

ebXML vs. Web Services

Comparison of ebXML and the Combination of SOAP/WSDL/UDDI/BPEL

Peter Gerstbach

Vienna University of Technology
e0125397, 066 933
peter@gerstbach.at

Abstract. E-business has been dominated for a long time by traditional electronic data interchange (EDI) standards. The development of XML led to new technologies promising a new way of conduct business using the Internet. This paper compares two well established approaches: on the one hand ebXML, a modular suite of specifications specialized in e-business, on the other hand Web Services, where the SOAP, UDDI, WSDL and BPEL can be combined to yield similar results. After the concepts of both technologies are introduced, they are compared to each other. The historical overview gives insights into the reasons for development; the technical comparison reveals commonalities and differences. Then approaches are discussed that may be used within both technologies to increase interoperability. Finally, the section about vendor support shows the current status of implementation.

1 Introduction

Business-to-Business (B2B) protocol specifications are used to integrate applications across organizations to perform electronic business. Two widely adopted standards, compared in this paper, are ebXML and a combination of Web Service technologies, being SOAP, WSDL, UDDI and BPEL. To improve readability the latter combination is simply referred to as “Web Services” in this paper, unless stated otherwise.

The idea of exchanging business documents between applications is nothing new. It has first been implemented in the 1960s using early, not standardized EDI systems. Although UN/EDIFACT provided an international standard in 1990, electronic business is expensive and not widely used. The implementation of EDI is based on many different protocols, requires rare skills and is almost impossible accomplish for all but the largest companies. The emerging Internet and the development of XML gave e-business a significant upturn. Although most technical barriers have been tackled, the most important question for doing e-business is *what* trading partners say and *how* they say it.

Web Services and ebXML are both representatives of the Service Oriented Architecture (SOA). This architecture depends on loose coupling, dynamic binding and high interoperability to offer modular applications: *Services*. Web Services

are a collection of different languages whereby each covers a specific aspect of the whole. Although they are mainly used for business-to-business collaboration, they are not limited to it. In parallel to Web Services, ebXML was developed to enable enterprises to conduct business over the Internet. Both approaches have a different background and vary in vendor support. Furthermore they are not compatible. ebXML has its power in its strong conceptual background, while Web Services stand out with a good tool support and dynamic development. That is one reason why ebXML might be more suitable for more stable B2B scenarios, whereas Web Services are capable of doing loosely coupled collaboration.

1.1 E-Business using Web Services

Web Services provide a standard means of interoperating between software applications. The Web Service Architecture provides a conceptual model that describes the relationship between the components of Web Services [2].

A *service* provides an abstract set of functionalities, an *agent* is a concrete piece of software or hardware that sends and receives messages to form a service. The party providing an agent that implements a particular service is called the *provider entity*, the party that consumes the service is the *requester entity*. To make communications between the provider and the requester possible the service must be described in a machine readable specification. It defines the message formats, datatypes and transport protocols used during communication. Figure 1 outlines the roles, operations and artifacts in the web service architecture.

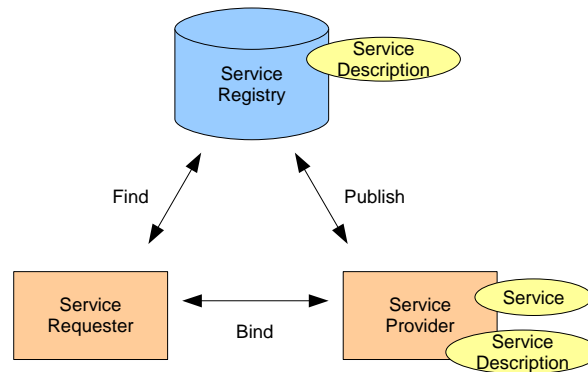


Fig. 1. Web Service Architecture: Roles, Operations and Artifacts

Among the two entities provider and requester, there is a central registry that allows to find published services. This kind of discovery is known as the *Registry Approach* where a central registry is involved. The provider publishes its services on the registry while the requester finds it and invokes (*binds*) the

service. However, an invocation of a service without using a registry is as well possible and currently more widespread.

Current Web Services use the combination of SOAP, WSDL and UDDI to implement the architecture. SOAP is a standard for exchanging XML-based messages, normally using HTTP as a transport protocol. WSDL or *Web Service Description Language* is used to describe the services in a machine-processable way. It defines the operations and messages, the used transport protocol, a message format and an endpoint where the service can be invoked. UDDI (*Universal Description, Discovery and Integration*) provides the distributed directory where service can be published and found.

When automating business processes between different organizations, it is required to specify these processes to achieve interoperability. BPEL (*Business Process Execution Language*) is an XML-based language for this purpose. It allows specifying business processes and business interaction protocols. BPEL extends the Web Services model and enables it to support business transactions and to adopt the concept of *programming in the large* (programming long-running asynchronous processes).

BPEL is based on Microsoft's XLANG and IBM's WSFL, two languages with similar purpose, and was proposed by the two companies and BEA under the name BPEL4WS in 2002. Later it was enhanced and renamed to BPEL. The language relies on WSDL and thus provides facilities to extend web services:

- Primitive activities including sending and receiving a message, replying to a message and error handling.
- Structured programming constructs such as conditions, loops, sequential and parallel execution
- Data handling to support variables and multiple expression languages (e.g. XPath, XSLT)

With BPEL it is possible to link different services together into a business process using composition. On the one hand it is an implementation language, on the other hand it can be used to describe the interfaces of a business process, using *abstract* processes.

Implementing an executable business process with BPEL means defining a new Web Service by composing already existing services. This kind of collaboration between services is also called *orchestration*, where an executable business process is under control of a single endpoint. The interface of the composite service is described as a collection of WSDL ports. Figure 2 illustrates a Web Service implemented as a BPEL process. Inside the process a request to another service is sent, when the acknowledgment, and finally the response to the request has been arrived, the BPEL process comes to the end. The trigger starting the process might be a web service too. This is indicated by the two arrows on the left-hand-side of the figure.

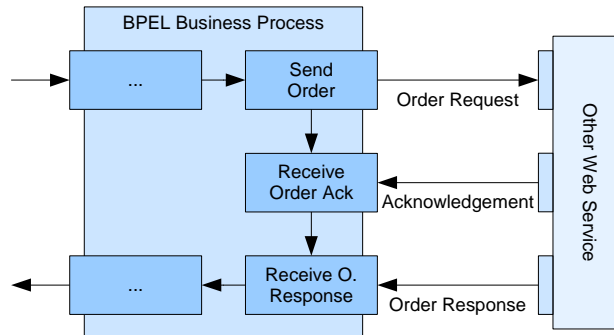


Fig. 2. A Web Service implemented as a BPEL Process

1.2 Introducing ebXML

ebXML (Electronic Business using eXtensible Markup Language) is a suite of specifications sponsored by OASIS and UN/CEFACT that enables enterprises to conduct business over the Internet [12]. The specifications cover the analysis of business processes and business documents, the documentation of a company's capabilities and the document transfer to conduct e-business.

Unlike many other XML implementations ebXML does not solely define an XML grammar and vocabulary. It provides an entire architecture and defines a new way of thinking about business and documenting it. The very data-centric formula *Internet + WWW + XML + some XML vocabulary + some new business model = the new way of doing business* [9] is overcome by focusing on process-centric methods. The starting point of ebXML in 1999 was the combination of UN/CEFACT's business knowhow with the experience of leading XML experts to form open and inter-operable standards.

With ebXML, companies are able to define how to conduct business using a specific vocabulary. Core components are used to build predefined documents. Messages are sent using standardized protocols and formats. All of this information is stored in ebXML registries.

Business Processes and Business Document has to be created prior to their use. A Business Process Specification describes how a business works and a Business Document Specification describes the information exchanged between the partners. Both documents are defined by using the *Business Process Specification Schema* (BPSS). Later, they can be translated to XML Schema or DTD. To facilitate their creation and avoid reinventing the wheel, these documents can be composed of reusable and extendable *Core Components*.

The *ebXML Messaging Service* (ebMS) provides a simple way to exchange business documents using standard protocols. It is based on the SOAP specification, but provides additional headers and envelopes to include information about transport, routing, policies and security.

An *ebXML Registry* provides means for finding organizations, business processes, core components and other objects. Therefore it does not store the actual objects but metadata about and associations between them.

Using an ebXML system involves the steps described in figure 3 and the following list [3]:

1. **Search for a Trading Partner.** An ebXML registry contains information on potential trading partners. During the search the registry is queried for a trading partner that offers the wanted service. Services are described using Collaboration Protocol Profiles (CPPs).
2. **Create an Agreement.** Based on the CPPs of both partners a Collaboration Protocol Agreement (CPA) is composed. It specifies what kind of business is to be performed and how. Therefore it includes information on technical issues such as protocols and requirements regarding security or acknowledgments. Usually the CPA is negotiated after being proposed by one party.
3. **Configure both Business System Interfaces using the CPA.** Based on the agreement it is now possible to configure an ebXML enabled application.
4. **Begin performing Business Processes.** The final step is the execution of the process. Based on the fact that both partners use the same documents describing the business process (the CPA) collaboration between them is possible.

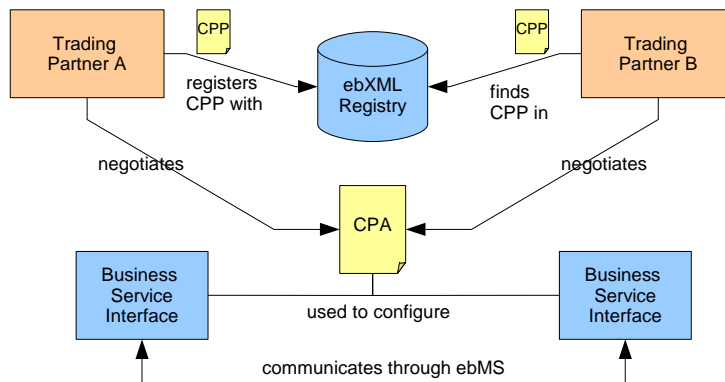


Fig. 3. ebXML System Overview

2 Comparison

The first section introduced the two architectures *Web Services* and *ebXML* providing common background information. The following comparison uses different views on the subject to outline the pros and cons of each approach: first the history of both technologies are compared, followed by a discussion on the technical implementation. Finally some aspects about interoperability illustrated, and finally a market overview about vendors of B2B products is given.

Web Services and ebXML are built using the same technology but with different purpose. The Service oriented Architecture and the use of XML are the most important terms here. More precisely, both are using SOAP as a messaging protocol, rely on registries to make discovery using dynamic binding possible (UDDI and ebXML Registry resp.) and use other XML standards such as XML Schema to describe documents. However, the purpose of Web Services and ebXML differs: Web Services expose applications on the Web so that everybody can call them from everywhere - a service-based approach; ebXML is intended for business partners to exchange well defined documents using an agreed business process - a contract-based approach. Since Web Services are more flexible (they have a broader area of application) they can also be used for an ebXML-like process, especially by using BPEL to describe processes.

2.1 Historical Comparison

The success of XML has been fundamental for the evolution of Web Services. The driving forces behind the development of new technologies and standards have been on the one hand companies and on the other hand standardization organizations like the W3C and OASIS. Companies soon identified the advantages of XML related concepts, sometimes trying to gain an advantage in competition, sometimes driving standardization to help leveraging the market. Standardization organizations try to develop interoperable standards to increase the potentials of existing technologies and to increase their convergence and adoption. These driving forces caused the development of a multitude of approaches related to XML and Web Services. The number of the *WS-** specifications has increased considerably in the past. The result is a very dynamic and lively community leading to flexible and feature-rich solutions for Web Services. However, the market is complex and barely predictable now.

The development of ebXML is somehow different to Web Service's. EDI has probably reached its critical mass after more than 20 years of deployment. Up to 95% of the Fortune 1.000 are using EDI compared to only 2% off Small and Medium Enterprises (SMEs) [14]. EDI is generally agreed to be too complicated, too expensive and not flexible enough. It is obvious that there is still a digital divide between small and mediumsized enterprises. The limited degree of computerization is a constraint for the adoption of e-business among smaller companies [6]. ebXML is the attempt to create a global market where

enterprises of any size, anywhere can find each other and conduct business electronically using commercial off-the-shelf software. While the parties concerned with Web Services are mainly from the IT industry, ebXML is backed by OASIS and UN/EDIFACT, providing strong background knowledge gained from many years of B2B experience. Thus, ebXML is more a holistic framework to support business to business transactions using XML.

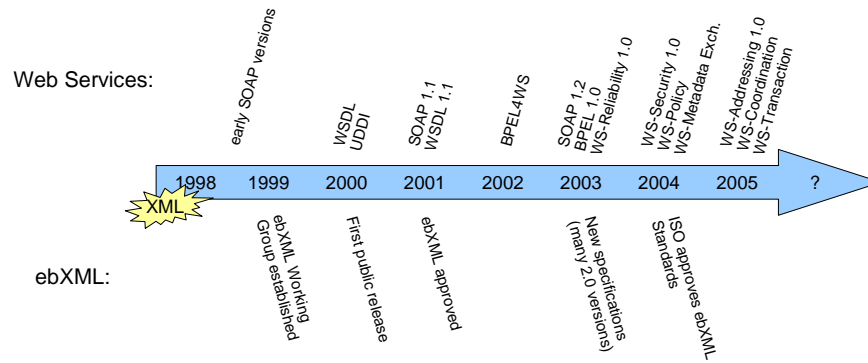


Fig. 4. ebXML and Web Service Timeline

Figure 4 shows a timeline when relevant standards and specifications for ebXML and Web Services has been published or adopted, respectively. It all started with the XML Recommendation in 1998, followed by the establishment of the ebXML working group and the release of the first ebXML standards on the one hand, and the development of SOAP, WSDL, UDDI and finally BPEL on the other hand. While the development of new Web Service related standard has accelerated for the last two years, ebXML is now already in a period of consolidation.

The current status of ebXML and Web Service-related Standards and drafts are listed in table 1. The ebXML specifications are maturing, many are in version 2.0 approved standards. In addition there are conformance tests available. Since Web Service standards are developed by different organizations some standards are widely adopted, others are still under development.

2.2 Technical Comparison

The historical comparison of Web Services and ebXML already pointed up their different background. Regarding the design principles, ebXML follows a top-down approach¹: First the requirements for e-business are evaluated, secondly

¹ compare [11]

Specification	Approved Version	Draft	Specification	Approved Version	Draft
Messaging Service	v2.0	v3.0	SOAP	v1.2 (W3C)	
Registry	v2.0	v3.0	UDDI	v3.0 (OASIS)	
CPPA	v2.0	v2.1	WSDL	v1.1 (W3C)	v2.0
BPSS	v1.0	v2.0	BPEL	BPEL4WS v1.1 (various)	v2.0
Core Components	v2.0	v2.1	WS-Reliability	v1.1 (OASIS)	
			WS-Security	v1.1 (OASIS)	
			WS-Policy	v1.1 (various)	
			WS-Addressing	v1.0 (W3C)	
			WS-Transaction	v1.0 (various)	

Table 1. Status of ebXML and Web Service Specifications

the specifications are implemented to fulfill them. Keeping the business background of UN/EDIFACT in mind, this design approach is evident. Based on the strong knowledge about business processes and the experience with EDI, the aim is to build an integrated e-business suite. In contrast, the development of Web Services follows a bottom-up approach. Specifications for specific core requirements are implemented (SOAP for messaging, UDDI for discovery, WSDL for service description, and other WS-* specifications) and afterwards being assembled to build a whole. The bottom-up approach yields an increased flexibility but may not cause all requirements for e-business to be met. Web Services are suitable for conducting e-business but they are neither specialized in e-business nor limited to it. In fact Web Services are useful for many different domains, including catalog and web search, maps and traffic, payment and clearing, publishing of photos and blogs, and of course e-business in the large.

This section discusses the two technologies regarding their technical realization. First, the Web Service Protocol Stack is compared to ebXML outlining the commonalities and differences on each layer. Secondly, the kind of collaboration is investigated, and finally, the impacts on coupling between different parts of the e-Business systems are compared. The subsequent section uses a comparison approach based on eCo Layers and Workflow Aspects to compare both approaches.

Figure 5 shows a comparison of languages used for Web Services and ebXML. It is restricted to a subset of the Web Service Protocol Stack that is relevant to both technologies (for example the transport protocols are left out).

On the messaging layer, Web Services uses the SOAP protocol that is mostly transported using HTTP. SOAP has been initially developed by Microsoft and is now widely used in version 1.2, recommended by the World Wide Web Consortium. SOAP is a simple XML-based message exchange protocol. It uses envelopes to package messages, while an envelop consists of a header containing metadata

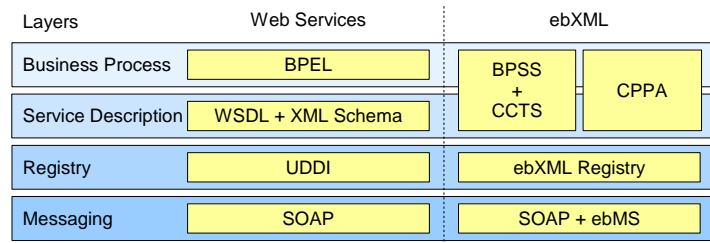


Fig. 5. Languages of Web Services and ebXML compared

and a body with the payload. Since SOAP does not support reliability or secure messaging, other specifications such as WS-Reliability and WS-Security have to be used to extend it.

ebXML uses its own messaging protocol, namely *ebXML Messaging Service* (ebMS). It is an extension of SOAP messages with attachments including features for security, digital signatures, non-repudiation, and reliability. As such it provides more features, however the Web Service Protocol Stack is growing continuously and will provide comparable features in future.

On the registry layer, UDDI competes with the ebXML Registry. A registry acts as a focal point for information required by e-business. It is used to discover trading partners and their capabilities, classifies the information stored in it, and allows the registration and discovery of documents and service descriptions. UDDI has been initially developed in 2000 by Microsoft, Ariba, and IBM to create a standard for service description and discovery.

The ebXML Registry Specification is part of the ebXML initiative. While Web Services can be used without UDDI (it is the prevailing form of interaction today) the ebXML Registry is a central component of the ebXML Architecture (see figure 3). It serves as a storage hub and discovery facility for many artifacts involved in e-business processes using the ebXML framework. In contrast to UDDI that offers only artifacts for discovery the ebXML Registry also stores documents required for collaboration (CPP, CPA and BPSS). Because the two registry types focus on different aspects, their information models and architectures vary[4]. However, version 3 of both technologies brought the two registry closer; and in 2005 Sun announced the Sun Service Registry that combines both technologies and proved their interoperability.

The layers describing the service and the business processes are hard to compare. Web Services are described using WSDL that defines the operations (or methods) exposed by a service and its location. Although one WSDL file may contain different services, they cannot be linked together to form a business process. This task can be fulfilled using BPEL describing business processes consisting of single Web Services. The ebXML architecture runs another path: it first requires the processes and the documents to be defined and then offers

a way to negotiate the artifacts of two trading partners to form an agreement. This agreement can then be used to configure an ebXML compliant system. While BPEL depends on the use of WSDL to describe the outgoing and incoming messages, ebXML does not follow a layered approach to enhance a service with business logic in the large. This concept shows once more that ebXML is aimed to be rather a holistic framework than a conglomeration of different specifications.

The Business Process Specification included in ebXML provides business-level semantics. As such it has no equivalent in the Web Service Stack. The same applies to the Core Components that can be reused to assemble business documents [10]. Another language associated with the ebXML Core Components is the *Universal Business Language*, an effort to define a library of standard electronic XML business documents. UBL was born out of the desire of some ebXML participants to define a payload format for ebXML. However, UBL can as well be used in Web Services without ebXML.

As already stated above a Web Service implementation for e-business is more service-based, while ebXML uses a contract-based approach (figure 6). Usually, a Web Service is described, provided and published by one single entity (unilateral). This service is then found and bound by the requester. The contract-based ebXML approach is bilateral (or even multilateral): two trading partners negotiate their profiles and come to an agreement. Based on this contract they execute their common business process.

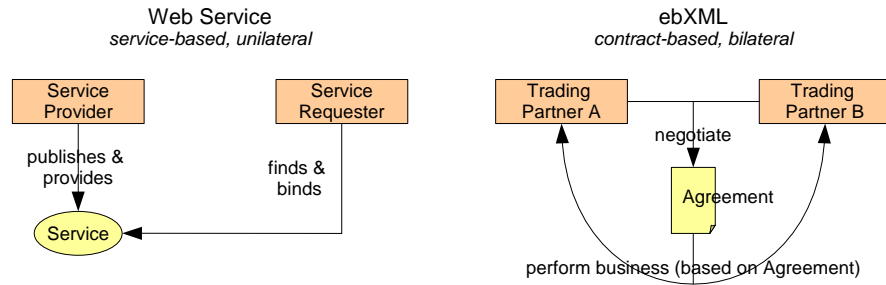


Fig. 6. Service- vs. Contract-based Approach

This distinction applies especially to a Web Service implementation without BPEL. BPEL enriches a simple Web Service with business process logic. However, BPEL is not a language to express collaboration using choreography: It is used to describe business processes whose activities are implemented using Web Services. Contrariwise, ebXML's contract-based approach allows two enti-

ties to collaborate - but each entity autonomously. The trading partners follow a contract, the CPA, to conduct e-business.

The next distinction is about coupling². On the one hand, basic Web Services tend to be tightly coupled to the backend implementation. Thus, switching between different web service engines may cause compatibility issues. This aspect is again influenced by the amount of different standards used and the degree of their maturity. There are many different combinations of standards and versions possible that all fit some special needs. Not all standards are supported completely by different vendors. On the other hand, a Web Service is loosely coupled with the callers: there is no fixed model on how a service has to be used. Looking at ebXML, coupling is different because ebXML focuses on business processes and documents. Using ebXML the implementation is tightly coupled to the business processes that change seldom. On account of this coupling with backend systems is loose, because ebXML conformant products are able to generate the necessary documents out of business process specifications. Given the fact that ebXML standards have been approved by the International Standards Organization (ISO), long-term vendor support for ebXML is very likely.

2.3 Comparison using eCo Layers and Workflow Aspects

In [1] the authors proposed a comparison of the two approaches using a framework based on the eCo Framework and a workflow model proposed by Rausch-Schott. The eCo Framework defines seven layers, whereby only four layers are used for comparison (figure 7). The *Service Layer* describes the services and their interfaces provided and used by the trading partners; The *Interactions Layer* is about the type of interactions behind a service and the message types used; The *Document Layer* describes the document types that are used within the messages; finally the *Information Items Layer* describes the types of information items that may be used in document types.

The Web Service approach relies on XML Schema to specify documents and information items, uses WSDL and BPEL to describe the services and WSDL in combination with other WS-* specifications for the interaction layer (WS-Policy, WS-Security, WS-Routing, WS-Reliability, etc.; [1] discusses WSSP, the Web Services Security Policy, at this point).

The ebXML approach uses XML Schema as well namely to specify the information items. In contrast, it uses components specified in the Core Components Technical Specification (CCTS) that are reused to assemble business documents. These business documents are described using BPSS (Business Process Specification Schema) that offers as well a methodology to specify collaboration and services. Thus the interaction layer and the service layer employ BPSS and CPPA (Collaboration Protocol Profile and Agreement Specification). BPSS is responsible for the generic aspects, while CPPA provides technology and business details.

² compare [13]

Layers	Web Services		ebXML	
Services	WSDL	BPEL	BPSS	CPPA
Interactions	WSDL	WS-*	BPSS	CPPA
Documents	XML Schema		BPSS	CCTS
Information Items	XML Schema		XML Schema	CCTS

Fig. 7. Languages of Web Services and ebXML compared using eCo Layers
Source: [1], slightly adopted

The other model used in [1] for comparison is a workflow model by Rausch-Schott modified to meet the conditions of B2B protocols instead of workflows. The covered aspects are *functional, operational, behavioral, informational, organizational, organizational, transactional, causal* and *security* aspects. These aspects refine the comparison of the four layers where they are relevant.

For example, on the *Information Items Layer* there is a difference regarding the *causal aspect* that is concerned with reason behind the design of information item. Web Services use solely XML Schema to describe this aspect, whereas ebXML offers Core Components used to build information items with aggregation. Core Components describe semantics and are independent of business contexts and thus may be reused widely. However, for the *informational aspect* there are standardizations efforts such as the Universal Business Language (UBL) defining standard business documents that may be used within both Web Services and ebXML.

On the *Interactions Layer* many differences can be found. For example, regarding the *behavioral aspect* that addresses the control flow during an interaction, ebXML has an enhanced request/response model compared to Web Services. ebXML allows to use acknowledgments signals indicating receipt or acceptance of a message and it supports to define validation or timing constraints. Comparing the *security, transactional* and *operational aspects*, ebXML is once ahead. Using Web Services only integrity, authenticity and confidentiality of messages are supported, while ebXML also supports authentication and non-repudiation in respect of the role. ebXML has limited support for transactions by using the atomicity property. Web Services may use WS-Transaction specification that can be combined with BPEL processes to support transactions.

On the *Service Layer* ebXML distinguish from Web Services by providing bilateral services (between two entities) and multilateral services (supporting many entities). Web Services follow an unilateral approach where the service functionality is described from the provider's point of view.

The *functional aspect* addresses the work provided by the service and its decomposition. A BPEL enabled Web Service decomposes a service into interactions (invoke, receive and reply). The combination of these constructs form the service. ebXML supports pre- and postconditions and decomposes a service into sub-services and interactions. The *organizational aspect* concerns the roles that take part in a business transaction. It is about identification, authorization and obligations of a role in a service. Other than Web Services, ebXML supports multilateral services; but both, BPEL and ebXML, allow to define the roles of a service.

The *informational aspect* is about the data used in a service type. By using Web Services, BPEL supports variables, assignments, data flow between variables and interactions, expression languages using e.g. XPath, and correlation sets used to identify different messages that belong to one service instance. ebXML does not support variables or the definition of data flows. Though, the notion of message correlation is not relevant because ebXML handles the correlation of message by an underlying layer.

The *behavioral aspect* describes the dynamics of a service type, for example using states, control flow, conditions, and exceptions. For Web Services BPEL provides structured programming constructs (conditions, loops, sequential and parallel execution) that may be yield atomic and composite states. Furthermore, BPEL supports exceptions and compensation using *try-catch-throw* semantics. A scope may be defined for an activity, multiple fault handler assigned, and compensations handlers added that compensates the entire scope. In ebXML the concept for specifying behavior is defined for the service as a whole and is similar to those of UML's activity diagrams. Activities may be defined as interactions or sub-services.

2.4 Similarities and Interoperability

Despite the differences discussed in the preceding sections there are as well many aspects of ebXML and Web Services that can be seen in combination. A number of experts say that the two technologies are complementary: for example, ebXML uses Web Service protocols such as SOAP. At that time when ebXML was developed, Web Services did not really exist. The ebXML initiative then tried to incorporate Web services components in ebXML. But there is a conflict whether supporting ebXML or Web Services. The two technologies do not interoperate: if a trading partners sends an ebXML message to another one, it has to support ebXML or it will not work. The same applies to Web Services [5].

Related to the ebXML vs. Web Services debate, some people argue, that ebXML helped to accelerate the Web Service evolution. At the end of ebXML's 18-months initiative, the modular suite of specifications has been available. Now, after BPEL and other WS-specifications, Web Services have cached up and now support many features originally unique to ebXML.

Business Processes and Documents. Independent from the B2B framework chosen, the knowledge about a company's business processes has to be acquired.

Therefore it is recommended to use a well-defined and consistent methodology such as UMM, the UN/CEFACT Modeling Methodology. UMM is a methodology for capturing business process knowledge, it is based on UML (Unified Modeling Language) and uses ideas from the MDA (Model Driven Architecture). The advantage of this approach is to create a model independent from the concrete technical implementation and the platform used. This model contains a machine interpretable description that can be translated to configure a specific e-business system.

The ebXML Business Process Specification Schema is based on concepts introduced by UMM. On the one hand a UMM business collaboration model can be mapped to BPSS and so it can be used to configure an ebXML enabled system [7]. On the other hand also other e-business systems can benefit from UMM because it is possible to transform parts of a UMM model to BPEL [8] and workflow systems.

The preceding paragraphs show how UMM can be translated to other business process specifications. Such a business process consists of a set of business transactions expressed as an exchange of business documents. While UMM can be used to describe the process, the Universal Business Language (UBL) can be used to compose business documents from it. UBL is a free library of standard electronic XML business documents such as purchase orders and invoices. It can be plugged directly into existing business systems. Despite the fact that ebXML and Web Services are not interoperable, UMM and UBL can be used to share knowledge about business processes and documents in both environments.

2.5 Vendor Support

Emerging technologies always depend on the availability of implementations. This applies in particular to e-business approaches. Currently, EDI (UN/EDIFACT and EDI-X12) is still the most important technology on the market. There always existed better technologies than EDI. However, it succeeded despite its risks and disadvantages. Finally, it matters if a technology is able to yield competitive advantages by lowering operation costs and improving tactical and strategic decision making.

Maybe, one of the new technologies, such as ebXML and Web Services, will once take on EDI's position as the leading B2B framework. Many companies are still waiting to see which technology becomes widely accepted before making investments at the moment. However, the standardization organizations try to increase operability of their approaches and providers try to integrate the new technologies in existing solutions.

Regarding the current situation on the worldwide information technology sector, the success of a new technologies depends also on the politics of the major players. Almost every large IT company participate in the development of Web Service standards including IBM, Microsoft and Sun Microsystems. Sun also strongly advocates ebXML but Microsoft, maybe in opposition to Sun, does not support it. Instead Microsoft provides its own BizTalk server. IBM also participated in ebXML but now focuses more on the development of Web Services

as many others do. For example, the BPEL4WS Specification was developed by IBM, BEA Systems, Microsoft, SAP and Siebel Systems.

According to [6], *Systems such as ebXML have reached a point where they are now ready for full scale deployment.* While ebXML has its strengths mainly on the markets in Europe and Asia, Web Services are already deployed world wide.

In [15] ebXML solutions are reviewed. The products are divided into four categories, summarized in the following:

- **Enterprise integration solutions** that support ebXML based messaging transport and partially ebXML BPSS for the business process execution. Vendors of this category are among others: BEA Systems, Fujitsu, Sonic Software, Sun Microsystems, Sybase and TIBCO.
- Traditional **EDI-style integration solutions** focus primarily on ebMS messaging service to support ebXML-based messages. These categories includes products from Innovis, Seeburger AG, Adobe and freebXML.
- Among this there are some providers of **ebXML Registry solutions**. In this category, there are pure ebXML Registry providers and those providing existing repositories in combination with ebXML compliant interfaces. Some providers: ebXMLsoft, freebXML, Sun Microsystems.
- The fourth category consists of modeling tools managing business processes and artifacts for ebXML implementations. The tools use a model-driven approach and support BPSS, CPA and CCTS artifacts.

The most comprehensive ebXML products are commercial products but there are as well open source products available. The freebXML initiative provides a central site for sharing free ebXML code, applications and development experience. For example, it includes the Hermes Message Service Handler (an open source implementation of an ebXML messaging server), ebxmlr (the OASIS ebXML Registry Reference Implementation) and freebXML BP (a tool designed to create generic and domain specific Business Process Specifications).

Regarding Web Services the market is much larger compared to ebXML. On the one hand there exist tools, specialized on specific parts of the Web Service stack, on the other hand more and more products become available offering all-in-one solutions for Web Services including BPEL execution. Since Web Services are needed for many different reasons, vendors from various domains have filled the gap. First, there are implementations of specific standards, such as SOAP or WSDL. As these basic standards are important for nearly every Web Service product, there are implementations for most programming languages available. For example, Microsoft provides the .NET Remoting Framework for this purpose and for Java the Apache Axis project is the prevalent open source SOAP implementation. Also, many established products have recently been enriched with Web Service interfaces to increase interoperability with other products. For example, databases support XML and allow calling external Web Services

from stored procedures, and ERP systems use Web Services to integrate data from external systems. Even office products and simple desktop tools integrate Web Services. Finally, there are all-in-one Web Service solutions supporting the basic standards and integrating as well many enhanced standards to provide full support for e-business scenarios. This category is occupied by the big players but also by some innovative smaller companies. The most known products supporting Web Services with BPEL include BEA Weblogic, IBM Websphere, Microsoft BizTalk, Oracle BPEL Process Manager and SAP NetWeaver.

3 Conclusion

Business-to-business electronic commerce is typically an automated process between trading partners. One of the most well known and established B2B standards is still UN/EDIFACT. New XML-based standards promise reduced costs, increased flexibility and higher business value. This work compares ebXML and Web Services, in the combination of SOAP, UDDI, WSDL and BPEL. The comparison shows that both technologies have their own advantages. The holistic approach of ebXML is an all-in-one solution and places emphasis on the business and its processes. It is ready for deployment and during the first years of experience the specifications have matured. In contrast, Web Service specifications have been usually developed independent of each other, leading to a sometimes incoherent but flexible technology. It has good vendor support since all major players on the IT market engage in it.

To conduct e-business specific requirements such as security and reliability have to be met. Currently, the Web Services protocol stack does not fulfill all needs: some specifications are not widely adopted and some are still drafts. In this field ebXML has still a slight advantage over Web Services. In future it can be expected that Web Services will continue to absorb the strengths of ebXML. In some areas interoperability brings the two approaches together, for example, by using the Universal Business Language to describe business documents or by following the UN/CEFACT Modeling Methodology to acquire knowledge about business processes. After all, the main difference will remain: ebXML's top-down design approach will continue to yield different results and addresses different audience than the bottom-up approach of Web Services.

References

1. Martin Bernauer, Gerti Kappel, and Gerhard Kramler. Comparing WSDL-based and ebXML-based Approaches for B2B Protocol Specification. In *Proceedings of the 1st International Conference on Service-Oriented Computing (ICSOC-03)*, pages 225–240. Springer-Verlag, 2003.
2. David Booth, Hugo Haas, Francis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard. Web Services Architecture. <http://www.w3.org/TR/ws-arch/>, February 2004.
3. Nicholas Chase. Introduction to ebXML. <http://www-128.ibm.com/developerworks/edu/x-dw-xebxml-i.html>, 2002.

4. Joseph Chiusano. UDDI and ebXML Registry: A Co-Existence Paradigm. <http://www.oasis-open.org/archives/regrep/200304/pdf00000.pdf>, March 2003.
5. Paul Desmond. Butting heads over B2B. <http://www.networkworld.com/power/2003/1222b2b.html>, 2003.
6. European Commission. The European e-Business Market Watch 2005: e-Business Interoperability and Standards. http://www.ebusiness-watch.org/resources/documents/TR03_Interoperability_2005_web.pdf, 2005.
7. Birgit Hofreiter and Christian Huemer. ebXML Business Processes - Defined both in UMM and BPSS. In *Proceedings of the 1st GI Workshop XML4BPM - XML Interchange Formats for Business Process Management*, 2004.
8. Birgit Hofreiter and Christian Huemer. Transforming UMM Business Collaboration Models to BPEL. In *Proceedings of the OTM Workshop on Modeling Inter-Organizational Systems*, 2004.
9. Birgit Hofreiter, Christian Huemer, and Wolfgang Klas. ebXML: Status, Research Issues, and Obstacles. In *RIDE '02: Proceedings of the 12th International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems (RIDE'02)*, page 7, Washington, DC, USA, 2002. IEEE Computer Society.
10. Saket Saith. ebXML The new global standard for Electronic Business . <http://www.itsc.org.sg/events/ietc0305/ebxml%20-%20Saket%20Saith.pdf>, March 2005.
11. Aphrodite Tsalgatidou and Thomi Pilioura. An Overview of Standards and Related Technology in Web Services. *Distrib. Parallel Databases*, 12(2-3):135–162, 2002.
12. UN/CEFACT, OASIS. ebXML Website. <http://www.ebxml.org/>.
13. David Wai-lok Cheung. A Closer Look at ebXML. http://www.cecid.org/Event/20030530%20-%20eb_deployment_ITSD.pdf, 2003.
14. Dale Waldt. ebXML Web Services & EDI. http://www.ebxml.org/presentations/ws_ebxml_london_may03.pdf, 2003. XML Europe 2003, London.
15. David R.R. Webber. ebXML Forum's Software Resources and Products Guide. http://www.ebxmlforum.net/articles/ebfor_SoftwareProducts.html, September 2004.