging), http://msdn.microsoft.com/ws/2003/03/ws-reliablemessaging/, March 13, 2003.

Gannon, D. et al., "Programming the Grid: Distributed Software Components, P2P and Grid Web Services for Scientific Applications," *Cluster Computing, 5*, pp. 325-336, 2002.

Gisolfi, D. "Web Services Architect Part 1: An Introduction to Dynamic e-Business," http://www-106.ibm.com/developerworks/webservices/library/ws-arc1/, 2001.

Gottschalk K., S. Graham, H. Kreger, and J. Snell, "Introduction to Web Services Architecture," *IBM Systems Journal*, Volume 41, Number 2, 2002, pp. 170-177.

Hagel III, J. *Out of Box: Strategies for Achieving Profits Today and Growth through Web Services*, Boston: MA, Harvard Business School Press, 2002.

Jurison, J., *A Risk-Return Model for Information Technology Outsourcing Decision*, John Wiley & Sons Inc., New York, USA, 1998.

Kreger, Heather. "Fulfilling the Web Services Promise." *Communications of the ACM, 46*(6), pp. 29-34, 2003.

Luarn, P. and H. H. Lin, "A Customer Loyalty Model for E-Service Context," *Journal of Electronic Commerce Research*, 4(4), 2003.

Leymann, F., D. Roller, and M. T. Schmidt, "Web Services and Business Process Management," *IBM Systems Journal*, Vol. 41, No. 2, 2002.

Luis Diaz, A., et al., Specification: Web Services for Remote Portals (WSRP), Note, http://www.ibm.com/developerworks/library/ws-wsrp, January 21, 2002.

Maruyama, H., "New Trends in e-Business: From B2B to Web Services," *New Generation Computing*, *20*, pp. 125-139, 2002.

McIlraith, S., T. C. Son, , and H. Zeng, , "Semantic Web Services," *IEEE Intelligent Systems*, Special Issue on the Semantic Web, Volume 16, No. 2, pp. 46-53, March/April, 2001.

Microsoft, BizTalk 2004 Beta, http://www.microsoft.com/biztalk/beta/, July 17, 2003.

Orchard, D., Web Services Pitfalls, http://www.xml.com/pub/a/2002/02/06/webservices.html, Feb. 13, 2002.

Papazoglou, M. P., "Web Services and Business Transactions," *World Wide Web: Internet and Web Information Systems*, *6*, pp. 49-91, 2003.

Richmond, W. B., A. Seidmann, and A. B. Whinston, "Incomplete Contracting Issues in Information Systems Development Outsourcing," *Decision Support Systems*, *8*, pp. 19-22, 1992.

Scribner, K. and M. C. Stiver, *Applied SOAP: Implementing .NET XML Web Services*, SAMS Publishing, 2002.

Serviceoriented.org, Service Oriented Architecture, September, http://www.serviceoriented.org/service_oriented_architecture.html, 2003.

Sprott, D., et al., Web Service Roadmap, CBDI Forum Ltd., August 2003.

Terdiman, R., "ASPs: What are the Current Market Trends," Research report, GartnerGroup, USA, 2000.

The DAML Services Coalition, DAML-S: Semantic Markup for Web Services, http://www.daml.org/services/daml-s/0.9/daml-s.html, May 2003.

W3C, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3.org/TR/REC-xml, October 2000.

W3C, XML Schema Part 0: Primer, W3C Recommendation, http://www.w3.org/TR/xmlschema-0/, May 2, 2001.

W3C, Web Services Architecture Working Group Charter, v 1.46, http://www.w3.org/2002/01/ws-arch-charter, July 25, 2002,

Willcocks, L. and G. Fitzgerald, "Market as an Opportunity? Case Studies in Outsourcing Information Technology and Services," *Journal of Strategic Information Systems*, *2*, pp. 223-242, 1993.