Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Int. J. Human-Computer Studies 58 (2003) 259–279
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Factors affecting the adoption and diffusion of XML and Web services standards for E-business systems

Minder Chen*

Decision Sciences and MIS, MSN-5F4, School of Management, George Mason University, 22030, USA

Received 29 October 2002; accepted 25 November 2002
Paper accepted for publication by Editor, B. Gairies

Abstract

Integration is a very important issue in e-business systems integration. There are three basic mechanisms to integrate diverse applications: data interchange, application program interfaces (API), and shared repositories. Standards and standardization efforts play important roles in all three areas. XML is a standard for defining data interchange standards. Web services are a set of standards for calling remote procedures over the Internet. UDDI and other e-business registries are examples of shared repositories. This paper discusses the XML and Web Services (including UDDI) standard-related technologies in the context of e-business systems. A technology adoption life cycle model is adapted to analyse various stages of standards adoption. Organizations that tend to adopt e-business standards in each stage are identified. Based on our study of e-business standards and research of relevant literatures, several factors that affect the adoption decision of e-business standards are identified. The implications of these factors are discussed. Two case studies of the implementation of e-business standards are presented. Further studies of how standards may affect the success of e-business strategies and how companies should evaluate and implement e-business standards are proposed.

1. The importance of standards in e-business systems

The integration needs of e-business initiatives are driven by enterprise application software implementations and e-commerce initiatives. Enterprise resource planning

*Tel.: +1-703-993-1788.
E-mail address: mchen@gmu.edu (M. Chen).

1071-5819/03/$ - see front matter
doi:10.1016/S1071-5819(02)00140-4

(ERP) packages are the first enterprise application software package implemented by many companies to integrate their backend operations in productions, finance, and human resources. The adoption drivers for ERP include the need to integrate internal business processes triggered by business process reengineering projects in the mid-1990s, and the replacement of some legacy systems to avoid Y2 K problems in the late 1990s. Some companies have also implemented supply chain management (SCM) software to complement existing ERP packages to integrate enterprise resource planning functions with suppliers' inventory and demand forecast systems. Customer relationship management (CRM) software is the latest addition to e-business packages to create more integrated customer information systems. These include functions such as marketing, sales, and customer services. More recently, e-business application packages have been web-enabled to extend their functionalities beyond organizational boundaries connecting to business partners in B2B e-commerce systems.

The e-commerce initiatives in many companies started with business-to-consumer (B2C) e-commerce in the mid-1990s, and then moved towards business-to-business (B2B) e-commerce in the late 1990s. An IBM study (2001) shows that launching and maintaining a web site consumes only 10% costs of the IT budget, while 90% of the costs are related to integration, software, and staffing (IBM, 2001). An e-commerce site needs to be fully integrated with its backend systems internally, and with suppliers' information systems externally, to provide more timely information regarding product price and availability to its customers. E-commerce projects have created urgent needs to integrate fragmented internal backend applications to support the customer facing B2C e-commerce applications.

Most ERP, SCM, and CRM projects, as well as e-commerce initiatives, involve consolidation of data from various sources and integration with legacy systems. One study, from IBM, found that 85% of all e-business infrastructures are some form of evolution, mutation, or integration of existing systems (IBM, 2001). The integration of various IT applications and components inside and outside the enterprise boundary is costly and time consuming due to the heterogeneity of the computing environments involved.

Using standards to facilitate the communications between two different systems is the most promising approach to facilitating e-business integration. There are several compelling reasons for adopting e-standards. First, standards prevent IT users from locking in with any specific vendors; therefore, IT users have more bargaining power in acquiring IT products. Second, applications or tools based on standards can be integrated easily. This allows IT implementers to take the "best of breed" approach to build their system architecture in achieving their business and technical objectives. Third, there are more IT personnel resources available when widely adopted standards are used. This reduces the cost of recruiting and training IT personnel.

Many standards have been developed. However, there are few research studies on standards adoption and diffusion. In this paper, the author will first review two essential standards, XML and Web services, in the context of e-business integration. A technology adoption life cycle model is used to examine issues or organizations involved in various phases of standards adoption. Factors affecting adoption

decisions based on literature research are discussed. Two in-depth case studies of organizations that adopted XML and Web services are presented. The paper is concluded with discussions on future research issues and directions.

2. XML and E-business

Potential applications of e-business standards may be applied to the following areas: enterprise application integration (EAI), enterprise portals, business partner integration, content/document management, business process management and integration, and B2B exchanges. There are many standards related to e-business that have been developed. The two most essential e-business standards are XML and Web services, which address data integration and component integration issues. We will discuss these standards in the context of e-business in this section and the next section.

XML stands for "eXtensible Markup Language". It is extensible because it is not a fixed format like HTML. It is designed as a simplified Standard Generalized Markup Language (SGML) on the World Wide Web. SGML is a very complicated standard mainly used for desktop publishing software as a data interchange standard for technical documents. XML should be considered as a suite of standards. There are three levels of abstraction related to XML: (1) XML definition language, (2) XML DTD or XSD file, and (3) XML document. This often confuses people who try to understand what XML is all about. Additional supporting standards such as XSL, XML DOM or Simple API for XML (SAX) are required to use XML in applications. Fig. 1 depicts the relationships of XML-related standards and the context in which they are used.

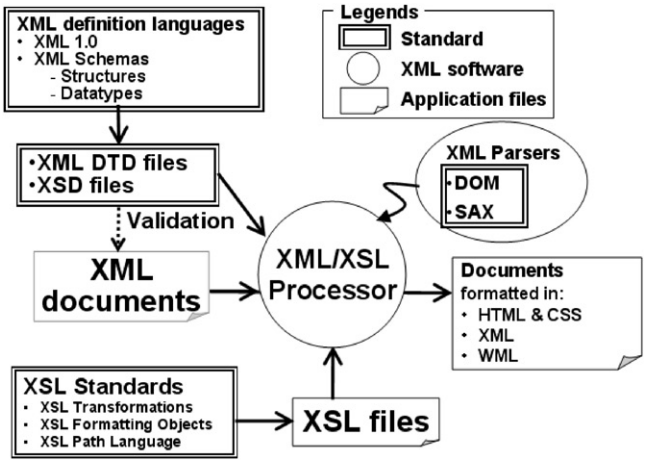

Fig. 1. XML related standards and usage scenarios.

2.1. XML definition languages

An XML definition language is a meta-language to formally define new markup languages. Currently, there are two "standards" that have been in widely used as XML definition languages. The first standard is XML 1.0 W3C Recommendation (W3C, 2001). XML 1.0 Recommendation defines a set of production rules or grammars that developers can use to define new markup languages. The basic constructs of an XML definition language are Elements (markup tags) and Attributes. A markup language definition contains elements that are grouped into a hierarchical structure by defining parent–child relationships between elements. An XML tree has only a single root element, known as the document root. Attributes can be attached to elements to further describe these elements. XML uses Unicode so that it can be encoded in different national languages.

The second XML definition language standard is XML Schema. XML Schema has two parts: XML Schema Structures and XML Schema Datatypes. XML Schema is more precise and rigorous compared to XML 1.0 Recommendation. Therefore, XML Schema is gaining momentum. It has been used for defining several critical Web services standards. XML 1.0 Recommendation and Schema can be used to define XML DTD files and XML Schema Definition (XSD) files, respectively. DTD and XSD files are basically definitions of markup languages. They define the structures and constraints of specific documents used in certain industries or applications.

An XML document is an instance of DTD or XSD that consists of markup tags and attributes defined in DTD or XSD. XML documents represent actual application data or documents. DTD and XSD files can be used to validate XML documents to ensure the data integrity. Data validation (i.e., error checking) is a critical function for data interchange between e-business applications. For example, if someone quotes you a price on a product but forgets to mention the currency used, your application can alert you the situation. Since the meanings of the tags in an XML document must be agreed upon by the applications that exchange data, the developers of these applications (e.g., suppliers, customers, departments, and application programmers) should work together to determine what existing DTD or XSD standards to use or design new ones. The Organization for the Advancement of Structured Information Standards (OASIS), a nonprofit consortium that drives the development of e-business standards, operates a community clearinghouse (i.e., XML.ORG) for XML application schemas (DTD or XSD) (OASIS, 2002). Many industry groups are working on industry-wide XML application schemas as standards for e-business data and document exchanges.

2.2. eXtensible style language (XSL) and XML parsers

There are three aspects of a document: Structure, Content, and Display. The structure of a document is defined by XML DTD or XSD files. The content is captured in actual XML documents. There are two approaches to handle the display

aspect of XML documents: XSL, and XML parsers. The relationships among these standards are illustrated in Fig. 1.

eXtensible Stylesheet Language (XSL) 1.0 became a W3C Recommendation on October 2001 (W3C, 2001). XSL consists of two parts: a language for transforming XML documents (i.e., XSLT: XSL Transformation), and an XML vocabulary for specifying formatting semantics (i.e., XML Formatting Objects). XSL uses templates such as 〈**xsl:templates match**="…"〉 to define how to transform XML elements. The **match** attribute is used to associate a template with an XML element to which the template applies. The template may contain additional XSL formatting information, including HTML tags, string literals, and CSS for formatting. XML Path Language (XPath) is another standard that is a language for addressing parts of an XML document, designed to be used by XSLT (W3C, 1999).

Server-side or client-side XML/XSL processors are programs that use XSL files to convert XML documents to HTML document for a web browser, to Wireless Markup Language (WML) for WAP-based browsers, and to another XML document for B2B applications. Server-side scripting programs can *dynamically* apply different XSL files to an XML document based on the requesting software agent's type.

Another approach to displaying or processing XML document is the use of XML parsers. XML parsers are software libraries that implement a set of APIs conforming to DOM or SAX standards, which are usually used by application programs to provide greater flexibility in integrating XML between applications. The Document Object Model (DOM) is a standard set of APIs that allow programs to dynamically access and update the contents and structures of documents (W3C, 2000). DOM requires the parsing of an XML document into an in-memory DOM tree. Another XML parsing standard, *Simple API for XML* (SAX), was the first widely adopted streaming API for XML in Java and is now supported in many languages (SAX Project, 2002). SAX is the de facto standard (not a W3C standard). SAX allows you to process an XML document in a forward only and read only fashion. It is more efficient, but not as powerful as DOM.

The XML standards, XML 1.0 and Schema, XSL, and DOM, were developed by W3C. Some DTD or XSD standards were developed by industry specific standards organizations with the involvement of business area domain experts and developers who are familiar with XML definition languages. Application programmers may use XML parsers for application to application or B2B integration. XSL is more declarative in nature compared to XML parsers; therefore, web user interface developers are more likely to use it for XML web content delivery and presentation. Proper role specialization in using XML may speed up the adoption of XML standards.

2.3. *Applications of XML*

XML is relatively simple. It is also extensible and platform independent. These factors may contribute to the broad and rapid adoption of XML in many industries,

tools, and application. The use of XML in various applications and rationales for its adoption in various applications are discussed in this section.

- *Data interchange*: An XML document is in pure text format, not binary format. This will allow the document to be sent and read easily across the Internet, passing through firewalls. XML can be compressed when bandwidth is a concern. XML has been used for data transfer of Enterprise Application Integration (EAI) within a company. It can be used between trading partners in B2B supply chain management applications. It is gradually replacing EDI as a new standard for e-commerce data interchange. XML is Unicode-compliant, so it supports internationalization. This is an important factor contributing to the adoption of XML for applications invloving global trade.
- *Disconnected web and mobile applications*: XML documents can be delivered and saved (i.e., cached) at client side to allow local filtering, viewing, computation, and manipulation of data in a disconnected scenario to facilitate Web and mobile computing. Locally stored XML files can be uploaded to the server and merged with centralized managed data. This approach also enhances the scalability of a distributed application.
- *Content management*: XML focuses on structure and content of data instead of presentation. By using an XSL or XML parser, we can present XML documents to multiple displaying devices. Therefore, XML has become a standard for content management systems. Customized tags used in XML documents may facilitate more meaningful document searches.
- *Specification of configuration files*: Many software vendors use XML to represent their configuration files used by their development tools, including J2EE and Microsoft .NET tools.

XML has also been used to define other e-business standards. For example, many Web services standards are defined in XML. The details of Web services are discussed in the following section.

3. WEB services standards and e-business

HTML can be used to build static and content-oriented web sites. Server-side scripting languages such as ASP, ColdFusion, or Java Server Pages, can create dynamically generated contents to support interactive e-commerce and virtual community sites. However, contents or functionalities of a web site developed by these technologies cannot be easily *embedded* in other applications that do not use the same technology. Web services allow companies to develop *programmable services* over the Web protocols to support program to program interactions and integration based on a set of standards.

Gartner, an IT research firm, defines Web services as "loosely coupled software components delivered over Internet-standard technologies" (Vecchio, 2001). "Loosely coupled" implies that the Web services are independent of any

programming languages, platforms, and object models. Using the ubiquitous and low-cost Internet, Web services can easily provide software functions over the internal networks and the public Internet to other programs. Business functions are readily shareable at a smaller granularity level. Companies can also link and configure Web services into a flexible process flow to support dynamic e-business environments (Foody, 2002). Similar to XML, Web services consist of a set of standards supported by many development tools. These standards and how they can be used in e-business applications are discussed in this section.

3.1. WEB services standards stack

Web services consist of a set of standards that can be organized as layers of standards as depicted in Fig. 2 (Gottschalk et al., 2002). The bottom of the stack is the basic Internet and transport layer protocols for Web services including TCP/IP, HTTP, Simple Mail Transfer Protocol (SMTP), etc. The message and data format layer use XML Schema definition language.

The actual messaging layer standard is Simple Object Access Protocol (SOAP). SOAP is a standard that defines XML formatted messages between two applications over the Internet protocols. 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 service method can be a simple value or a compound value (structure or array). Serializing a Web service 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 components, regardless of their implementations. They can be invoked remotely via SOAP messaging.

Web Service Description Language (WSDL) is the Service Layer standard. WSDL is still just a "*Note*" published by W3C (Christensen et al., 2001). WSDL is the

Layer of services	*Standards (Public and Private)*	Complementary Standards		
Business Process and/or Service Workflow definitions	Microsoft's XLANG, IBM's WSFL, ebXML's Trading Partner Agreement, RosettaNet's Partner Interface Process (PIP)	Security	Reliability (QoS)	Management
Service Publication and Discovery	UDDI, ebXML Registries			
Service Description	WSDL in XML format			
Service Messaging	SOAP in XML format			
Message and Data Format	XML Schema			
Network Transport	HTTP, SMTP, FTP, etc.			

Fig. 2. Web services architecture and standards stack.

equivalence of Interface Definition Language (IDL) 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.

Service Publication and Discovery Layer standards include UDDI and ebXML Registries, etc. The Universal Description, Discovery and Integration (UDDI) specification is developed by UDDI.ORG. UDDI defines a standard way to publish and discover information about Web services. The UDDI repository structure is defined in XML Schemas containing four element types. *Business entity* contains information about a company. *Business services* are provided by a business entity. *Binding templates* implement business services. "tModels" have references to technical specifications for services. The services registered in a UDDI repository are not limited to Web Services. However, when you register Web services, it is recommended that you should list the WSDL file of a Web service at the <overviewURL> entry of a tModel (Curbera et al., 2001)

As a core component of the UDDI project, the UDDI Business Registry allows businesses to programmatically locate information about Web services exposed by other organizations. The programmatic access interface to UDDI to register and search Web services is defined as a set of SOAP APIs. The Web services standards stack is developed by using existing standards to define new standards. This approach further facilitates the development and adoption of standards. 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.

The highest layer, the Business Process and Service Workflow layer, currently has many competing "standards" for combining Web services to implement a business process or a workflow application. Most efforts at this layer are still proprietary standards controlled by IT vendors, such as Microsoft's XLANG (used in Microsoft's BizTalk) and IBM's Web Services Flow Language. Standards also belonging to this layer are ebXML's Business Process Specification Schema (BPSS) [Ebxml.org, 2002] and RosettaNet's Partner Interface Process (PIP) (Rosettanet.org, 2002). BPSS and PIP focus on B2B integration of business processes across organizational boundaries.

Some of the major concerns among companies considering adoption of Web services are security, scalability, reliability, and manageability issues (Orchard, 2002). Existing standards and technologies can be used to support Web services, and new standards need to be developed to address these issues at various layers of the Web service 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). The WS-I initiative has been launched to ensure that implementations of Web services among tools vendors are consistent, so that we can achieve standards-based integration and interoperability across platforms, applications, and programming languages (WS-I, 2002).

3.2. Web service deployment

There are three roles in a comprehensive Web service model as described in Fig. 3: Service Registries, Service Providers, and Service Consumers. Understanding these roles will help us understand how Web services may be deployed. Here is a complete usage scenario involving all three roles:

1. A Web Service Provider *publishes* its business information, Web services information, and URLs to WSDL files in a Service Registry. Service Registries can be implemented as either a private UDDI for EAI or a public UDDI for B2B integration.
2. A Web Service Consumer sends search criteria of Web services to an UDDI to *find* appropriate Web services. The Web Services Consumer can also search a business entity and then drill down to find out the Web services provided by the business.
3. A Service Registry returns the matched URLs of WSDL files requested by the Web Service Consumer. This search process is often conducted by developers via a browser-based user interface offered by the Web Registry (e.g., UDDI) service.
4. A Web Service Consumer sends a request to a Web Services Provider to obtain a WSDL file. A WSDL file that defines a Web service is usually generated by Web services development tools. If you know a Web service's WSDL file's URL directly, you can use a Web service by skipping step 1–3.
5. The requested WSDL file is returned by the Web Service Provider to the Web Service Consumer. The Web Service Consumer then uses the WSDL file to generate the necessary Web service client-side stub to bind to the web service in order to use the functions provided by the Web service defined in the WSDL file.
6. The Web Service Consumer (i.e., a bound web service client application) sends a SOAP request to the Web Service Provider to *invoke* a function provided by the Web service.

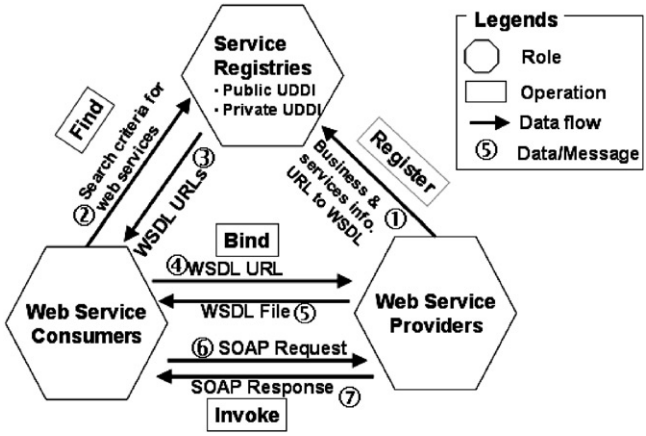

Fig. 3. Web services models.

7. The invoked Web service function responds to the request by returning a SOAP response containing returned parameters.

Web services are reusable software components that are devoid of presentation and focus on business logic or data access regardless of client devices. Therefore, more client software and devices can use data retrieved from Web services or share the same business logic. This represents the complete decoupling of business logic and business data from presentation and user interface. This actually gives user interface designers and programmers greater access to various business functions and data.

Increasingly, vendors are providing tools to support the development of Web services to ease their creation and consumption. From a programmer's viewpoint, creating and consuming Web services are similar to building and calling program functions or classes. Since many development tools can automatically generate the necessary client-side and server-side stubs (i.e., the plumbing) for Web services, the skill level required to use Web services is greatly reduced.

Web services have been used to wrap a legacy system's functions to support application to application integration inside a firewall (Vecchio, 2001). Web services have also been promoted as technology that could replace platform specific technologies such as CORBA and DCOM as a middleware standard for distributed computing applications.

4. Factors affecting adoption and diffusion of e-business standards

According to the innovation diffusion model by Rogers (1995) and the Technology Adoption Life Cycle by Moore (1991), we can create an XML and Web Services Technology Adoption Model as depicted in Fig. 4. This model is used in this paper to analyse factors affecting various adopters at various stages of e-business standard adoption. Factors identified through literature research and case studies are discussed in this paper.

4.1. E-business standards and technology adoption model

The XML and Web Services Technology Adoption Life Cycle Model may help us understand issues in standard adoption decision. The five adopter categories (innovators, early adopters, early majority, late majority, and laggers) used in Fig. 4 refer to individuals or organizations who decide to adopt a standard or technology during five phases of technology adoption life cycle.

1. *Innovators*, creators of the new standards, who are involved in the standard organization during the *inception phase* of these standards. These are individuals who have prior experiences with other related earlier standards or are vendors who have vested interests in promoting these "solutions" as standards.

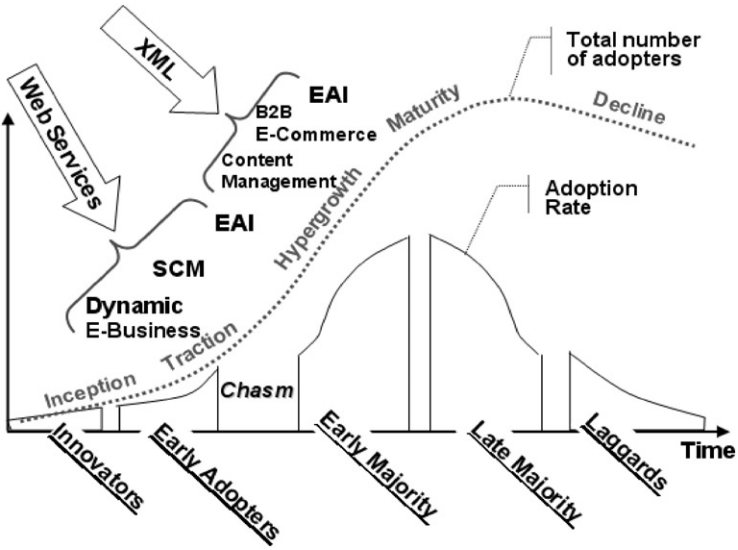

Fig. 4. XML and web services technology adoption model.

2. *Early adopters* have the vision to adopt an emerging technology because of business opportunities or technology needs. In this *Traction phase*, the supporting tools and products are not matured enough to support the standards. Therefore, there are tremendous obstacles in building applications using e-business standards. Many software companies and system integration firms belong to this category.
3. *Early majority* prefer to stay away from bleeding edge technologies to avoid the risk associated with emerging standards. But, they are quick to adopt and implement these standards when benefits are demonstrated by the early adopters. This *hypergrowth phase* has the highest adoption rate due to network externality effect (Church and Gandal, 1993). This is the beginning of a mass market. *The Chasm* is the time gap in the technology adoption life cycle, which is between the Traction phase and Hypergrowth phase (Moore, 1991).
4. *Late majority* represent about one-third of potential users who dislike discontinuous innovations. They believe in traditional technologies rather than emerging ones. They adopt e-business standards mainly because their major trading partners require them to use the same standards. Small to medium enterprises (SMEs) that do not have up-to-date IT infrastructures often belong to this category. This is when the standard has entered the *maturity phase*.
5. *Laggers (Traditionalists)* do not engage with standards or new technology products. They may not have the resources, skillful IT professionals, organizational disciplines, or the business foresights to deploy standards. When they are ready for these standards, newer standards may have been developed to replace existing ones. This is when the technology has entered the *decline phase*.

Currently, Web services are still in the traction phase, while XML is considered to be in a hypergrowth stage (Colan, 2001). To ensure the success of e-business standards, companies that implement these standards ought to identify barriers that prevent the early majority from crossing the chasm and to deploy tools and strategies to reduce the traction for e-business standard adoption.

4.2. *Decision perspective of e-business standards adoption and diffusion*

We are taking a decision making perspective to study the adoption and diffusion of e-business standards. Based on our literature reviews, we have identified several factors affecting the adoption and diffusion decisions. Fig. 5 illustrates possible relationships among these factors according to an adapted IDEF diagramming notation (Marca and McGovan, 1993). Inputs in Fig. 5 are factors that may influence the *decision maker* in charge of the e-business standards adoption decision. The decision criteria are controls of the decision, while decision makers are treated as mechanisms that carry out the decision process. The outputs of the adoption decision include answers to the following questions: What applications should use these standards? What standards should be adopted? How and when should these standards and related technologies be implemented?

The input factors that have direct influences on the adoption decision process are grouped into three categories:

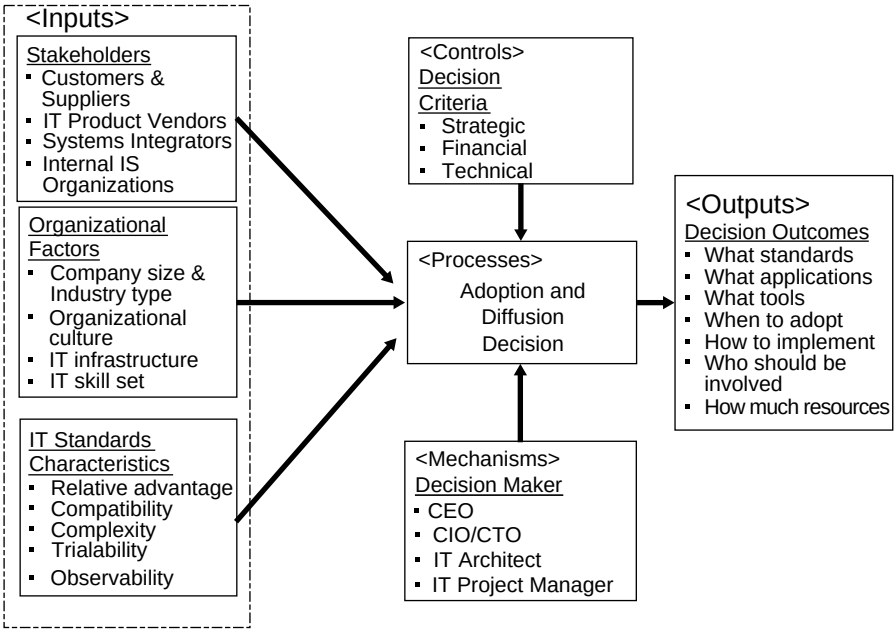

Fig. 5. Factors affecting e-business standards adoption and diffusion.

Stakeholders: The stakeholder analysis method used in strategic planning is applied in this research (Mason and Mitroff, 1981). Stakeholders in the context of e-business systems standards adoption are defined as individuals or organizations who have influence over or may be impacted by the adoption decisions. These stakeholders include customers and suppliers, IT product vendors, systems integrators, and internal IS organizations, as well as individuals working in these organizations.

1. *Customers and suppliers*: Many B2B integration projects may be forced upon firms, particularly SMEs, by their major suppliers or customers. The more firms adopt a standard; the faster it will be adopted by the general population due to the network externality effect. Some firms, such as Dollar Rent a Car, proactively choose to build extended enterprises using standards that allow them to reach out to more customers or suppliers and achieve seamless integration to gain competitive advantages.
2. *IT product vendors* usually are the Innovators and Earlier Adopters in standard adoption. Major IT vendors such as Microsoft, IBM, and Oracle are actively participating in e-business standardization efforts and are incorporating these standards in their software products and tools.
3. *Systems integrators* have many highly skillful IT personnel and more resources to keep up with new standards. Some systems integrators, such as CommerceQuest, are promoting their use of standards as beneficial because they provide the promise for lower maintenance and support costs (CommerceQuest, Inc., 2002). In one project, CommerceQuest estimated that they achieved 70% reduction in development time and costs, reduced maintenance costs, and 100% payback within 1 year by its customer, a wireless telecommunication firm (IBM, 2002).
4. *Internal IS organizations*. A firm's IS personnel are either making the adoption decision or are affected by the standard adoption decision. Some IT personnel are reluctant to accept new standards due to potential steep learning curve. However, others may want to experiment and learn how to use these standards to gain marketable skills.

Organizational Factors: There are several key organizational factors that are affecting the e-business standard adoption decision. Here are some potential impacts of these factors.

1. *Company size and industry type*: Big manufacturing and retailing firms that need to integrate with their suppliers often belong to the Early Adopter category. The Early Majority category also includes defense industry and large government agencies that are traditionally strong supporters of standardization efforts due to the distributed nature of their organization structures and IT infrastructures.
2. *Organizational culture*: Firms that have a culture of innovation are more likely to experiment with e-business standards at earlier stages. IT organizations that emphasize conformance are more like to adopt e-business standards.
3. *IT architecture*: IT standards adoption decisions are made in the context of the overall IT architecture. IT architecture consists of components including

application development tools, databases, networks, computer hardware, and business applications. Firms that have distributed IT architectures as well as e-business initiatives are more likely to adopt e-business standards. Standards-based IT architectures allow firms to build information systems that are independent of any specific vendor's technologies.

4. *IT skill set*: Companies such as IT vendors and system integration firms that have highly skillful IT personnel tend to be earlier adopters of e-business standards. Small and medium size enterprises (SMEs) are often in the Late Majority category due to lack of the necessary IT skill sets and resources. They are often pressured by their large customers to adopt the certain standards such as XML to transmit data (Gonsalves, 2002).

Roger found five perceived attributes of an innovation that may determine the innovation's rate of adoption (Roger, 1995). These attributes are quite useful in assessing the adoption decision of XML and Web services standards:

1. *Relative advantage* is the degree to which an innovation is *perceived* as better than the idea it supersedes. The degree of relative advantage may be measured in economic terms such as faster development, less maintenance, and cost saving, but strategic advantages and prestige for using emerging technologies are also important factors. The greater the perceived relative advantage of an innovation, the more rapid its rate of adoption will be.
2. *Compatibility* is the degree to which an innovation is perceived as being consistent with the existing technologies and past experiences of potential adopters. XML and Web services can be implemented as wrappers of older technologies to provide interoperability of new technologies with legacy systems. Companies that have experiences of using EDI and building distributed systems may find that these experiences are very valuable in using XML and Web services.
3. *Complexity* refers to the degree to which an innovation is perceived as difficult to understand and use. There are several factors affecting the complexity of e-business standards related to their adoptions. In our study, we found that many e-business standards are relatively simple, comparing to other standards or alternative approaches. Reducing the complexity of standards is critical to the standards' success. One of the design goals in XML stated that "XML documents should be human-legible and reasonably clear [W3C, 2000]." Therefore, even though there are no development tools that support XML, developers can still use existing tools such as Notepad to create and view XML documents. Web services are simpler than CORBA (an open standard) and DCOM (a Microsoft proprietary standard) and may quickly replace the later two older "standards." From both the IT producer's and consumer's viewpoint, a simpler standard usually accelerate the support of the standard in various tools provided by IT vendors. The adoption of these tools by IT consumers may further simplify the adoption of the standard.
4. *Trialability* is the degree to which an innovation may be experimented with on a limited basis. XML and Web services can be introduced incrementally in an

organization because preliminary experimentations of these two standards do not require major IT infrastructure change. In one Web services project at Dollar Rent a Car, a proof of concept was developed in two weeks (Microsoft, 2002). Standards, such as Web services, that have high Trialability may be adopted more quickly than innovations that are not divisible (e.g., ERP packages). Through initial pilot projects, organizations can reduce the uncertainty and risk for major deployment of these standards.

5. *Observability* is defined as the degree to which the results of an innovation are visible to others. XML is used to represent data structures and contents, not presentations. Web services are non-visual business logic; Web services have been referred to by some people as "*web sites without the user interface*." Therefore, XML and Web services are not very observable from a user interface viewpoint. However, Web services are often used in the integration of heterogeneous applications inside and outside the firewall. These applications usually are usually strategic information systems, therefore, if successful, the results are quite visible. Explicit promotion and articulation of the e-business standards used in these projects may be required to increase the observability to major decision makers. This awareness, as well as continuous support and commitment of decision makers in the e-business adoption process, are important factors to ensure the diffusion of these standards to other relevant projects (Rogers, 1995).

In summary, standards that are perceived by individuals as having *greater* relative advantage, compatibility, trialability, observability, and *less* complexity will be adopted more rapidly than other standards. In the case of e-business standards adoption, the only characteristic that may hinder the adoption of XML and Web services is the lack of *direct observability*.

Decision criteria: Companies decide to adopt standards may use different criteria when they make the decision. Multiple criteria are often used in the adoption decision. According to InfoWorld CTO Network Survey and compiled cases from Microsoft and IBM (Sullivan, 2002; Microsoft, 2002; IBM, 2002), we have identified and categorize these criteria in the following:

1. *Financial*: Companies often cited reusability and faster development time, factors that translate to short- and long-term cost saving. Integration projects are very expensive, and Web services will help reduce costs by 30% or more. The investments in using XML and Web services are incremental in most cases. Reaching out to new customers brings in new revenue streams. This is one of the major reasons that firms can justify their investment in e-business standards.
2. *Strategic*: Many integration projects that trigger the use of standards are inter-organizational information and strategic information systems. Therefore, the decision to adopt these standards is of strategic importance. Faster development cycles, an often quoted benefit of using these standards, enable firms to gain a competitive edge because of shorter time-to-market of new products and services.

3. *Technical*: The technical merits of XML or Web services over competing standards include their simplicity and their interoperability in a distributed computing environment regardless of programming languages, platforms, and object models. There is some performance penalty in using XML and Web services; however, the flexibility often outweighs the performance concern.

Decision makers: At the early adoption stage, adoption decisions of deploying XML and Web services are often made by senior systems architects or IT project managers at the project/application level. The adoption process often started with a proof-of-concept pilot project to reduce the risk involved. Once there are some successful cases in implementing these standard-based technologies, then the CTO and CIO are involved in enterprise-wide standard adoption to ensure interoperability among applications. Companies can be both consumers and providers when deploying these standards. As Web services providers, companies can expose almost any functionality or content to their business partners' information systems. As Web services consumers, companies can also integrate their business partner's Web services to provide more comprehensive services and satisfying experiences to their own customers. These capabilities may change the underlying business models of many companies. Sooner or later CEOs will be involved in business and strategic decisions regarding how these e-business standards and technologies may impact their businesses and how they should seize the potential new business opportunities.

5. Case studies of XML adoption and diffusion

In this section, we report studies of two e-business adoption cases. The first case is based on in-depth interviews of a firm that uses XML and Web services for internal enterprise integration. The second case is based on secondary data of a firm that use Web services for business to business e-commerce. These cases shed some light on dynamic aspects and business drivers for the adoption and diffusion of e-business standards.

5.1. A case of enterprise application integration

A major telecommunications company located at United States' East Coast was merged with another telecomm firm. Many of the internal application systems were running on different platforms and using different languages. Combining two applications that support the same business function into one required costly and lengthy redevelopment efforts. After careful evaluation, two major business functions, billing and trouble ticket management, were given high priority for this EAI initiative. The merged new firm hired a consulting firm that suggested the use XML standards. With the consulting firm's assistance, several XML DTD files were defined. These XML DTD standards were used to integrate applications data from the original two companies. XML parsers and message queue application servers were used to implement the EAI project.

In 2001, a DSL provisioning system developed in Java Server Pages and Java Servlets required an address validation function at this company. After careful evaluation of 5 different vendor's solution, a senior architect suggested the use of a third-party product that provided a set of DLL-based APIs to access a proprietary address validation database. CORBA was considered, however, CORBA does not work with this third party product. A Web services approach was taken to solve this integration problem. First, several high level address validation functions were developed in Active Server Pages (ASP) that called the APIs from third party products running behind an IIS web server. A set of SOAP interfaces was developed using SOAP Development Toolkit from Microsoft to wrap around the ASP programs. Hence, the main application written in Java can interact with these validation functions via SOAP method invocation. This project illustrated true interoperability of Web services technologies.

No obvious e-business standardization efforts were observed in this firm. Decisions to use standards based technologies were often determined by project level senior architects. The decision was mostly based on the need to solve technical problems instead of standardization policy of the firm.

5.2. *A case of B2B integration*

Dollar Rent A Car is one of the world's largest car rental agencies, with more than 400 locations in 26 countries. This includes more than 250 locations and a fleet of 75,000 cars in the United States, along with significant presences in Australia, Canada, the Caribbean, and Latin America. Dollar's parent company is the Dollar Thrifty Automotive Group, Inc. (NYSE:DTG), which posted more than USD$1 billion in combined revenues for 2001. "In the 2002 first quarter, over 39% of Dollar's retail reservations were booked on the Internet, compared to 31% in the 2001 first quarter (Dollar.com, 2002)."

Dollar Rent A Car is an earlier adopter of Web services (Metz, 2001; Microsoft, 2002; Ricciuti, 2001). Using Microsoft's SOAP Toolkit, Dollar has built a service that is embedded in the Southwest Airlines Web site. Travelers can book cars from Dollar after purchasing airline tickets from Southwest web site. The rental car booking data is then sent in plain XML format to Dollar's Web services. Once there, the data are converted into SOAP before entering the company's back-end VMS-based application known as Quick Key.

Dollar carefully evaluated some existing technology for the integration project with Southwest Airlines. Dollar identified four possible technologies for connecting with other systems—CORBA and the Internet Interoperability ORB Protocol (IIOP), Java RMI, DCOM, and socket programming. There were significant drawbacks with each option. For examples, CORBA and DCOM can be used to support integration of certain computing platforms such as UNIX or Microsoft Windows. CORBA generates port numbers randomly for network communications. When you try to use CORBA over the Internet, CORBA messages may be blocked by firewalls. IIOP is considered to be too complicated. Java RMI is limited to Java language implementation. Web services standards are designed to work with existing

Internet protocols and therefore can be used inside and outside the firewalls to support distributed applications. The relative advantages of Web services convinced Dollar to take the risk to use Web services for this B2B integration project.

The communication behaviors of the major decision maker in this case are consistent to observation of an early adopter by Roger (1995). Peter Osbourne is the Group Manager of Advanced Technology Group at Dollar Rent A Car. He is the head of the 10 person development group initiated and implemented the Web services project. He learned about XML and SOAP at an XML conference in September 1999 and he also heard about how SOAP can be used to create Web services in January 2000 at a Microsoft sponsored seminar on Visual Basic. He had more social participation by attending professional conferences to learn about new standards and technologies. In the spring of 2000, Osbourne suggested that Dollar used XML Web services to integrate with new business partners. After a proof-of-concept prototype built by a developer in 2 weeks, the simplicity of the solution convinced the top level IT manager, the Executive Director of Application Development at Dollar, to go along with Osbourne's decision to use Web services.

Adoption theory suggests that "earlier adopters are more highly interconnected through interpersonal networks in their social system than later adopters (Roger, 1995, p. 273)." In the early evaluation stage, Osbourne posted a technical question to Internet newsgroups regarding whether SOAP could support the transaction volumes required. His question was answered by Kent Sharkey, a Microsoft evangelist for SOAP and XML Web services. Using newsgroups, an Internet era's "personal network" in adopters' social system, the earlier adopters, such as Osbourne, also have more change agent (i.e., Kent Sharkey) contact. Since tools such as newsgroups and other online resources are ready available to everyone, many adopters can have access to critical information about the standards and technologies to help them in their adoption decision.

Since the first project with Southwest Airline, Dollar has used the XML Web service-based interface several times over to integrate its reservation system with other airlines or travel agencies. All Dollar has to do is translate the airline's data to the reusable SOAP interfaces that extends the usefulness of its legacy reservation system. The low cost and rapid development cycle using Web services allows Dollar to capitalize on new business opportunities.

Dollar developed a mobile web site reusing the Web services to interface with the mainframe reservation system in less than 30 days. The project was at the request of Dollar's marketing group. They learned about a mobile-enabled site initiative by one of Dollar's competitors. The Microsoft Mobile Internet Toolkit and Visual Studio.NET were used to create a mobile site that can be accessed by a wide range of mobile devices and standards. The web site was launched on August 29, 2001. Dollar is the first car rental company to deliver a fully mobile-enabled Web site.

Due to the success of several high profile projects using Web services, there is top management support and resources to expand the usage of Web services. Tour operators are a critical sales channel for Dollar. Tour operators currently either use FTP to transfer flat files, or rely on an EDI connection to book a car reservation electronically. The EDI connection was expensive and inflexible. It took 8 weeks on

average to develop a parser to process the unique file format from a new partner. Currently, Dollar has recently deployed Microsoft BizTalk Server 2000 to supports the business process and workflow layer functionalities of the Web Services Standards stack. A proof-of-concept prototype was developed in 2 days that received a flat file, parsed it, and used Dollar's existing XML Web services interface to send reservations to the mainframe. Using the BizTalk tool that support high level Web services functions may reduce the time required to integrate each new business partner by 75%. Dollar "keeps finding uses for Web services (Whiting, 2002)." The continuous and successful diffusion of Web services in Dollar are in part a result of its careful planning of standards adoption via proof of concept prototype, and well-defined business objectives.

6. Conclusions

Our study found that technical merits of XML and Web services are strong reasons for people to adopt them. Many adoption decisions are made at the project level, instead of at the enterprise level. Once there are more successful implementations of XML and Web services in high-profile strategic systems, the standards adoption issues may rise to the CTO and CIO levels. Soon, innovative business strategies and business models, as well as supply chain integration relying on Web service-based architecture will become the driving force for XML and Web services adoption (Sullivan, 2002).

Standards such as HTML and HTTP have facilitated the adoption of Web technologies. New e-business standards such as XML and Web services may also accelerate the adoption of service-oriented architecture by forward-looking firms. This could generate a second wave of e-commerce revolution. Companies may soon participate in dynamic e-business via collaborative Web services that provide remote access to programmable business services and use XML for data interchange. Further studies are required to help organizations to understand organizational implementation issues of standards, new Web services development methodologies, and new business models and strategies to take advantages of Web services.

Roger's (1995) diffusion of innovation model has identified additional categories of variables that may determine the rate of adoption of innovation. These additional aspects include the nature of social system participated by standard adopters, the promotion efforts of change agents for standards, the communication channels about standards, and the type of innovation decision (i.e., optional, collective, and authoritative). More research is required to formally identify these factors and examine relationships among them.

Another dimension of adoption research is the process through which individuals or decision making units are going through various stages in evaluating, adopting, and implementing these standards. Longitudinal studies are required. The challenges in empirical research of standards include the difficulty in separating individual and organizational decisions, as well as the dynamic interactions of stakeholders. Another confounding factor is that standards are often embedded in development

tools; therefore, it is difficult to separate the standard adoption from the adoption of tools. A more refined standard adoption and diffusion model may help standards organizations to formulate strategies in promoting current and future e-business standards. It will also help companies launching e-business initiatives to make timely adoption decisions and formulate appropriate diffusion plans for e-business standards.

References

Atkinson, B., et al., 2002. Web Service Secuirty (WS-Security), Version 1.0 05 April 2002 http://www-106.ibm.com/developerworks/library/ws-secure/.

Christensen, E., Curbera, F., Meredith, G., Weerawarana, S., 2001. Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, at http://www.w3.org/TR/wsdl.

Church, J., Gandal, N., 1993. Complementary network externalities and technological adoption. International Journal of Industrial Organization 11, 239–260.

Colan, M., 2001. Evolution or Revolution? A Technical Overview of Web Services, at http://ibm.com/developerworks/speakers/colan.

CommerceQuest Inc., 2002. WebSolutions, June 12, 2002, at http://www.commercequest.com/web_solutions.asp.

Curbera, F., Ehnebuske, D., Dan R., 2001. Using WSDL in a UDDI Registry1.05, UDDI Working Draft Best Practices Document, June 25, 2001, at http://www.uddi.org/pubs/wsdlbestpractices-V1.05-Open-20010625.pdf

Dollar.com, 2000. Dollar Thrifty Automotive Group Reports Record First Quarter Earnings, at http://www.dollar.com/company_information/press_releases/view_new.asp ?id = 24/.

Foody, D., 2002. Accelerating the adoption of web services in your enterprise. Web Services Journal 2 (6), 6–10.

Gonsalves, A., 2002. GE Unit Adds Web Services, InformationWeek, April 30, 2002 at http://www.informationweek.com/story/IWK20020430S0008.

Gottschalk, K., et al., 2002. Introduction to web services architecture. IBM Systems Journal 41 (2), 170–177.

IBM, 2001. The Executive E-Business Infrastructure Guide, IBM.

IBM, 2002. CommerceQuest Solves Telco's Integration Challenge Using IBM Technology, 2002, at http://www-3.ibm.com/software/ebusiness/jstart/casestudies/commercequest.pdf

Marca, D.A., McGowan, C.L., 1993. *IDEF0—SADT Business Process & Enterprise Modelling*. Eclectic Solutions Corp, October 1993.

Mason, R.O., Mitroff, I.I., 1981. Challenging Strategic Planning Assumptions: theory, cases, and techniques. Wiley, New York.

Metz, C., 2001. Testing the Waters, at http://www.pcmag.com/article2/0,4149,154693,00.asp

Microsoft, 2002. Dollar rent a car breathes new life into legacy systems using .NET connected software, Microsoft .NET in Real World, March 2002.

Moore, G., 1991. Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. Harper Collins.

OASIS, 2002. About OASIS, at http://www.oasis-open.org/who/.

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

Ricciuti, M., 2001. Strategy: Blueprint shrouded in mystery, at http://news.com.com/2009-1001-274344.html

Rogers, E.M., 1995. Diffusion of Innovations, 4th Edition. Free Press, New York.

SAX Project, 2002. About SAX, June 2002, at http://www.saxproject.org/.

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

Sullivan, T., 2002. App dev on the web services path, at http://www.infoworld.com/articles/fe/xml/02/06/10/020610feapp.dev.xml

W3C, 1999. XSL Transformations (XSLT) Specification Version 1.0, W3C Working Draft 21 April 1999, at http://www.w3.org/TR/xslt.

W3C, 2000. Document Object Model (DOM) Level 2 Core Specification, Version 1.0, W3C Recommendation 13 November, 2000 at http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/.

W3C, 2001. Extensible Stylesheet Language (XSL) Version 1.0, W3C Recommendation, October 15, 2001, at http://www.w3.org/TR/xsl/.

Vecchio, D., 2001. Legacy software: Junkyard Wars for Web Services? Gartner Symposium / ITxpo 2001, October 2001.

Whiting, R., 2002. Dollar Discovers New Uses, Along With A Few Skeptics, May 27, 2002, at http://www.informationweek.com/story/IWK20020524S0003.