Implementation of ebXML compliant Business Process Management System

SUNGHWAN CHOI, JAEGAK HWANG, YONG-JOON LEE and HYE JIN PARK
Postal Technology Research Center
E-Logistics division
Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon
REPUBLIC OF KOREA

Abstract: Internet based B2C e-Commerce is flourishing, but not so much in the area of business to business. For over 20 years, EDI has established VAN based solution of exchanging business information in electronic form. However EDI solutions are only accessible to large organizations due to the cost factor. Moreover lack of well accepted B2B business process standards is hindering the success in promoting interoperability between organizations of any size. ebXML work is focused on defining the standard B2B business process context and its runtime semantics in order to remove these hindrances. The paper first address the current status of ebXML from an implementation view and identify open issues to be solved. Then we briefly introduce ebXML compliant Business Process Management System (BPMS). The BPMS is an application of ebXML specification that executes and monitor business process model defined by ebXML Business Process Specification Schema (BPSS) in automated fashion.

Key-Words: - ebXML, BPSS, BSI, eCommerce

1 Introduction
Given the fact that commercially successfully implemented B2B standard like EDI for exchanging data to do business have already been established long time ago, the field of electronic commerce has quite a long history.

EDI was commercially successful at creating a standard vocabulary of electronic document exchange [8]. EDI has allowed automation of interchange of business transaction between large corporations and their trading partners using EDI data interchange format. Unfortunately, EDI not only deploys tightly coupled, inflexible architectures but also requires significant technical expertise. Moreover EDI applications are most often deployed on expensive dedicated networks to conduct business with each other. As a result, EDI adoption has been limited to primarily large enterprises and selected trading partners, which represents a small fraction of the world’s business entities. Looking at the statistics of who is currently utilizing EDI: only top 10,000 companies on a global scale are using EDI. For the rest of the business world only 5% are using EDI and therefore today common business processes are dominated by paper transactions [6].

Realizing the full potential of B2B exchange in the field of electronic commerce requires more than creating a new e-business vocabulary of electronic document. Due to the complexity and dynamic nature of rapidly evolving electronic commerce, data centric approach suffers from the aspects of interoperability, flexibility, and reusability. Although EDI was commercially successful at creating a standard vocabulary of electronic document exchange, it failed to gain critical mass for small to medium enterprises (SME) due to high implementation and maintenance cost.

In order to overcome the obstacles described above, the United Nations body for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS) started B2B standard, the ebXML (electronic business XML) [4]. The ebXML framework focuses on both the business process and the technical aspect of B2B.

The ebXML business process models describe interoperable business processes that allow business partner to collaborate [1]. The ebXML business process specification schema (BPSS) aims to provide process centric method of capturing real life business problems by adopting subset of UMM (UN/CEFACT
Modeling Methodology) which focuses on specifying business semantics required to facilitate interaction between business partners as well as runtime aspects of defined models. The UMM meta-model facilitates the specification of reusable, interoperable business process models that are technology and protocol insensitive. [5] In detail, the UMM works with a meta-model that has a few fundamental views namely; business domain view (BDV), business requirements view (BRV), business transaction view (BTV) and business service view (BSV). What the UMM basically says is that as long as the transition of the meta-models between the views, for example going down from the BDV to the BRV and then to the BTV is consistent and complete, under the BSV it is independent of the underlying implemented technology.

Technical component of ebXML specification consists of ebXML Messaging Service Handler (MSH), and Business Service Interface (BSI). ebXML Messaging Service Specification focuses on defining a communication-protocol neutral method for exchanging electronic business messages in XML. It defines specific enveloping constructs supporting reliable, secure delivery of business information. ebXML Business Process Specification Schema (BPSS) describes interoperable business processes that allow business partners to collaborate. These models must be executed by software components that collaborate on behalf of the business partner. BPSS provides configuration parameters for the partner’s runtime system in order to execute that collaboration between a set of e-business software components. The ebXML describes such software as Business Service Interface (BSI).

The BSI is built of two components; the Business Interface and the Service (Technical) Interface. Since the technical interface includes details of endpoint, protocol etc, it uses API of MSH for interacting at the service layer. Therefore the complete ebXML application that executes the collaboration process between business partners must consist of ebXML MSH and BSI. In our implementation of ebXML we combined the function of MSH and BSI, and named it Business Process Management System (BPMS).

The structure of the paper is as follows. Section 2 summarizes the current status of ebXML BPSS and identify the open issues from implementation point of view. Section 3 describes basic system architectures of the BPMS. Section 4 describes the features of the BPMS. Section 5 outlines our future work in this area and concludes the paper.

2 Current Status of ebXML BPSS

ebXML BPSS is a major component of ebXML framework. It provides for the first time the capability to express formally and in a machine-readable format a shared understanding of the interactions between business partners. ebXML BPSS describes business semantics of B2B scenario using a consistent modeling methodology, UMM. The UMM meta-model provide a set of business process centric viewpoints that describe the business semantics and artifacts that are required to facilitate business process and information integration and interoperability. Using the UMM methodology and the UMM meta-model, the user may create a complete business process and information model. The ebXML BPSS adopts subset of UMM to support the direct specification of the nominal set of elements necessary to configure a runtime system in order to execute business transactions. The relationship between ebXML BPSS and UMM is illustrated in Figure 1 below.

While the BPSS has been tested in proof-of-concept demonstrations, it is in a preliminary state from an implementation point of view. Current version of the BPSS provides basic support for runtime transaction semantics; however it fails to specify how failures can be handled in clear manner. Moreover it only defines state transitions at the level of collaboration and it is hard to implement business transactions semantics when timeout and other exceptions occur. Hence the
full business process definition, ranging from the business requirements to the runtime aspects as defined by the UMM has to be considered. However, ebXML does not mandate the use of UMM during the process of business process modeling. It states that UMM meta-model in its entirety is not part of the formal set of ebXML specifications [4]. Only the semantic subset represented by the BPSS is part of the formal set of ebXML specification. Other important aspects concerning BPSS are multiparty business collaborations. Currently, multiparty collaborations are always synthesized from two or more binary collaborations. This work in most real life business scenario, but it will not work when a reply is coming from another party than the request was sent to. Moreover ebXML BPSS does not describe transaction roll backs when any exception happens. ebXML recommends that transaction roll-back can be explicitly modeled as part of the choreography of business process. Research should extend the BPSS meta-model by adding transactional roll-back mechanism.

3 Basic System Architecture of BPMS
BPMS (Business Process Management System) consists of BSI and ebXML Messaging Service Handler (MSH). With the BPMS system, business process models described in BPSS can be executed and monitored in automated fashion. The BSI is an abstract layer above ebXML MSH. The BSI is built of two components: the Business Interface and the Service (technical) Interface. Since the technical interface includes details of endpoint, protocol, security configuration etc, it uses API of MSH for interacting at the service layer.

MSH provides protocol neutral and reliable messaging service for BSI system. Figure 2 illustrates the relationship of BIS and ebXML. As depicted in figure2, BSI client application prepares XML formatted business document and send it to MSH using Java Messaging Service (JMS) protocol. When the MSH receives the XML document, it packages XML document as specified by ebXML Messaging Service Specification (ebMS). ebMS is defined as a set of layered extensions to the base Simple Object Access Protocol (SOAP). ebMS extends SOAP by adding security and reliability features necessary to support international e-business. Once the message packaging is complete, MSH sends ebMS with XML documents to destination MSH. The core function of BSI Engine is to execute business transactions by following state transition models defined by BPSS. It also provides real time monitoring of running business process models.

4 Features of BPMS
BPMS consist of ebXML compliant BSI engine that execute business process models described in BPSS. It uses BPSS instance document (modeling artifacts) to configure runtime environments, for example some transactions like purchase order proposal may require non-repudiation and authorization service while some transactions that involve simple notification only require reliable messaging service. Depending upon the nature of transactions involved BSI Engine need to configure itself using BPSS instance document. Also BSI engine need to follow the state transition information provided by BPSS instance document in order to keep the synchronization of running transactions.

Figure 3. BSI Client Application
Our implementation of BSI provides full automation of business transaction execution and monitoring which happens at real time. Figure 3 illustrates client application GUI of BSI Engine. It provides real time monitoring of currently running business process in graphical representation.

5 Conclusions and Future works

In this paper, we have overviewed the current status of ebXML BPSS and identified open issues from an implementation point of views. Furthermore we have implemented ebXML BPSS and briefly introduced BPMS. As aforementioned, the current version of BPSS has limitations. Research should identify further limitations of BPSS and extend BPSS meta-model.

References: