ebXML-Based Electronic Business Interoperability Framework and Test Platform

Chapter · May 2011

DOI: 10.4018/978-1-60960-485-1.ch018

9 authors, including:

Peng Liang
Wuhan University

Bing Li
Wuhan University

Zaiwen Feng
Huazhong Agricultural University

Tao Peng
Wuhan University of Technology

ebXML-based Electronic Business Interoperability Framework and Test Platform

Peng Liang, Keqing He, Bing Li, Zaiwen Feng, Tao Peng,
Shuai Chen, Yu'e Pan, Wei Qing, Dan Song
State Key Lab of Software Engineering (SKLSE)
Wuhan University, China

ABSTRACT

Electronic Business (e-Business) system is a typical heterogeneous information system, which is concerned with various data types, communication protocols, business processes, and exchanged business contents between these systems. The interoperability assurance between e-Business systems is a key issue for the implementation and application of a seamless e-Business environment around the world. The ebXML (e-Business XML) initiative is one of the most promising standards in the field of e-Business, whose mission is to provide an open, XML-based infrastructure to enable the global use of e-Business information in an interoperable, secure, and consistent manner by all trading partners, targeting to the B2B market. The ebXML specification itself can not guarantee the interoperability between various ebXML-based B2B implementations since different organizations may understand and implement ebXML specification in a different way. With more and more e-Business systems supporting ebXML, the interoperability assurance among them becomes even challenging. This chapter focuses on how to address the interoperability issues in ebXML-based solutions, and how to achieve the interoperability assurance through testing in a cost-effective way. This chapter proposes a two part solution to address these issues: an interoperability framework for ebXML-based e-Business solutions, and a platform for automatic interoperability test. The benefit of our approach is that it provides both a conceptual interoperability framework and interoperability test infrastructure for ebXML-based B2B solutions. This allows industrial organizations to introduce the ebXML-based interoperability framework out of the box, with a readily-used interoperability test infrastructure to be incorporated to their own test tools and environment.

1. INTRODUCTION

The interoperability between Electronic Business (e-Business) solutions is a key issue for the implementation of a seamless e-Business environment around the world, especially in a B2B[i] context, but the complexity of the heterogeneities of B2B systems is too great for technical integrations only (Kajan & Stoimenov, 2005). We argue that the interoperability assurance (IA) between B2B e-Business systems should be guaranteed through a systematic approach, including the technical, human, and domain aspects (Liang, 2006). In this chapter, we focus on the technical aspect of IA through interoperability test and general technical solution of IA for B2B systems without considering other two aspects since they are mostly involved with concrete B2B systems and business cases, which are too broad to be investigated in one chapter.

The e-Business systems are implemented based on certain e-Business specification, e.g., Electronic Data Interchange (EDI) standard, consequently the interoperability of the e-Business solutions are constrained by the e-Business specification, on which the e-Business solutions are implemented. With the popularization of Web service techniques, the XML-based e-Business specifications/standards are becoming the main stream in e-Business environment, e.g., ebXML (Gibb & Damodaran, 2002), cnXML (Xu, *et al.*, 2005), etc. The standard specification itself can not guarantee the interoperability between B2B e-Business solutions since it only provides a general reference framework for system implementation without constraining the technical details of the implementation. This makes the IA of B2B systems quite difficult to achieve even though the B2B systems are implemented based on the same e-Business specification (e.g., ebXML). The appearance of various B2B e-Business specifications makes this problem even worse (Liang, *et al.*, 2005). In a technical perspective, the IA scenarios of B2B systems can be classified in two types: the interoperability between the B2B solutions based on the same specification (e.g., ebXML); and the interoperability between the B2B solutions based on the different specifications (e.g., ebXML, RosettaNet, or cnXML). For the first scenario, a method should be provided to validate the IA of e-Business solutions that are based on the same e-Business specification; while for the second scenario, an IA framework should be provided to guide the IA realization of e-Business solutions that are based on different specifications.

XML is a fundamental description language and de factor standard for data portability, and Web service framework is specially designed for e-Business applications (Kajan & Stoimenov, 2005). In this chapter, we propose a general interoperability framework for the IA of B2B systems, targeting to the XML-based e-Business specifications. The methods for addressing the issues of two types of IA scenarios in B2B systems are presented. In order to show the effectiveness of our proposed framework and methods, we demonstrate a concrete case of IA on B2B solutions based on ebXML specification through the interoperability test.

The rest of this chapter is organized as follows. Section 2 provides a general introduction on ebXML, the use case scenario, and basic concepts of ebXML. Section 3 presents the interoperability framework in three levels with detailed methods for achieving the interoperability in each level. Section 4 presents the automatic interoperability test platform SKLSE-ebIOT for the interoperability test at messaging level with test cases and results. Section 5 briefly reviews the related work on the interoperability of Web services-based systems and B2B systems, followed by conclusions and future work directions in Section 6.

2. EBXML BACKGROUND

Electronic Business using eXtensible Markup Language, commonly known as e-Business XML (ebXML), is a family of XML-based standards sponsored by OASIS and UN/CEFACT whose mission is to provide an open, XML-based infrastructure that enables the global use of e-Business information in an interoperable, secure, and consistent manner by all trading partners.

A classical ebXML use case scenario is presented in Figure 1 (OASIS, 2001b) for a better understanding of ebXML usage in e-Business, especially in a B2B context. The numbers in the figure denote the steps of interaction between two organizations, and a series of specifications are defined to fulfill the functions in this scenario. The message specification of ebXML is called ebXML messaging service, short for ebMS (OASIS, 2002a). ebXML message defined by ebMS specification is used to transport all the messages between business organizations (e.g., message exchange between Company A and B in Step 5). CPA (Collaboration-Protocol Agreement) specification is used to define and simulate the business context of business collaborations. The CPA files are XML-based documents specifying a trading agreement between trading partners, and describe the formal relationship between two parties, including information like identification information e.g., the unique identifiers for each party and their roles within the trading relationship, communication information e.g., the protocols that will be used when exchanging documents, etc. which is shown in Step 5 in Figure 1. ebXML BPSS (ebXML Business Process Specification Schema, ebBPSS) specification provides a schema on how a business process can be defined in ebXML, so that business transactions can follow (Step 6). These (ebMS, CPA, and ebBPSS) are three fundamental specifications in the family of ebXML specifications.

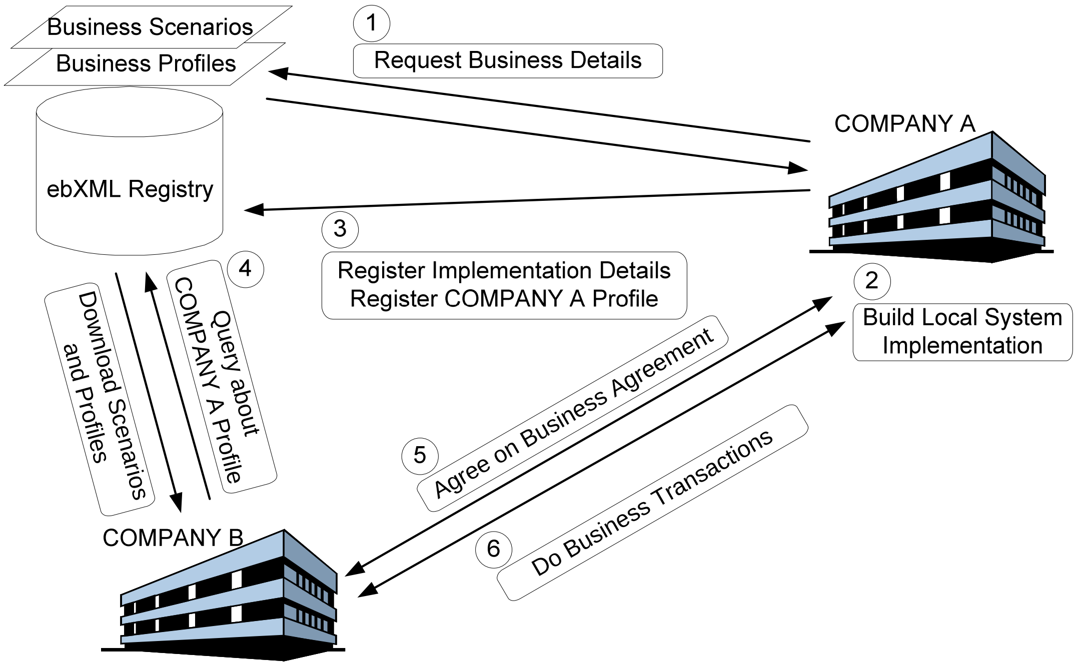

Figure 1. ebXML use case scenario between two companies conducting e-Business

3. INTEROPERABILITY ASSURANCE FRAMEWORK FOR B2B SYSTEMS

e-Business system, especially the B2B system, is a complex system composed by many heterogeneous components. For the interoperability of such complex system, we can classify the interoperability issues in different levels. The structure of the IA of B2B systems is comparable to the seven layers in the network protocol OSI (Open System Interconnection). The IA of each upper level (e.g., application layer in OSI) should be guaranteed based on the IA of the lower level (e.g., presentation layer in OSI). According to the technical architecture specification of ebXML (OASIS, 2001b), which provides a common technical architecture for the e-Business systems, there are basically three layers in an e-Business system: data, semantic, and process layers. Considering this technical division, we argue that the IA of B2B systems should follow these layers division as well. The proposed IA framework for B2B systems is shown in Figure 2. The data layer corresponds to the messaging service in B2B transactions; the semantic layer corresponds to business terms exchanged between B2B systems; and the process layer represents the business processes to achieve a business goal. Similar to the OSI structure, the lower level in this interoperability framework provides support to the interoperability at a higher level, for example, the IA of messaging service provides support to the interoperability of business term communications and business process transactions by exchanging ebXML messages.

The basic functions of each layer in an XML-based e-Business specification are described below. The messaging service specification defines the basic format and delivery mechanism of business messages, such as *sending order message*. This level is the lowest layer in the B2B

Interoperability framework, consequently it is the fundamental layer for the IA of the whole B2B system. The business term specification defines the commonly used/exchanged business terms and the format to specify these terms, such as *Order*, *Invoice*, and *Address* etc. The communication of business terms can only be realized by the support of messaging service. The business process specification defines how a business process can be specified, including a set of elements and their relationships. The ebXML business process specification focuses on the message sequence in a fixed order, such as *Ordering-Shipping-Invoicing* process. The business process employs the business terms that are defined according to business term specification, so this level locates in the highest layer of the B2B interoperability framework. The detailed description of the IA framework with related methods for addressing the IA issues in the three layers is presented in subsections below.

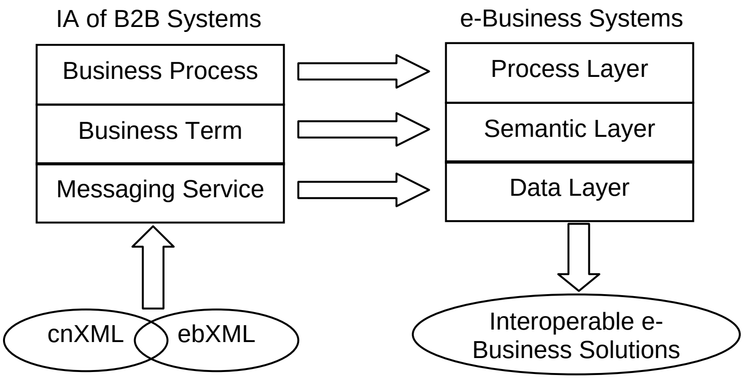

Figure 2. Interoperability framework and layers of e-Business systems (Liang, et al., 2005)

3.1 Data/Messaging Level

As discussed above, the IA scenarios of B2B systems can be classified in two types: the interoperability between B2B solutions that are based on same specification (e.g., ebXML); or based on different specifications (e.g., ebXML and cnXML[ii]). For the first scenario, we introduce interoperability test to guarantee the IA through standardized test based on the e-Business specification. For the second scenario, we create a structural mapping between two different e-Business specifications (i.e., structural e-Business messages), and then perform the interoperability test based on the mapping. We detail these two IA scenarios and their related methods in the following section.

3.1.1 IA of B2B Systems based on Same e-Business Specification

We demonstrate this IA scenario through the interoperability test of ebXML-based B2B systems. In this layer, the ebXML message defined by ebMS specification is an ebXML umbrella that provides a secure and reliable SOAP-based (W3C, 2000) transport protocol to the ebXML architecture. The core part of ebXML message is a SOAP message with additional elements for reliable messaging and secure messaging, etc. The interoperability test items at the messaging level are generally defined in two steps: (1) select basic interoperability levels from ebMS 1.09 Conformance Clause[iii] (OASIS, 2002b) defined by OASIS IIC (OASIS ebXML Implementation, Interoperability and Conformance) Technical Committee; (2) discuss with domain experts to refine (add/modify/remove) the interoperability test items according to the functional requirements in a real business environment, which are most important for the realization of a B2B system. Finally the ebMS interoperability test items are classified in six levels (T1-T6) as shown in Table 1, and each test level is responsible for the IA of certain business function.

Test level	Functional description
T1	One way messaging
T2	Synchronized message reply
T3	SSL (Secure Sockets Layer) secure messaging
T4	XML signature messaging
T5	Reliable messaging
T6	Error messaging

Table 1. ebXML interoperability test levels at messaging level

An interoperability test procedure is specified in order to guide test engineers to perform the test in a uniform way. According to the test levels classification in Table 1, two test process models are defined, "*Single Transmission*" and "*Echo Back*" model. Figure 3 illustrates the "*Echo Back*" test process model. In this model, message receiver should send back an acknowledge message to the sender (Step 5) when receiving a message. On the contrary, in "*Single Transmission*" test process model, the acknowledge message is not needed. Except that, all the other steps are the same in both "*Single Transmission*" and "*Echo Back*" test process models.

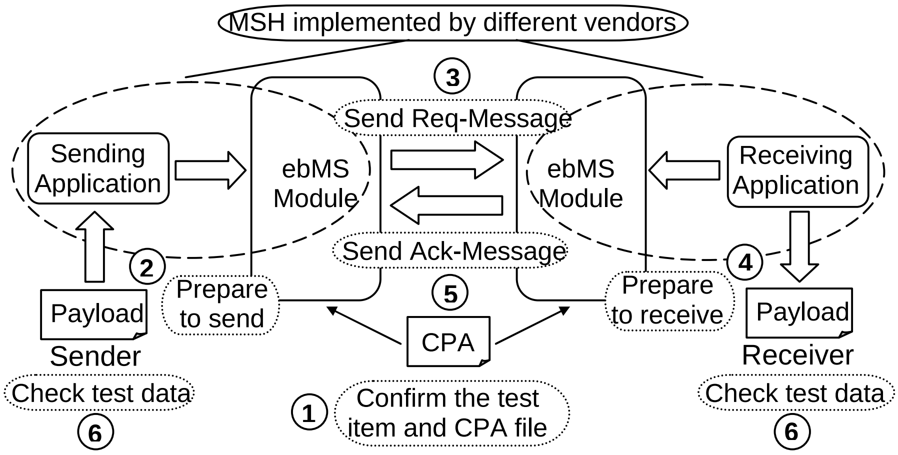

Figure 3. ebMS interoperability test process model "Echo Back"

In Step 1, both test parties confirm the content of the CPA file (Collaboration-Protocol Agreement) used to simulate the business context of a certain test item; Step 2, sender attaches the message payload file to the message, and prepares to send the message to receiver; Step 3, sender sends out the message; Step 4, receiver receives the message and extracts the message payload from the message; Step 5, receiver sends out the acknowledgement message; Step 6, both parties check the test results by following the test evaluation rules. In Figure 3, the ebMS module used for ebXML message sending/receiving is named as Message Service Handler (MSH). This test procedure and test results evaluation can be performed manually by test engineers, e.g., in (Liang, 2003), and it can also be done automatically by a test platform, e.g., in (Chen, 2005). The SKLSE-ebIOT is an automatic test platform for this purpose, and it will be further described in Section 4.

The interoperability test evaluation rules define the criteria for determining whether the test results satisfy the interoperability requirements of a certain test item. The evaluation rule is defined for each test item, and it includes general information of test item and test inspection items for both test parties. As mentioned above, these rules can be evaluated manually by test engineers, or automatically by a test platform. We demonstrate below the interoperability test evaluation rules by an example, the test item T1-1-1 in T1 level of ebMS interoperability test.

Test item ID	T1-1-1
Test item name	One way messaging
Test item overview	Sender sends a message with one payload to Receiver
CPA template	SKLSE-CPA-T111 (named and prepared by test organization)
Message sequence	As shown in Figure 4

Table 2. General information of test item T1-1-1

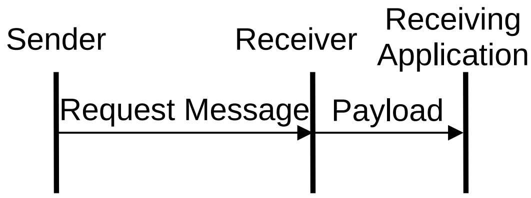

Figure 4. Test message sequence of T1-1-1

Test inspection items are specified for the evaluation by *Sender* and *Receiver* respectively. In test item T1-1-1, for *Sender*: (1) Inspect that there is no error occurred internally, no error message sent, no HTTP failure result code received. For *Receiver*: (1) Inspect that there is no error occurred internally, no error message sent, no HTTP failure result code received; (2) Extract the payload in the message and compare it with the payload sent by sender, and it should be exactly the same (size, filename, and content); (3) Inspect that the value and the type attribute value of the <*Service*> element in the message is the same as defined in the CPA file; (4) Inspect that the value of the <*Action*> element in the message is the same as defined in the CPA file; (5) Inspect that the value of the <*CPAID*> element in the message is the same as defined in the CPA file. The decision of the test item, i.e., whether the test item is successful or not, is determined by a full check of these inspection items. Any failed inspection item will result in the failure of the test item.

3.1.2 IA of B2B Systems based on Different e-Business Specifications

The IA of B2B systems that are based on different e-Business specifications is more complex and difficult to achieve than the previous IA scenario. At the messaging/data level, the elements of a business message are specified according to specific XML schema, e.g., ebMS schema. These structural/syntactical information (i.e., ebMS schema defined in ebMS specification) is comparably stable than the semantic information (e.g., business terms), whose meaning are variable depending on specific business context. We propose that a structural mapping between different message specifications should be constructed before conducting the interoperability test at the messaging level. A message translator can be employed for a bidirectional transformation between heterogeneous message formats in the interoperability test, e.g., between ebMS and cnXML messaging service (cnMS). For XML-based e-Business specifications, XSLT (XSL transformations) can be used to perform the transformation.

We show a concrete example of structural mapping of e-Business message specifications between ebMS and cnMS. Messaging service specification in B2B system defines the message format and elements, which are described and formulated by XML schema. The message format of cnMS and ebMS are both based on SOAP specification (W3C, 2000), including header container and payload container. In Figure 5, the XML schema comparison between cnMS and ebMS is demonstrated with a structural mapping between their elements. The detailed analysis of this mapping is presented below. Note that, this mapping has to be done manually by experts or a technical committee, but not automatically. This mapping will facilitate automatic translation between the messages in different message schemas (e.g., cnMS and ebMS) through e.g., XSLT.

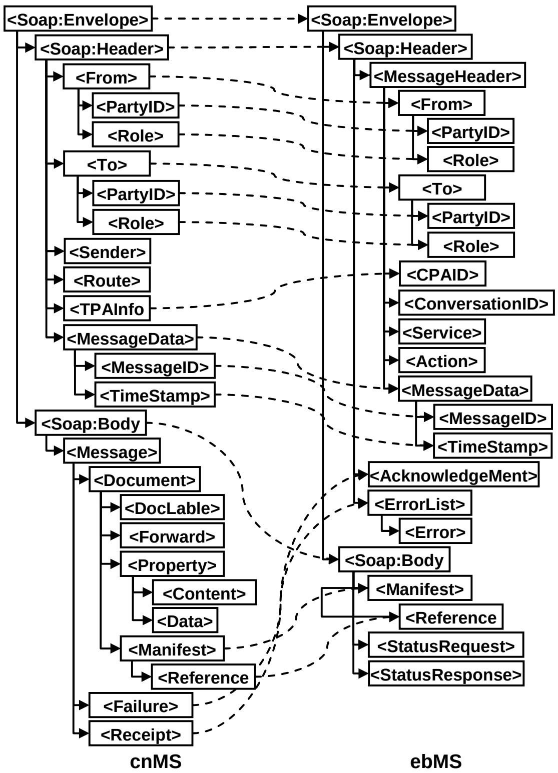

Figure 5. Structural mapping of XML schema between cnMS and ebMS (Liang, et al., 2005)

In the header container of XML schema, both the cnMS and ebMS include a <*Header*> element and <*Body*> element. Some elements in cnMS and ebMS can be easily mapped since they have the same name, such as <*From*>, <*To*>, and <*MessageData*>. Some elements have the same meaning without sharing the same name, such as elements <*Failure*> and <*ErrorList*>, and <*Receipt*> and <*Acknowledgement*>.

As shown in the structural mapping result between cnMS and ebMS in Figure 5, the XML schemas of the message format are quite similar between cnMS and ebMS. The similarity provides the basic for the transformation between cnMS and ebMS using XSLT. A translator, which stands between two B2B systems for translating, sending, and receiving messages based on the structural mapping, is responsible to deal with the different message format. Due to the inherent heterogeneity of e-Business specifications proposed and defined by various organizations, it is not possible to guarantee that all the elements in one message format can be mapped to the other or vice versa (e.g., <*Forward*> element in ebMS can not be mapped to any element in cnMS, since ebXML messaging service does not support message forward function). But these unmapped elements will not affect the interoperability test between B2B systems since the quality of IA can be evaluated/quantified by the IA test levels that two B2B systems can both satisfy as shown in Table 1, e.g., two B2B systems can only satisfy the T1 and T2 test levels, but can not fulfill the T3. It is also worthy noting that all business functions that are based on <*Forward*> element will fail in the B2B interoperation between ebXML and cnXML-based B2B solutions, but this is common in B2B interoperation that not all the business functions are interoperable (i.e., partial interoperability).

3.2 Semantic Level

As defined by (Kajan & Stoimenov, 2005), semantic interoperability in B2B systems is the ability to share information at the application level, without knowing or understanding the terminology of other systems. In order to achieve semantic interoperability, systems should be able to exchange data in a way that the precise meaning of the data (the semantics) can be translated by any system into a form it understands. Business terms are core content exchanged between two business parties using messaging service, and these business terms are further employed in business processes, e.g., *Order*, *Invoice*, and *Address* etc. In the field of semantic interoperability, ontology and ontology mapping have been recognized and widely used to address the issues of semantic interoperability in information systems, especially in the Web-based systems (Doerr, 2003).

The ontology mapping approach provides a useful and flexible way to address the interoperability issues at the semantic level. The business terms (i.e., local-ontology) in a specific e-Business system can be mapped/translated to the terms of other systems, or mapped to a top-ontology, e.g., Universal Business Language (UBL) (OASIS, 2008), which acts as a semantic bridge between heterogeneous local-ontologies. The weak point of ontology mapping for the semantic interoperability in B2B systems is that it requires considerable mapping effort by domain experts when dealing with a large number of business terms (e.g., the number of concepts in medicine can easily go up to 10^6). In the interoperability framework for B2B systems, we propose a classification-based method for ontology mapping (Pan, *et al.*, 2007), which can significantly alleviate the workload of ontology mapping at the semantic level of B2B systems.

The main idea is that: in B2B systems, most of business terms are domain-oriented. Each business term (i.e., the concept in an ontology) belongs to certain domain or sub-domain, such as *banking transaction service*, *billing service*, and *transportation service* etc. The common structure of ontology is a tree structure with hierarchy relationships, especially the *subclassOf* (subsumption) relationship. Domain experts can decompose the ontology to be mapped into smaller classification trees, and try to create the ontology mapping assertion between the root concepts of classification trees from two ontologies, and then the semantic similarity calculation between heterogeneous business terms (concepts) will be limited to the concept pairs within the mapped classification trees. In this way, the ontology mapping effort can be reduced dramatically, especially for the B2B systems. We explain this method by an example as shown in Figure 6. The business terms *trade-name* and *article* are two concepts from two different ontologies O_1 (Computer Online Store) and O_2 (Property Management System). When domain experts create the ontology mapping assertion between these two root concepts (i.e., *trade-name* is equal to *article*), the ontology mappings of their sub-concepts (e.g., *PC* and *workstation*) will be limited within these two classification trees, which means that the concept in one classification tree (e.g., *PC* as a sub-concept of *trade-name*) will try to map to the concept in another classification tree without traversing the huge number of concepts outside the classification tree. This method provides a good balance between human effort by domain experts and automatic mapping support by semantic similarity calculation tool with decent ontology mapping results. Note that, sometimes the classification trees may overlap with each other in one ontology when a concept belongs to more than one classification tree, and in this case, this concept can be duplicated in the trees that it belongs to, so that all the classification trees are self-complete and independent. For a detailed description of classification-based method of ontology mapping, readers can refer to (Liang, *et al.*, 2007).

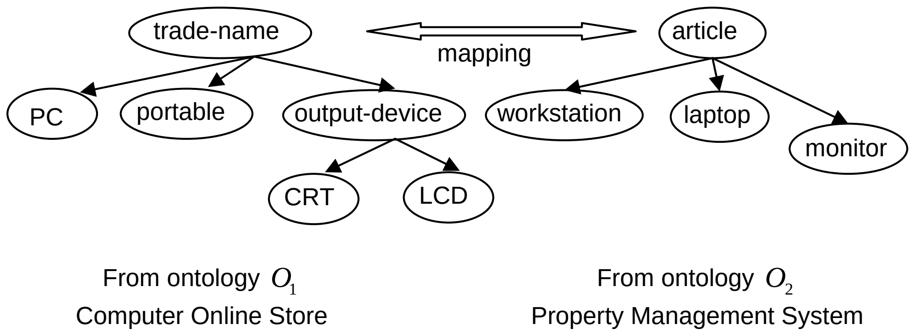

Figure 6. Example of mapped classification trees from two ontologies

The classification-based ontology mapping method we introduced in this chapter has been employed in our proposed B2B interoperability framework. We found with some cases that this ontology mapping method is especially beneficial for B2B systems when two trading parties employ the ontology from a similar domain. In such situation, it is easier for domain experts to create the ontology mappings between root concepts that have more sub-concepts, but this finding should be further evaluated in more complex B2B environment. This ontology mapping method focuses on the IA issues of content communication of business terms and alleviating the effort of ontology mapping by domain experts without covering formal representation mechanisms of ontology, e.g., OWL (W3C, 2007) or WMSO (W3C, 2005). Note that, this method can be used to improve, but not guarantee, the semantic interoperability of business terms communication since the semantic interoperability between two systems can only be improved, but not one hundred percent guaranteed (King, 2007). There are many other interoperability issues to be investigated at the semantic level of B2B systems, including but not limited to: understanding of business terms in various business context; decomposition and aggregation of business terms in different granularities for ontology mapping; business terms mapping with semantic relationships and constraints, etc. These interoperability issues at semantic level should be further studied and addressed in the future work.

3.3 Business Process Level

Business process is composed of a collection of related, structured activities or tasks that produce a specific service or product. In B2B systems, business process is normally employed to serve a particular goal for a customer. When an e-Business system is composed of multiple B2B systems, various business processes need to interact with each other (e.g., processes integration, message communications) in order to fulfill more complex tasks. For example, a business process of *shipping product to buyer* needs to communicate with another business process of *locating the address of buyer* in order to deliver the product to the right address.

Due to the popularity and wide applications of B2B systems in various domain, many existing business process specifications have been proposed and used in the global market of B2B systems, including Wf-XML, WSFL, XLANG, BPEL4WS (IBM, *et al.*, 2003), ebBPSS (OASIS, 2001a), and RosettaNet PIP, etc. As shown in Figure 7, various business process specifications have been constructed and evolved into an crossing-onion structure. Wf-XML is the core of this crossing-onion structure, which is based on the Workflow Reference Model (WMC, 1995) with an XML representation. Wf-XML provides a general framework and the core elements for business process specification. Other business process specifications (e.g., BPEL4WS, ebBPSS) are defined based on the Wf-XML, with different features. For example, BPEL4WS focuses on the coordination (e.g., orchestration and choreography) of Web services, while ebBPSS stresses upon message communications in business transactions. These diversities make the IA of B2B systems at the business process level a great challenge. The objective of IA at business process level is to achieve a transparent integration and communication between various business processes, which are formulated according to specific business process language.

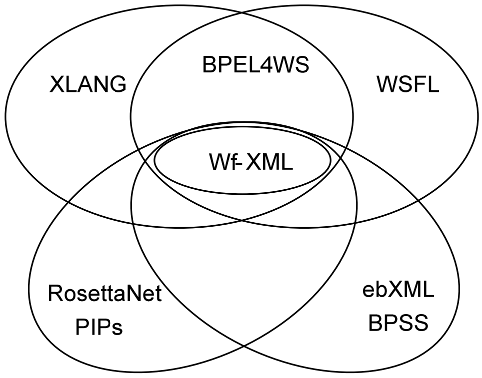

Figure 7. The relationship among various business process specifications

We proposed a meta-modeling method to guarantee the IA at the business process level. We explain this method through the interoperability between two mainstream business process languages: ebBPSS and BPEL4WS. The basic idea is that: any business process specifications have their own/specific meta-model, which describes the business process elements and relationships in the process (e.g., *Collaboration*, *BusinessState* elements in ebBPSS, and a *Collaboration* has one or more *BusinessStates*). The interoperability at the meta-model level (the meta-model of business process specification) can guarantee the IA at the model level (He, *et al.*, 2005), i.e., the instances of business process specification language. First, we construct the meta-models according to the specifications of these two business process languages; second we compare these two meta-models, and construct a Business Process Concept Model (BPCM) as a core meta-model to coordinate the interoperability between these two business process languages. The construction of the core meta-model BPCM, follows the construction rules below based on the concept comparison results between ebBPSS and BPEL4WS:

(1) For equal concepts, e.g., the *Role* concept in ebBPSS and the *Partner* concept in BPEL4WS, create a single concept to represent them in the BPCM;

(2) For similar concepts, e.g., the *BusinessActivity* concept in ebBPSS and the *Activity* concept in BPEL4WS, use sub-concept relationship to link them, e.g., *Activity* is a sub-concept of *BusinessActivity*;

(3) For concepts that are composed of other concepts, e.g., the *CorrelationSet* concept in ebBPSS is composed of the *Activity* concept in BPEL4WS, ignore the composite concept (i.e., *CorrelationSet*) in the BPCM;

(4) For concepts that are proprietary for each business process specification, e.g., the *BusinessTransaction* concept in ebBPSS and the *CompensationHandler* concept in BPEL4WS, extend the BPCM by adding these concepts with appropriate relationships (e.g., sub-concept relationship).

The resulting core meta-model BPCM is shown in Figure 8. Readers can refer to (Liang, 2006) for a detailed description about how the construction of BPCM can address the business process interoperability between ebBPSS and BPEL4WS.

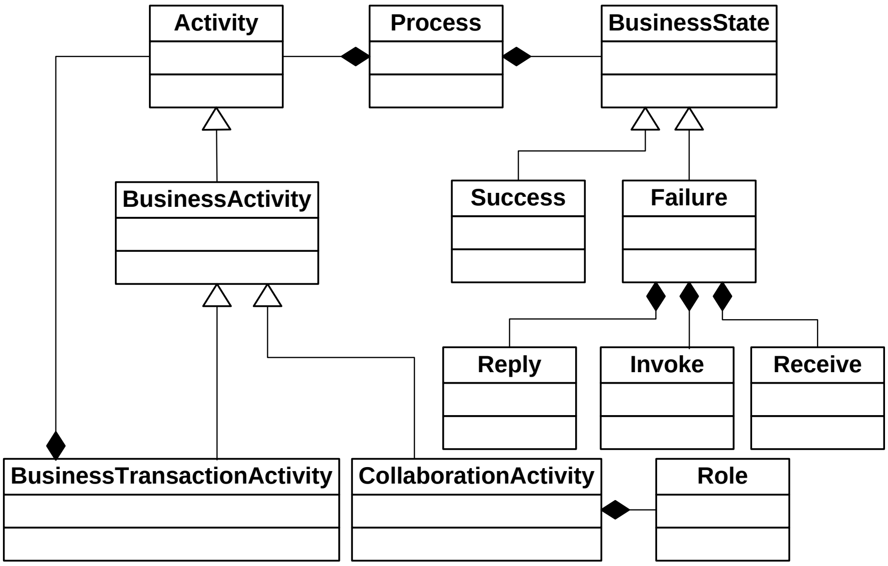

Figure 8. BPCM core meta-model for the business process interoperability between ebBPSS and BPEL4WS

4. SKLSE-EBIOT INTEROPERABILITY TEST PLATFORM

ebXML is composed of a family of XML-based standards (e.g., messaging service specification, business process specification, etc.) that enable the global use of e-Business information in an interoperable, secure, and consistent manner by all trading partners. It is the most promising standard to fulfill the B2B e-Business market, especially for small and medium enterprises (SME), with relative lower cost and technical barrier than traditional EDI-based B2B systems. As discussed before, the ebXML standard itself can not guarantee the interoperability of B2B systems that are implemented based on ebXML specification, because it only provides a general reference architecture for the B2B system implementation without constraining technical details. In this section, we present the SKLSE-ebIOT, an automatic tool for the interoperability test at the data level of ebXML-based B2B systems, and this interoperability test belongs to the first IA scenario discussed in Section 3.1.1, the IA of B2B systems based on same e-Business specification (i.e., ebXML).

4.1 Interoperability Test Requirements

There are six detailed levels in the ebXML interoperability test at the data/messaging level, and the test in each level is composed of several test items. Each test requirement statement in the test specification defines a test item, which corresponds to a specific requirement in ebMS specification. The test requirement of each test item specifies the function to be tested, which is used to generate the test case. In the end, the test case will be written by test script to be executed automatically. An interoperability test requirement is composed of three parts:

Precondition: The business scenario of the test item is defined in the precondition, which helps test engineers to understand how and when he/she can employ the test item. The test driver module (further discussed in Section 4.2) in SKLSE-ebIOT will prepare the test environment according to the precondition.

Statement defines the test assertion of each test item, and is composed of a list of inspection items as presented in Section 3.1.1 (e.g., <*RefToMessageID*> in the receiving message should be the same as the <*MessageID*> in the sending message). The test statement is associated with a key word from the satisfaction degree (discussed below), e.g., *MUST* (a test item should be satisfied), which means that if the test precondition is satisfied but the test statement fails, the result of the test item is *FAIL*.

Satisfaction Degree specifies the significance and priority of a test item, which consists of several key words: *RECOMMEND* (test item had better be satisfied), *MUST* (test item should be satisfied), and *MAY* (whether the test item should be satisfied or not is up to the stakeholders of the test item). The element representing the satisfaction degree is integrated in the test script in XML.

4.2 Architecture of SKLSE-ebIOT Test Platform

We design and implement the SKLSE-ebIOT test platform according to the requirements of the interoperability test specification of ebXML at the messaging level. The SKLSE-ebIOT can execute the test items defined in the interoperability test specification automatically, and record the test results. In this section, we describe the architecture of SKLSE-ebIOT in details.

The architecture of SKLSE-ebIOT test platform is in accordance with the OASIS IIC ebXML test framework specification. As shown in Figure 9, the test platform adopts a B/S (Browser/Server) structure in order to realize distributed test operation in a Web environment. The SKLSE-ebIOT can be deployed on various operating systems, such as Windows and Linux, since the platform is implemented in Java language, which is also beneficial for the processing of XML-based e-Business messages. The SKLSE-ebIOT test platform is composed of three parts which are described in details below:

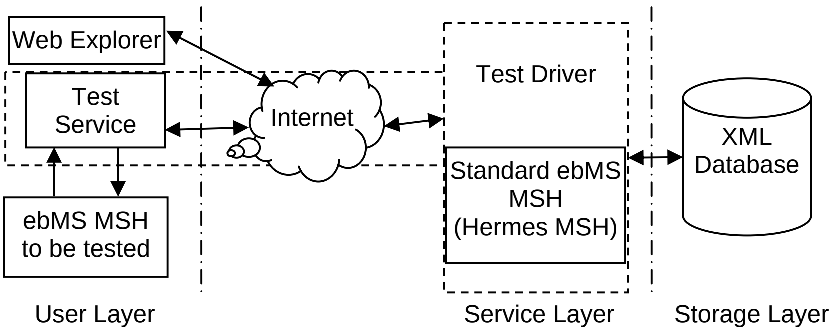

Figure 9. Architecture of SKLSE-ebIOT interoperability test platform

User Layer: This layer consists of Web explorer as the user interface, Javascript and applet components for the test service implementation. This layer is responsible for the user communication in the test: (i) test engineers select and execute test cases through the Web explorer, and get the test results of test cases in the Web explorer. (ii) test service module, as a message mediator, is responsible to forward the test message from the ebMS MSH (to be tested) to the test driver, and deliver the response message back to ebMS MSH (to be tested) through the Internet. The ebMS MSH in this layer represents the ebXML messaging service implementation to be tested by SKLSE-ebIOT. In this case, it is Fujitsu Interstage ebXML component that are presented in Section 4.4.

Service Layer: The layer is composed of a test driver and an ebMS MSH implementation, the Hermes MSH[iv], which is in accordance to OASIS ebXML IIC test specification (OASIS, 2002b). We regard the Hermes MSH as a standard ebMS MSH implementation, which acts as a test-bed for other ebMS MSH implementations/solutions. The detailed functions of the test driver and standard ebMS MSH are presented below:

(i) **Test driver** is responsible for loading test configuration, monitoring and receiving test requests from the Web explorer in the user layer, retrieving corresponding test script and test material (such as message payload) from XML database in the storage layer according to specific test request. The test driver parses the test script and generates test instructions. Each test instruction and the corresponding test material will be sent to standard ebMS MSH. Standard MSH will send correct message to the ebMS MSH to be tested. In addition, test driver is responsible to check the response messages back from the ebMS MSH to be tested according to inspection items, and record the check results into XML database for generating the test report.

(ii) **Standard ebMS MSH**: as mentioned before, the SKLSE-ebIOT adopts Hermes MSH V1.0, which is developed by the Center for E-Commerce Infrastructure Development (CECID) at The University of Hong Kong, as a standard ebMS MSH implementation. Hermes MSH implements a set of APIs that can package, send, and receive the ebMS message based on ebMS specification. Hermes MSH is an open source project which can be easily integrated with the test driver, and has been validated by the OASIS IIC ebXML test specification. The Hermes MSH component is responsible to receive the test instructions from the test driver, send test messages to ebMS MSH to be tested, and receive the response messages from the test service module.

Storage Layer: This layer is mainly composed of an XML database, which stores executable test script and ebMS messages in XML. It is responsible to store messages that standard ebMS MSH sends to, and receives from the ebMS MSH to be tested. All the recorded message dump and test log information will be used for later inspection automatically.

4.3 Functions of SKLSE-ebIOT

The major functions/services provided by SKLSE-ebIOT are as follows (1) Conversation service is responsible to manage conversation between test service and test driver, including services such as user identification, authority administration, etc. (2) Test case management service: user can add, remove, and modify test cases and test material using this service. (3) Inquiry service: user can trace the sending & receiving messages during the test process by this service after the completion of selected test case. (4) Test parameter configuration service: test parameters need to be configured or modified during the test process, such as the URL endpoint of ebMS MSH to be tested.

4.4 Test Cases and Test Results

We employed SKLSE-ebIOT to test the interoperability of Fujitsu Interstage ebXML component, an ebXML-based messaging service implementation deployed at Fujitsu Interstage Application Server V5.0. All the test items included in the six test levels (see Table 1) on ebMS are tested. The automatic interoperability test environment is deployed in two PCs in the LAN network: SKLSE-ebIOT test platform with Hermes MSH on one machine, and Fujitsu Interstage ebXML component on the other. We simulate the Internet environment (globally accessible IP address) for each machine by using the local IP addresses.

We got successful interoperability test results on most of test items except for the test items in the T3 level: SSL security messaging. We inspected the test log files generated by SKLSE-ebIOT, and found that the failed test items were due to the different encryption algorithms used for the generation of root CA (Certificate Authority) file (e.g., different CA files may be generated using different bits of keys, e.g., 512 bits and 256 bits) for the verification of security messages (sending and receiving). The detailed description about the test environment, test cases, test script, and test results can be found in (Chen, 2005).

5. RELATED WORK

B2B e-Business system is a typical Web services-based system. In this section, we briefly introduce and review the related work on the initiatives and research on the interoperability of Web services-based systems, including B2B systems.

The WS-I (Web Services Interoperability) Basic Profile is a specification defined by the WS-I industry consortium. It provides interoperability guidance for core Web services specifications, such as SOAP and WSDL (Web Service Description Language), which are located in the data level in our proposed interoperability framework. It actually defines a much narrower set of valid services than the full Web service specifications in order to achieve the IA by more rigorous specification. WS-I Basic Profile also provides tool support to validate Web services implementation so that all the Web services validated by WS-I Basic Profile can interoperate. There are also some interoperability problems that WS-I Basic Profile can not address, e.g., *NullArray* processing in SOAP message, and *namespace* definition and resolution in WSDL, etc (Feng, *et al.*, 2006).

The ebXML interoperability test initiative organized by ebXML Asia Committee (ebAC) currently focuses on the IA at the data (messaging service) level. An interoperability test specification is defined before the test, including the test levels, procedures, and evaluation criteria. All the tests are performed in a manual way by test engineers, and the test activity is organized in an $n*(n-1)$ manner (n represents the number of business organizations who participate the interoperability test), which means that an organization should complete the interoperability test with all the other participants in order to get qualified at certain test level. A degree of IA will be awarded to the participating organization depending on how many interoperability test items it succeeded. There are two major problems in this initiative: the test is performed manually, including the test procedure and test results inspection, which is error-prone; and the number of interoperability test will increase exponentially when the number of test participants increases. A standardized and automatic interoperability test platform can fairly address these issues.

Lampathaki *et al.* performed a thorough review on various XML-based data standards for B2B e-Business, and try to come up with a common understanding of XML Data Integration Standards for B2B in order to achieve the B2B interoperability at data level (Lampathaki, *et al.*, 2009). The major challenge of this work is the acceptance of this integrated standard by industry due to the existence of various XML data standards for B2B.

The ultimate aim of business document interoperability is to exchange business data (content) among partners without any prior agreements on document syntax and semantics. Kabak and Dogac made a survey and analysis on e-Business document standards (Kabak & Dogac, 2010). Their work stresses upon the content (business documents and terms) exchanged between B2B systems, which is located at the semantic level in our proposed IA framework. Janner *et al.* employed the OASIS UBL and the UN/CEFACT Core Component Technical Specification (CCTS) standards to model business documents and terms (Janner, *et al.*, 2008) in order to achieve semantic interoperability in B2B e-Business, but this approach limits the IA at the semantic level between B2B systems based on same specification.

Kajan and Stoimenov proposed the B2BOOM, an ontology-driven framework to address the semantic interoperability issues in B2B systems in (Kajan & Stoimenov, 2005), which is at the semantic level in our proposed interoperability framework. In the B2BOOM framework, the meaning of the terminology of each B2B system is formally specified using a local ontology, and a translation (mapping) by a translator (wrapper) is defined between each B2B system terminology (local ontology) and an intermediate terminology (top-level ontology). A semantic data mediator is employed to perform the translation when a specific term is requested, and all the shared/common data (terms, term mappings) are maintained in the B2B server that all the business parties can access. Kajan and Stoimenov extended their work to address the semantic interoperability at B2B process level through semantic process mediators in (Kajan & Stoimenov, 2009). The B2BOOM framework didn't cover the interoperability issues at the data level discussed in this chapter.

Green *et al.* present an ontological evaluation model based on Bunge-Wand-Weber (BWW) models to address the business process interoperability in ebXML, i.e., the ebBPSS, against the evaluation of enterprise systems interoperability standards (Green, *et al.,* 2005). Their work focuses on the evaluation of business process interoperability based on ebBPSS specification through the mapping of the ebBPSS elements into BWW representation model. The interoperability validation of ebBPSS in a concrete implementation will be done in their future work.

6. CONCLUSIONS AND FUTURE WORK

e-Business systems are one of the dominant applications in the Internet age, and have been widely used in our society and part of our life. The interoperability of e-Business systems is the most fundamental issue for the success of e-Business than other factors, like security and trust etc, especially in a B2B market. The B2B systems should be able to handle all kinds of interoperability issues, from technical, domain to human aspects. In this chapter, we focus on the technical aspect of the interoperability issues for the B2B systems, and present a general solution for the IA of such systems, including (1) an interoperability framework in three layers for B2B e-Business systems; and (2) an automatic interoperability test platform for achieving the IA at the data/messaging level. We demonstrate the effectiveness of our proposed interoperability framework and related methods with concrete cases from ebXML, a promising XML-based standard for the next-generation of B2B systems.

Although the proposed interoperability framework and methods (e.g., interoperability at data level, and the interoperability test) are demonstrated with ebXML examples, it is generally applicable to other e-Business specifications. For the general interoperability framework, it has to be adapted and customized into an existing e-Business specification before it is put into practice. For the automatic interoperability test platform, it can be easily modified to adapt to other e-Business message specifications, e.g., cnXML.

We outline our future work in several points (1) extend the interoperability test platform to the interoperability test of other e-Business solutions (e.g., based on cnXML), and also the interoperability test between B2B systems based on different e-Business specifications with the support of message translator. (2) the interoperability of B2B systems at the semantic level and business process level can not be fully guaranteed, but we can specify and provide detailed test levels to quantify the interoperability degree. (3) the interoperability of e-Business repository is an unexplored area that needs further investigation, e.g., UDDI, ebXML Registry and Repository, how these repositories can interoperate to provide repository service to e-Business systems. (4) the interoperability issues on security between B2B systems is also a critical aspect in our future work.

ACKNOWLEDGEMENT

This research project has been partially supported by the National High Technology Research and Development Program of China (863) under Grant No.2006AA04Z156, the National Basic Research Program of China (973) under Grant No.2007CB310801, the National Natural Science Foundation of China (NSFC) under Grant No.60903034, No.60970017. Thanks to the effort by ebXML Asia Committee (ebAC) for providing and organizing interoperability test in Asia. Thanks to Fujitsu Co. Ltd. for providing the Interstage ebXML component and Interstage Application Server V5.0, and thanks to the Center for E-Commerce Infrastructure Development (CECID), The University of Hong Kong for providing the open source Hermes MSH (Message

Service Handler) V1.0 for our research. We also would like to thank all the reviewers for their constructive comments.

REFERENCES

Chen, S. (2005). *Auto-testing tool for e-Business interoperability test.* Master Thesis, Wuhan University, China.

Doerr, M. (2003). The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI Magazine*, 24(3), 75-92.

Feng, Z., Liang, P., Liu, Y., Wu, Z., & He, K. (2006). Interoperability test for web service descriptions. *Application Research of Computers*, 23(11), 191-195.

Gibb, B.K., & Damodaran, S. (2002). *ebXML: concepts and application*. New York, USA: John Wiley & Sons.

Green, P.F., Rosemann, M., & Indulska, M. (2005) Ontological evaluation of enterprise systems interoperability using ebXML. *IEEE Transactions on Knowledge and Data Engineering*, 17(5), 713-725.

He, K., He, F., Li, B., He, Y., Liu, J., Liang, P., & Wang, C. (2005). Service-oriented ontology & meta-modeling theory and methodology. *Journal of Computer*, 28(4), 524-533.

Kabak, Y., & Dogac, A. (2010). A survey and analysis of electronic business document standards. *ACM Computing Surveys*, 42(3), 1-31.

Kajan, E., & Stoimenov, L. (2009). An approach for semantic-based EC middleware. In *Proceedings of the IADIS International Conference on e-Commerce (EC)* (pp. 69-76). IADIS Press.

Kajan, E., & Stoimenov, L. (2005). Toward an ontology-driven architectural framework for B2B. *Communications of the ACM*, 48(12), 66-72.

King, M. (2007). Semantic interoperability: is it always achievable? *Technical Talk*, Sydney.

Lampathaki, F., Mouzakitis, S., Gionis, G., Charalabidis, Y., & Askounis, D. (2009). Business to business interoperability: A current review of XML data integration standards. *Computer Standards & Interfaces*, 31(6), 1045-1055.

Liang, P., He, K., Pan, Y., & Feng, Z. (2007). A classification-based method for ontology mapping. *Technical Report WHU-SKLSE-07-L01*, Wuhan University.

Liang, P. (2006). *Interoperability for semantic web services.* PhD Thesis, Wuhan University, China.

Liang, P., He, K., Li, B., & Liu, J. (2005). The interoperability between different e-Business specifications. In *Proceedings of the International Conference on Information Technology: Coding Computing (ITCC)* (pp. 409-413). IEEE Computer Society.

Liang, P., He, K., Li, B., & Liu, J. (2004). Interoperability test of ebXML e-Business solutions. In *Proceedings of the 4th International Conference on Computer and Information Technology (CIT)* (pp. 1004-1007). IEEE Computer Society.

Liang, P., He, K., Yao, J., Song, D., & Zhou, B. (2003). ebXML interoperability test report. *Technical Report WHU-SKLSE-03-L01*, Wuhan University.

IBM, BEA Systems, Microsoft, SAP AG, Siebel Systems. (2003). *Business process execution language for web services.* Version 1.1. Retrieved December 29, 2009, from http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf

Janner, T., Lampathaki, F., Hoyer, V., Mouzakitis, S., Charalabidis, Y., & Schroth, C. (2008). A core component-based modelling approach for achieving e-business semantics interoperability. *Journal of Theoretical and Applied Electronic Commerce Research*, 3(3), 1-16.

OASIS. (2008). *Universal Business Language 2.0 - International Data Dictionary Vol. 1*. Retrieved December 29, 2009, from http://docs.oasis-open.org/ubl/idd/cs-UBL-2.0-idd01/cs-UBL-2.0-idd01.zip

OASIS. (2002a). *ebXML messaging service specification*. Version 2.0. Retrieved December 29, 2009, from http://www.ebxml.org/specs/ebMS2.pdf

OASIS. (2002b). *ebXML test framework*. Version 0.3. Retrieved December 29, 2009, from http://xml.coverpages.org/IIC-ebXMLTestFrame10.pdf

OASIS. (2001a). *ebXML business process specification schema*. Version 1.0.1. Retrieved December 29, 2009, from http://www.ebxml.org/specs/ebBPSS.pdf

OASIS. (2001b). *ebXML technical architecture specification*. Version 1.0.4. Retrieved December 29, 2009, from http://www.ebxml.org/specs/ebTA.pdf

Pan, Y., Liang, P., He, K., & Feng, Z. (2007). Classification-based method for ontology mapping and mapping tool implementation. *Application Research of Computers*, 24(10), 213-215.

Pereira, A., Cunha, F., Malheiro, P., & Azevedo, A. (2008) ebXML overview, initiatives and applications. In *Proceedings of IFIP International Federation for Information Processing, Volume 266, Innovation in Manufacturing Networks* (pp. 127-136). IFIP Press.

W3C. (2007). *Web ontology language (OWL) structural specification and functional-style syntax*. Version 1.1. Retrieved December 29, 2009, from http://www.webont.org/owl/1.1/owl_specification.html

W3C. (2005). *Web service modeling ontology (WSMO)*. Version 2.0. Retrieved December 29, 2009, from http://www.wsmo.org/TR/d2/v2.0/

W3C. (2000). *Simple object access protocol (SOAP)*. Version 1.1. Retrieved December 29, 2009, from http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

WMC. (1995). *Workflow reference model*. Retrieved December 29, 2009, from http://www.wfmc.org/standards/docs/tc003v11.pdf

Xu, B., Jiang, L., & Ma, F. (2005). On the new B to B e-Business enabling platform: cnXML in China. In *Proceedings of the 7th International Conference on Electronic Commerce (ICEC)* (pp. 681-684). ACM Press.

KEY TERMS & DEFINITIONS

e-Business Specification: The specification defines how an e-Business system should be implemented. It is normally composed of a series of sub-specifications, e.g., specification for messaging service, business terms, and business processes, etc. EDI, ebXML, and cnXML are typical examples of e-Business specification.

cnXML: An Electronic Business specification language similar to ebXML, which is also based on XML, but targeting to the Chinese market.

Interoperability Test Platform: A test platform, which is used to test the interoperability issues between two systems, or the interoperability of a system against a standard implementation.

Messaging Service: It is a fundamental service provided by the B2B systems. It is responsible to send, receive, and forward messages between business organizations.

Ontology: In computer science and information science, an ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It can be used to reason about the properties of that domain, and also be used to define the domain.

Universal Business Language: It is a library of standard electronic XML business documents for describing the commonly-used terms in e-Business, such as purchase order and invoice.

Ontology Classification Tree: This is a concept original from the structure of ontology, which is composed of a group of concepts that belong to same root concept in an ontology.

Business Process: It is a collection of related, structured activities or tasks that produce a specific service or product for customers.

---

[i] In this chapter, we focus on the B2B e-Business, and the terms B2B, B2B system, and B2B solution have the same meaning as B2B e-Business system.

[ii] cnXML is an XML-based B2B standard similar to ebXML specification in technical architecture, but targets to the Chinese market with special support on Chinese business terms and business processes, etc. The cnXML message structure is different from the ebXML message.

[iii] The interoperability test items are a subset of, and based on conformance test items.

[iv] Center for e-Commerce Infrastructure Development, The University of Hong Kong, http://www.freebxml.org/msh.htm.