E-BUSINESS INTEROPERABILITY
Challenges and Opportunities

George Feuerlicht[1,2]
[1]*Faculty of Engineering and Information Technology, University of Technology, Sydney*
P.O. Box 123, Broadway, NSW, 2007, Australia
[2]*Faculty of Information Technology, University of Economics, Prague, Czech Republic*

Keywords: e-Business interoperability, SOA, e-Business standards.

Abstract: Starting with EDI in the 1970s numerous efforts have been made to facilitate electronic business communications. Early attempts based on proprietary document formats and VAN (Value Added Networks) networks have been superseded by a range of international standards that include ebXML, UBL, BODs, GS1, RosettaNet and numerous other XML-based specifications. While successful in some industry domains, overall e-business standardization suffers from complexity of the specifications, difficult customization and limited extensibility leading to expensive implementation and low adoption rates. A common feature of such e-business standards is their focus on documents as the key artefacts of business communications and reliance on document engineering methods for the design of the standard specifications. In this paper we briefly review the main e-business document standards, and then argue that the document-centric interoperability model underlying most current e-business standards produces inflexible specifications that are difficult to evolve and maintain. As an alternative to the document-centric interoperability model we advocate a service-centric approach based on well-designed domain services.

1 INTRODUCTION

The vision of seamless electronic communications between organization dates back to the early days of networking in 1970s. Replacing paper-based business communications with electronic documents that can be transmitted at very low cost and acted on instantly has been the focus of considerable efforts initially in supply chain situations and later across entire industry domains. The economic benefits of reducing the cost and improving the accuracy of business to business transactions can be considerable. For example, a study that investigated the cost of poor interoperability resulting from a lack of standardization in the US automotive industry estimated the loss to be about one billion dollars per year (Brunnermeier and Martin 2002). In a recent study of the impact of e-business (electronic business) interoperability Legner et al. compare the cost of lack of interoperability across various application domains, including manufacturing and healthcare management and identifies a range of interoperability research issues (Legner and Lebreton 2007). Notwithstanding extensive standardization efforts over several decades, e–business interoperability remains an open problem. Numerous standardisation bodies including UN/CEFACT (UN/CEFACT 2011), OAGIS (The Open Applications Group. 2010), RosettaNet (www.rosettanet.org 2007) have produced a plethora of e-business standards in various stages of completion. Low quality of standard specification resulting from poorly controlled standardization process relying on "design by committee" rather than on an effective design methodology is a frequent source of ambiguities and complexity presenting challenges to organizations adopting such standards. Furthermore, some researchers argue that availability of a document standard while being an essential prerequisite is not sufficient to guarantee e-business interoperability and advocate service-based interoperability solutions (Feuerlicht 2005; Legner and Vogel 2008). In this paper we first discuss the concept of e-business interoperability (section 2), and then review e-business standardization efforts identifying their common characteristics and limitations (section 3). In section 4 we argue that the document-centric model that forms the basis of most

Feuerlicht G..
E-BUSINESS INTEROPERABILITY - Challenges and Opportunities.
DOI: 10.5220/0003618601930198
In *Proceedings of the International Conference on e-Business* (ICE-B-2011), pages 193-198
ISBN: 978-989-8425-70-6

major e-business standard specifications is fundamentally flawed and that a new service-centric approach is needed in order to improve the effectiveness of e-business solutions. In the following section (section 5) we describe the service-centric interoperability model, and in section 6 we present our conclusions.

2 E-BUSINESS INTEROPERABILITY

Interoperability requirements of e-business applications have been exhaustively documented in the literature (Bussler 2001), (McAfee, Bettiol et al. 2007). According to Bussler interoperability can be classified into three levels: 1) technical, 2) information, and 3) business process levels. Technical level interoperability addresses technical issues including communication protocols, programming language environments, and technology platforms used by individual business partners. Information level interoperability concerns data elements transmitted between partner organizations and can be further classified into: syntax, structure, and semantic interoperability. Syntax refers to formats used to represent data elements (e.g. delimited document formats, XML, etc.). Structure and semantics of data elements refers to organization of data elements into compound structures and the meaning of individual data elements, respectively. Business process interoperability is concerned with collaborative activities between partner organizations. Bussler classifies business processes into public and private. A private process represents a flow of business tasks within an enterprise, while a public process represents a flow of interactions between business partners. Business process interoperability is concerned with public processes that define external actions that participants perform. Classification of e-business interoperability into different types (levels) is shown in Table 1. Other authors use similar classifications of e-business interoperability and there is an agreement about the need to develop standards for all levels of interoperability to fully automate e-business interactions.

Wide acceptance of XML as a document formatting standard and the emergence of standard protocols for delivering document payloads as XML formatted messages (REST (www.ics.uci.edu 2002), and Web Services SOAP (WC3 2007)) provide an effective interoperability solutions for the technology and syntax levels.

Table 1: Levels of e-business interoperability.

INTEROPERABILITY LEVEL	DESCRIPTION
Business Process	collaborative activities between partner organizations
Semantic	meaning of individual data items
Structure	structure of data elements
Syntax	document formats
Technology	communication protocols and technology platforms

Most standardization efforts focus on the remaining interoperability levels and attempt to address structure, semantic and business process heterogeneity by specifying standard XML messages and business process interactions, either for a specific industry domain (vertical standards) or across industry domains (horizontal standards). Examples of vertical standards include RosettaNet (www.rosettanet.org 2007), Chemical Industry CIDX (Chemical Information Technology Center) (Open Application Group 2010), Healthcare HL7 (Health Level 7) (HL7.ORG 2007), and Travel OTA (Open Travel Alliance) (OTA 2010). Attempts at all-encompassing horizontal standardization centre around the UN/CEFACT (UN/CEFACT 2011) Core Components Technical Specification (CCTS) and include ebXML (ebXML 2007), UBL 2.0 (OASIS 2011), OAGIS Business Object Documents (BODs) (The Open Applications Group. 2010) and The Global Standard One (GS1) XML (European Article Number (EAN) and the Uniform Commercial Code (UCC)). We discuss some of these standards in the following section.

3 E-BUSINESS STANDARDIZATION

Over the last three decades e-business solutions have evolved from relatively simple proprietary point-to-point systems that rely on documents translation to highly complex XML-based specifications. EDI (Electronic Data Interchange) (UN/EDIFACT 2010) was the first attempt to establish e-business interoperability standard initially using delimited plain-text documents mainly for procurement, logistics and finance domains. Communication between partner organizations is implemented using VANs (Value-Added Networks) and EDI adapters that are used to interface internal systems to the network. While providing an overall framework by defining a set of common business messages, EDI requires that the parties agree on the format and content of business documents, with dominant partners often imposing their standard on smaller organizations. This leads to fragmentation of the specification and to proliferation of diverse EDI versions that are established over time by dominant partners or groups. Such standards do not scale well when a large number of organizations is involved as each new business relationship requires a new set of message standards and corresponding translation software that converts internal proprietary data formats into trading partner-specific standard messages. The creation of new business messages or changes to the existing standard message sets are complex and time consuming activities, as the modifications have to be approved by UN/EDIFACT (United Nations/Electronic Data Interchange For Administration, Commerce and Transport) standard committees. More recent versions of EDI use Internet protocols (i.e. HTTP) as the transport for XML formatted messages. Despite its well documented limitations EDI use is still growing (Kabak and Dogac 2010).

Following these early attempts to standardize e-business interactions with EDI related specifications, a wave of vertical (industry domain specific) and horizontal (cross-industry domain) standards emerged, some with perplexing inter-relationships and overlapping specifications. We discuss these standardization efforts in the following sections.

3.1 Vertical Industry Domain Standards

Vertical standardization efforts are typically driven by industry-domain consortia that produce XML message specifications and standard process definitions for a given industry domain. When compared to horizontal standards, vertical standards tend to be more narrowly focused and provide higher levels of interoperability as many of the concepts and processes are shared across all participants. Examples of vertical domain standards include RosettaNet, OTA, and HL7. RosettaNet is a consortium of major computer and consumer electronics, semiconductor manufacturing, tele-communications and logistics companies. RosettaNet defines standards at three levels: technology, vocabulary, and process levels, covering the entire interoperability spectrum described in Table 1, and supports multiple messaging standards, including Web Services. RosettaNet uses the concept of Partner Interface Processes (PIPs) to define business processes between trading partners. PIPs define the interfaces between processes running in different partner organizations that constitute a supply chain. PIP definitions include message structure specification and business process logic that controls the flow of messages. RosettaNet has been particularly successful in high-technology supply chain applications, but so far has not gained wide acceptance outside the high technology sector.

Another example of a vertical standard is the Open Travel Alliance specification that defines a comprehensive set of XML message schemas (over 230 message schemas) as the basis for electronic communications between partner organizations within the travel industry. The OTA specification covers all aspects of travel business, including air travel, cruises, hotel accommodation, car hire, and travel insurance, with the scope of the specification continuously expanding. To implement specific business functions, for example to book a flight, OTA uses request (RS) and response (RQ) message pairs that are typically transmitted between parties as Web Services SOAP messages. OTA standard messages are widely adopted by companies that implement travel applications (e.g. Sabre (Sabre 2007)) and follow strict naming conventions and design guidelines (OTA 2010). OTA messages are constructed using common data types that form a repository of reusable XML Schema components. OTA specification does not directly address business process interoperability; composing individual Web Services into process flows using BPEL (Business Process Execution Language) (Arkin, Askary et al. 2007) could be used to provide such functionality.

3.2 Horizontal Industry Standards

Vertical standards do not fully address the requirements of e-business application that span industry domains. Horizontal e-business standards are designed to be industry domain neutral and typically deal with much greater scope of standardization. Horizontal e-business standard-ization efforts are led by UN/CEFACT with its Core Components Technical Specification (CCTS). The main idea underlying the CCTS specification is the concept of Core Components - *context-free reusable building blocks* that are maintained in a common repository (Core Component Library) and used to construct business documents. Core Components are aggregated into Aggregated Core Components (ACCs) to represent *real-world* entities (e.g. Purchase Order). Core Components that are specialized (adapted) for a specific business context (for example, a geopolitical context of the European Union) are called Business Information Entities (BIEs). Core Components and other CCTS artefacts are stored in the Core Component Library (CCL) and are used across a number of standard specifications as the basis for defining business documents. These standards include OASIS

Universal Business Language (UBL) 2.0 (OASIS 2011), Open Applications Group Integration Specification (OAGIS) Business Object Documents (BODs) (The Open Applications Group. 2010) and The Global Standard One (GS1) XML (European Article Number (EAN) and the Uniform Commercial Code (UCC)). In some cases the standards have been re-designed for compliance with the UN/CEFACT CCTS (e.g. OAGIS BODs). Although all of these standards are based on CCTS with the aim to ensure maximum flexibility of the specification, customization and extensibility features vary considerably from standard to standard (Kabak and Dogac 2010). Furthermore, some standards allow documents to contain embedded command commands that specify the operation to be executed on the document. For example, OAGIS BODs introduce the concept of *Nouns* (documents) and *Verbs* (actions) with verb-noun combinations specifying the action to be performed on the object, e.g. CancelOrder. Multiple Nouns can be associated with a single verb, allowing for the same action to be performed on multiple documents. GS1 XML documents contain *Commands* (Add, Delete, Refresh) and *Transactions* that support the execution of multiple Commands within the scope of a single transaction.

4 LIMITATIONS OF THE DOCUMENT-CENTRIC APPROACH

A common characteristic of the interoperability approaches described in the previous section (section 3) is that they adopt the document-centric interoperability model characterized by shipping business documents between partner systems. The original motivation for the document-centric approach was the need to overcome technology level heterogeneity; as documents provide a level of abstraction that allows business interactions based on the mutual understanding of data semantics without the need for compatible technology platforms (i.e. no direct interoperability between the partner systems is needed as the receiving partner is responsible for mapping the documents into transactions against the target system). As noted in section 2, standard protocols (i.e. Web Service SOAP or REST) address technical level interoperability making this approach unnecessary today.

The principal limitation of the document-centric approach is its tendency to use large and complex documents that often mirror the original paper-based forms, as message payloads. Messages typically include all the information needed to perform a particular business function without any reference to information already received. While such *stateless* interactions reduce the number of messages needed to implement a particular business function, improving reliability in failure-prone high latency network environments, the externalization of complex data structures introduces high levels of data coupling (Feuerlicht 2007). The message payloads form the interface between e-business applications and introduce interdependencies, making changes to standard documents difficult to perform without causing undesirable side-effects that invalidate existing applications. Design of standard documents is typically based on Document Engineering (Glushko and McGrath 2008) or similar methods that construct documents by identifying and aggregating common data elements, e.g. UN/CEFACT CCTS Core Components. Embedding aggregated Core Component structures into multiple business documents causes high levels of data coupling with corresponding impact on applications when the specification evolves. Embedding complex data structures formed by aggregation of core components (ACCs, etc) in business documents is promoted in Document Engineering literature as a technique for achieving reuse, but this approach differs fundamentally from software reuse as it applies to software engineering (e.g. in the programming context). It can be argued that the use of aggregated data structures implicit in the document-centric approach limits, rather than enhances reuse.

5 SERVICE-CENTRIC INTEROPRABILITY MODEL

Service-centric approach provides an alternative interoperability model by changing the level of abstraction from document interchange to a programmatic approach based on well-defined APIs (Application Service Interfaces). Unlike documents, procedures (i.e. service operations) typically implement specific business functions and use simple data parameters as procedure *signatures* (i.e. service interfaces). Service interfaces can be designed to limit externalization of data and to avoid creating unnecessary interdependencies between services. Encapsulating data structures and externalizing method signatures (i.e. interfaces of service operation) that constitute a stable contract between the service provider and the service consumer results in improved stability and ability to accommodate change. Software engineering principles can be applied to the design of service interfaces maximizing cohesion and minimizing service coupling. This limits the impact of changes to a small number of services improving the stability and maintainability of services, (Feuerlicht 2004). Extensibility is supported by versioning individual service interfaces rather than the entire specification as is the case in most document-centric standards (e.g. OTA).

There are two key requirements for achieving information-level interoperability for service-centric e-business applications in a particular domain: 1) standardization of all data elements (i.e. a domain-wide vocabulary standard) and 2) standardization of service interfaces. Without a domain-wide service interface standard, equivalent services published by different providers will not be compatible, placing the burden for resolving the inconsistencies on service consumers. Such domain-specific service interfaces are conceptually similar to APIs that are used extensively in programming environments (e.g. JDBC, or ODBC for database access). The abstraction level of standardized, domain-specific APIs is closely related to business processes in the particular domain, so that the APIs constitute a *programming* environment for developing domain-specific applications. For example, a travel application could be implemented using a specialized set of travel APIs based on the OTA specification, but with low granularity service operations that perform airline flight bookings, hotel reservations, car rentals, and other travel related transactions. Unlike the current practice of using OTA based Web Services as a transport mechanism for OTA messages, this approach relies on a well-designed domain-specific programming language as the basis for developing travel applications.

6 CONCLUSIONS

The emergence of SOA (Service-Oriented Architecture) and related technologies (Web Services, REST services, etc.) has created an opportunity to address e-business interoperability using a new paradigm. So far, most e-business solutions use Web Services (or REST services) primarily as a transport layer for the delivery of standard documents, failing to take full advantage of the service-oriented approach. Faster and significantly more reliable network connectivity available today makes it possible to design lower granularity services that closely correspond to atomic business functions. This reduces the problem of managing complex document standards to a more manageable task of standardizing service interfaces (i.e. service APIs) for a given application domain. Such domain-specific service APIs are conceptually similar to APIs that are used extensively in programming environments. The key difference between document-centric and service-centric interoperability models is that service APIs can be designed to minimize interdependencies by encapsulating message data structures and externalizing stable service interfaces that constitute a contract between the service provider and the service consumer. Regarding domain standardization as a software project rather than an effort to define message structures for data interchange enables the deployment of proven software development methodologies resulting in a more stable and flexible specification. As is the case with other complex software projects, the ability to design standard specifications from software modules (i.e. services and service operations) that can evolve independently enhances the flexibility of the specification and provides a solid basis for accommodating future requirements.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the support of the Grant Agency, Czech Republic - GACR Grant No. GA406011, P403/11/0574 and of the Research Centre for Human Centered Technology Design at the University of Technology, Sydney, Australia.

REFERENCES

Arkin, A., S. Askary, et al. (2007). "Web Services Business Process Execution Language (WS-BPEL)." OASIS (www. oasis. org), Version 2.

Brunnermeier, S. B. and S. A. Martin (2002). "Interoperability costs in the US automotive supply chain." Supply Chain Management: An International Journal 7(2): 71-82.

Bussler, C. (2001). "B2B Protocol Standards and their Role in Semantic B2B Integration Engines." Bulletin of the Technical Committee on Data Engineering 24(1): 3-11.

ebXML. (2007). "ebXML - Enabling A Global Electronic Market." Retrieved 9 December 2007, 2007, from http://www.ebxml.org/.

European Article Number (EAN) and the Uniform Commercial Code (UCC). "Global Standards One

(GS1)." Retrieved 02-04-2011, 2011, from http://www.gs1.org/.
Feuerlicht, G. (2005). "Design of service interfaces for e-business applications using data normalization techniques." Information Systems and E-Business Management 3(4): 363-376.
Feuerlicht, G., Meesathit, S. (2004). Design framework for interoperable service interfaces. 2nd International Conference on Service Oriented Computing, New York, NY, USA, ACM Press.
Feuerlicht, G., Wijayaweera, A. (2007). Determinants of Service Resuability. 6th International Conference on Software Methodologies, Tools and Techniques, SoMet 06, Rome, Italy, IOS Press.
Glushko, R. and T. McGrath (2008). "Document engineering: analyzing and designing documents for business informatics and Web services." MIT Press Books 1.
HL7.ORG. (2007). "Health Level 7." Retrieved 10 December 2007, from http://www.hl7.org/.
Kabak, Y. and A. Dogac (2010). "A survey and analysis of electronic business document standards." ACM Computing Surveys (CSUR) 42(3): 1-31.
Legner, C. and B. Lebreton (2007). "Business interoperability research: Present achievements and upcoming challenges." Electronic Markets 17(3): 176-186.
Legner, C. and T. Vogel (2008). "Leveraging Web Services for implementing vertical industry standards: a model for service-based interoperability." Electronic Markets 18(1): 39-52.
McAfee, A., M. Bettiol, et al. (2007). Electronic Hierarchies and Electronic Heterarchies: Relationship-Specific Assets and the Governance of Interfirm IT, Citeseer.
OASIS. (2011). "Universal Business Language (UBL)." from http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl.
Open Application Group. (2010). "CIDX." from http://www.cidx.org/.
OTA. (2010). "OTA Specifications." Retrieved 6 May 2010, from http://www.opentravel.org/Specifications/Default.aspx.
Sabre. (2007). "Sabre Holdings :: World Leader in Travel Distribution, Merchandising, and Retailing Airline Products." Retrieved 10 December 2007, from http://www.sabre-holdings.com/.
The Open Applications Group. (2010). "Open Applications Group Integration Specification." Retrieved 02-04-2011, 2011, from http://www.oagi.org/oagis/9.0/.
UN/CEFACT. (2011). "United Nations Centre for Trade Facilitation and Electronic Business." Retrieved 02-04-2011, 2011, from http://www.unece.org/cefact/.
UN/EDIFACT. (2010). "United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport." from http://www.unece.org/trade/untdid/down_index.htm.
WC3. (2007). "SOAP Specifications." Retrieved 6 May 2008, from http://www.w3.org/TR/soap/.
www.ics.uci.edu. (2002). "Architectural Styles and the Design of Network-based Software Architectures." Retrieved 31 January 2008, from http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.
www.rosettanet.org. (2007). "RosettaNet Home." Retrieved 9 December 2007, from http://www.rosettanet.org/cms/sites/RosettaNet/.