

E-Business Standards

First Steps Towards Scalable Interoperability

Background - EDI

Automation of collaborative business processes has been proven to enhance the competitive advantage of participating trading partners. However, automation to date has been limited to a few key high-volume trading relationships between large enterprises. These connections have typically been built using EDI or other competing and non-interoperable e-business standards. Although EDI has been successfully used for over 20 years, some key limitations have prevented a more widespread adoption:

- **Weak semantics** - EDI is a standard for message syntax but not message semantics. In other words, EDI defines the structure (format) of a message but not the content. Although some standard code-lists are recommended, there is no mechanism to enforce a semantic standard. This means that different project teams may have different interpretations of what should go in a particular field. The consequence is that an EDI integration between company A and B often cannot be re-used without change for company C.
- **Not extensible** - B2B documents often need to be extended to suit particular industries. EDI, unlike XML, is a positional syntax standard (an element is defined according to its position in the document rather than by a label). The consequence of this is that a trading partner that does not specifically cater for the extensions cannot usually understand extended documents. So, again, the integration between A and B is not re-usable for C.
- **No process** - Transmission of a particular document (e.g. a purchase order) is only part of a bigger collaborative process (e.g. requisition to payment process). Unfortunately, EDI does not provide a framework to define and manage standard processes. The consequence is often that A and B agree on an arbitrary process (for example that an order response is a necessary pre-requisite for a change order) that is incompatible with C (who cannot issue order response documents).
- **No registry** - In order to trade electronically with B, A needs to know some information about B. Which document format to use? Where to send it? How to send it? Which processes does B support? A needs to maintain all this information about B in his EDI subsystem - and keep it updated whenever B changes something. If there are 1000 trading partners in a community, each one will have to maintain data about the other 999 - a total of 999,000 entries! A much more scalable approach is to provide a central registry service where each trading partner maintains their own profile and can search for the profile of others. Unfortunately EDI provides no framework for registry services.
- **Proprietary messaging** - A business document needs to be transported to its destination using some kind of electronic messaging protocol. EDI uses proprietary VANs (Value Added Networks) to manage the physical delivery of documents. Lets say A trades with B over a VAN provided by X. But C is part of an EDI community using a VAN provided by Y. If VAN X does not interoperate with VAN Y then A cannot trade with C. In addition to this limitation, VANs typically charge both trading partners a fee per transaction.

All these limitations mean that integration using EDI is un-scalable. A business manager who is developing a supply chain integration strategy must understand these limitations and look for alternative solutions if integration costs are to be minimised.

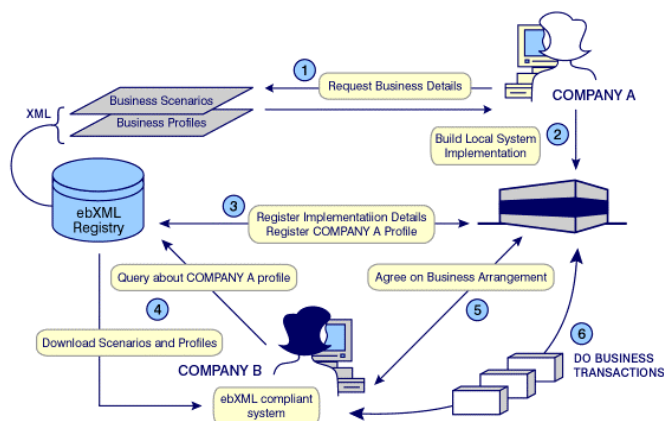
ebXML - The Business Process Standard

The good news for the supply chain manager is that there are a lot of initiatives designed to address the limitations of EDI. The bad news is that there are too many of them - and most of them are an incomplete solution. Should you use Microsoft's BizTalk? What about OBI? And then there is RosettaNet, OAG XML, cXML, XCBL, and at least 50 other XML "standards". What about electronic Marketplaces and Exchanges? The press is raving about "WebServices" - surely that is the answer?

The truth is that there are a lot of very good initiatives by various vendors and associations but there is only one framework that has wide support, is genuinely open, addresses all the limitations of EDI, and is created by international standards organisations. That framework is ebXML - a collection of standards which together provide a framework for scalable e-business collaborations. The following paragraphs provide a high-level overview of ebXML (for more details, visit www.ebxml.org).

- **Process standards** - ebXML provides a language for describing business processes called ebXML-BPSS (Business Process Specification Schema). A trading partner describes his business process capabilities using BPSS and the result is called a CPP (Collaboration Protocol Profile). Two trading partners may have different CPPs and the overlap of the two defines joint collaborative capability. This agreement between two trading partners is called a CPA.
- **Registry based** - ebXML provides a central registry where trading partners can publish their profiles. A profile includes; names & identifiers ("white pages"), categorisation - geographic and by good/service ("yellow pages"), and finally the electronic capability in the form of CPPs ("green pages"). Registries also provide a search interface and various business services such as transformation libraries and CPA calculators. The relevant standards are ebXML-RIM (registry information model) and ebXML-RSS (registry service specification).
- **Strong semantics** - ebXML provides a solid framework for common understanding of the meaning of message content. This framework is called ebXML-CC ("Core Components"). A core component is a syntax neutral description of a data element such as an address.
- **Open messaging** - ebXML provides a messaging framework for the secure and reliable delivery of business documents over the Internet. The messaging standard is called ebXML-MS ("message service").

The diagram below shows these standards working together in the high-level ebXML use case. Steps 1 to 3 describe the registration process (company A says who they are and how to trade with them). Steps 4 and 4 describe the discovery and binding process (company B searches for company A and works out the common capabilities). Step 6 is the actual exchange of business transactions.



There is only one problem with ebXML - it is not yet complete. There are two significant components still in development:

- **Reference processes** - Although ebXML provides a language for describing processes (BPSS) and a framework for discovering and negotiating common capabilities (CPP / CPA), it does not yet provide some actual standard reference processes. UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) is currently working on a specification of standard business processes.
- **Reference documents** - Although ebXML provides the core component library of business elements, these are (by design) syntax neutral. To be usable as an electronic business document (e.g. a purchase order), an XML or EDI implementation of core components is required. OASIS

(Organisation for the Advancement of Structured Information Standards) is co-author of ebXML with UN/CEFACT and is currently developing UBL, the "Universal Business Language". UBL is an XML schema based implementation of ebXML core components. Rather than starting with a blank page, the UBL committee chose an existing standard (xCBL) as the foundation for UBL.

Web Services – The Web Transaction Standard

"Web Services" are hot news at present. But what are Web Services and how are they different to ebXML? It is tempting to regard these two technologies as competitive because, at first sight, they seem to be addressing the same need. The truth is that ebXML is an example of a Web Service. It is just that ebXML is very focused on scalable and secure automation of business to business collaborative processes whilst Web Services is a more general framework for automation of any web transaction.

Standard	ebXML	Web Services
Background	EDI and B2B collaboration	The internet / web pages
Objective	Business process automation.	Web transaction automation.
Focus	Business – strong focus on process and semantics	Technology – focus on transaction and transport.
Created by	UN/CEFACT & OASIS standards bodies.	Vendors (IBM, Microsoft etc) – SOAP has been ratified by W3C to become an Internet standard.
Registry Standard	ebXML RIM 2.0 & ebXML RSS 2.0 (v 3.0 in development)	UDDI v2.0 (v 3.0 released for comments)
Process Standard	ebXML BPSS plus reference processes	Proprietary – XLANG (Microsoft), WSFL (IBM).
Service description	EbXML CPP & CPA	WSDL
Semantic Standards	Core Components, UBL	None
Message Transport	EbXML MS2.0 (based on SOAP – with security & reliability added)	SOAP, SOAP-RP (reliability) in development.

The **Web Services** framework has the Internet as its background and is focused on automating what have previously been manual interactions with websites. For example a currency exchange site requires someone to enter the URL in a browser and type in a value and from/to currencies. The website then calculates and displays the answer. To provide this same capability as a "Web Service" means that an application can send an XML document to the URL containing the value and from/to currencies. The Web Service would then respond with an XML document containing the converted value. A description of how to invoke this currency exchange service would be downloadable from a UDDI (Universal Description, Discovery, and Integration) registry as a WSDL (Web Services Description Language) document. Different currency conversion service providers would implement their own interface and accompanying WSDL. So a client application would need to be modified to work with a different provider.

By contrast, **ebXML** has EDI as its background. As a result, it has focused on addressing the issues that prevented EDI from scaling. If the currency calculator described in the previous paragraph were implemented as an ebXML process, the difference is that there would be a standard for the semantics and structure of the XML document used to invoke the service in order to ensure that a client application could invoke a currency calculator from any service provider without change. More importantly, if the service required a complex choreography of interactions (e.g. quote, response, order,

acknowledgement, etc) then ebXML provides a framework for process interoperability whilst Web Services does not.

So, if you want to automate a complex business process in such a way that it is scalable across a trading community, then look to ebXML. If you want to automate a simple web transaction then there is no need to complicate your life with ebXML to look so the simpler Web Services approach.

As time progresses it is likely that ebXML and Web Services will converge. In the meantime, corporate IT managers and supply chain managers need to understand both standards and use them appropriately.

The Real World – What Is Practical Today?

Development of standards is just the first step in creating a physical B2B integration infrastructure. Vendors such as SAP, WebMethods, Oracle, BEA, etc then need to release products that are compliant against the standards. Finally an end user can implement business solutions using these compliant applications.

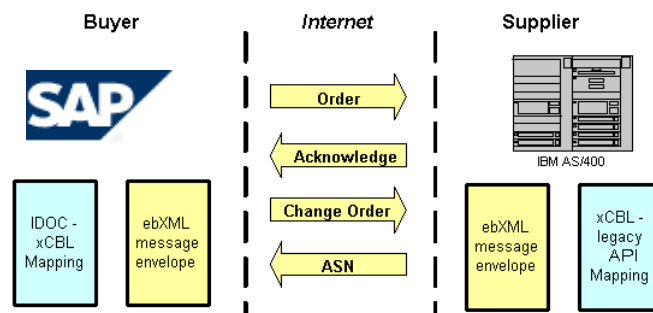
Our supply-chain manager is in a quandary. He has some real projects that need to be started now. But EDI is not scalable. A Babel of XML "standards" have evolved which bring some answers but a lot of confusion. The ebXML framework that promises to bring order to the chaos is not yet finished. So what to do?

The answer is that, once armed with a solid understanding of the issues described in the previous sections, the supply-chain manager is in a position to define a solid corporate strategy for B2B collaboration based on ebXML principles.

Once the strategy is in place, any decisions on infrastructure investment or specific integration projects can be taken in the context of the strategy. So a project can go ahead today as long as it is a step in the right direction by being compliant with one or more ebXML principles.

Lets consider the example of a large buyer with SAP as the back-office system who wants to automate the order process with a medium sized supplier who runs a custom legacy back office on AS/400. The obvious approach for the SAP user might be to send IDOCs (a SAP proprietary format) to the supplier using SAP Business Connector reliable messaging (a WebMethods proprietary protocol). The obvious approach for the supplier with the legacy system might be to expose his legacy system API as a Web Service and invent his own WSDL service description that closely matched the API.

Both of these solutions would work for the specific project but are very unlikely to be re-usable with other trading partners. A far better approach would be to use xCBL or OAG documents (since UBL is not yet ready) and ebXML messaging. This involves a little extra work on both sides but will be much more re-usable. It is also a step in the direction towards full ebXML compliance.



The supply chain manager will not be alone in his support for ebXML. The Australian government (via the National Office for the Information Economy – NOIE) and Standards Australia have endorsed ebXML as the preferred B2B standard. Furthermore, NOIE has taken steps towards the implementation of an ebXML Registry service and an ebXML messaging toolkit for use by Australian businesses.