

ebXML, an open standard.

Article for the December 2003 edition of the www.LinMagAu.com.au online magazine.

Revision : 1.2 DRAFT DRAFT DRAFT

Sacha Schlegel

E-Mail: sacha@schlegel.li

Web: <http://www.schlegel.li>

November 2, 2003

Contents

1	About the author	1
2	Before we get started	1
3	Open standards	2
4	ebXML	2
4.1	The ebXML Collaborative Business Processes	4
4.2	The Core Components	5
4.2.1	The Universal Business Language	5
4.3	The ebXML Collaboration Protocol Profiles and Agreements	6
4.4	The ebXML Registry/Repository	6
4.5	The ebXML Messaging System	7
4.6	What is missing	7
4.6.1	Enterprise Application Integration - MOM	8
5	Conclusion	9
6	Further Resources	9

1 About the author

Sacha Schlegel is currently doing a Masters by Research in Computer Science at Curtin University, Perth, Western Australia. The thesis looks at the ebXML Collaboration Protocol Agreement formation process. Sacha is into ebXML specifications since about 3 years. Sacha became an Individual Member of OASIS in October 2003 to contribute to the ebXML project.

2 Before we get started

Before I start with this article I want to alert the reader that different words, such as open, standard, and free have different meanings to different people. So please read with caution. On a similar note often there is not just one solution or an approach is not simply good or bad. If you want to find more detailed information, please follow the given links throughout the article.

New terms will be introduced throughout the article which, due to the fact that this article introduces ebXML, those introduced terms are not explained enough in this short article. Again please consult some resources from the list given at the end of the article to find more information. Parts and areas of ebXML are left out for simplicity reasons; there are whole books on ebXML.

What is provided along the article are links to open source projects, which the author thinks might be interesting to check up, if you are looking for open source solutions in the various areas.

3 Open standards

OASIS is an organisation which provides open standards. The process how these standards or specifications are worked out is done in an open manner. Open meaning that the work being done is open to the public and the public is welcomed to join. In the OASIS case, for example, all email archives of the various Technical Committees are readable by the public.

It is fair to say that OASIS is not fully open as there is a yearly membership fee¹ and because of that not everyone (in particular open source projects without funding) can participate in a standardisation process.

But it is important that the ebXML specification, for example, are provided free of charge and there are no royalties or fees associated with the use of the specifications.

As in the small world (personal, daily life) having issues with proprietary document formats, which are not accessible among different applications so is interoperability a big issue in the big world (company business life). If commercial products, or open source applications are based on open standards or at least support open standards, then interoperability among different applications is much more likely. The participation of big companies in OASIS in open standards shows that companies have to support open standards.

Open Source and Open Standards seem to be a good team as Open Source projects generally have the consensus to support open standards to allow interoperability between different implementations. Often, the technical challenge to allow interoperability, or the need to overcome constraints and limitations of proprietary software, might be a reason why open source project embrace to support open standards. One can think of why profit-driven companies are cautious with interoperability as that would allow their customers to switch to another software provider much more easily.

Software customers underestimate their power and must ask software vendors for support of open standards in their products. Software Vendors, normally, do listen to their customers otherwise customers will change to another software vendor.

So lets look at ebXML, an open standard.

4 ebXML

ebXML² stands for electronic business XML (Extensible Markup Language³). ebXML was a joint initiative by the Organisation for the Advancement of Structured Information Standards (OASIS)⁴ and the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT)⁵ and the project officially ended in May 2001⁶.

ebXML provides a framework for global electronic business, mainly for the business to business (b2b) environment. ebXML provides a suite of specifications which outline the ebXML framework. ebXML builds upon technologies such as XML, HTTP (Hypertext Transport Protocol)⁷, SOAP⁸ and knowledge gained from previous electronic business projects such as EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport)⁹. Most specifications provide an XML Schemas to structure ebXML artefacts, with the intention that XML instance document will be executed in a runtime environment.

So what is ebXML?

ebXML is a set of loosely coupled components, such as the ebXML Business Process, ebXML Messaging Service, ebXML Registry/Repository, ebXML Collaboration Protocol Profile and Agreement, and ebXML Core Components. ebXML took a top-down approach with analysing collaborative business processes between trading partners at a higher level and then working down towards all the details of how to exchange a concrete message.

Why do we need ebXML?

¹US\$ 250 for an Individual Member, US\$ 1'000 for NON-Profit organisation and US\$ 2'750 for organisations with more than 10 employees and US\$ 5'750 for organisations with more than 10 employees, as of October 2003

²Homepage at <http://www.ebxml.org>

³Homepage at <http://www.w3c.org/XML/>

⁴Homepage at <http://www.oasis-open.org>

⁵Homepage at <http://www.uncefact.org>

⁶Such an ambitious project is never finished and so both organisations are still working and improving the ebXML specifications.

⁷<http://www.w3c.org/Protocols>

⁸<http://www.soaprpc.com>

⁹<http://www.unece.org/trade/untddid/welcome.htm>

The driving force in electronic business are the big organisations which want to connect with partner's information systems. Big organisations are specially interested in hooking up all their Small to Medium Enterprise (SME's) customers or suppliers into their main system. With supporting an open standard, an organisation potentially can reduce the support of many different, partly proprietary technologies and concentrate on one open standard. With a wider adoption of open standards interoperability should be achievable. It was always a goal of ebXML to provide a solution which allows SME's to be part of the big game.

Figure 1 shows an ebXML scenario, which makes it easier to pick up the concept of ebXML. The example is taken from the Technical Architecture Specification ¹⁰

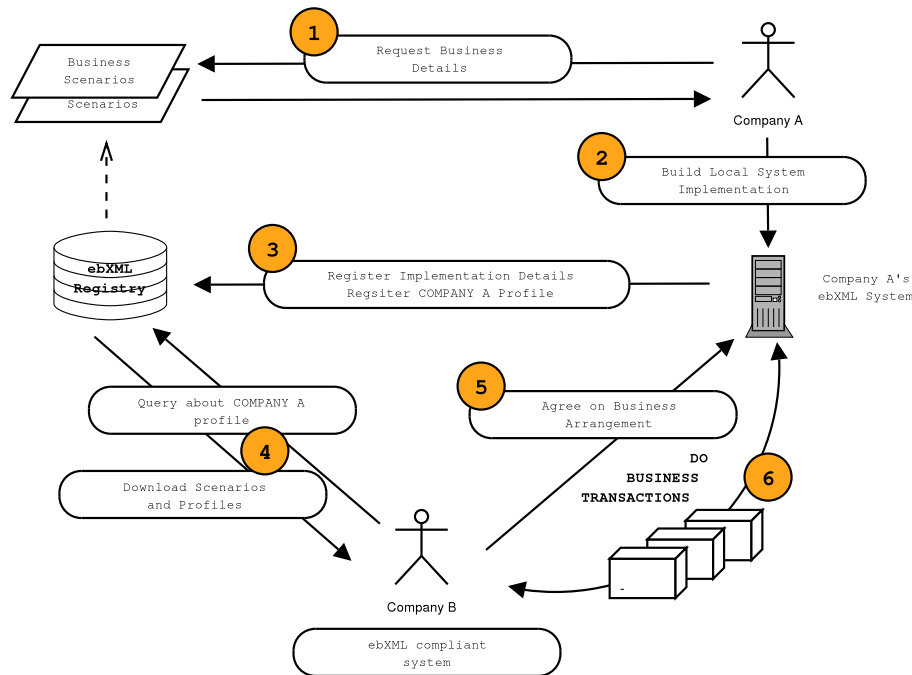


Figure 1: ebXML Overview (Adapted from the ebXML Technical Architecture specification)

The example shows how organisations prepare for ebXML, search for new trading partners and then engage in electronic business.

1. Company A browses the ebXML registry to see what is available online. At best, company A can reuse all the existing business processes, documents, and core components common to its industry that are already stored in the ebXML registry. Otherwise company A designs the missing parts, stores them in the ebXML registry and makes them available for its industry partners.
2. Company A decides to do electronic business the ebXML way and considers implementing a local ebXML compliant application. An ebXML Business Service Interface (BSI) provides the link between the company and the outside ebXML world. The company has to create a Collaboration Protocol Profile (CPP) which describes the supported business process capabilities, constraints and technical ebXML information such as choice of encryption algorithms, encryption certificates and choice of transport protocols.
3. Company A submits its CPP to a ebXML registry. From that point on, company A is publicly listed in the ebXML registry and is likely to be discovered by other companies querying for new trading partners.
4. Company B is already registered at the ebXML registry and is looking for new trading partners. Company B queries the ebXML registry and receives the CPP of company A. Company B then has two CPP's: Company A's CPP and its own. The two companies have to come to an agreement on how to do business, which is called a Collaboration Protocol Agreement (CPA) in the ebXML terminology. Company B uses an ebXML CPA formation tool to derive a CPA from the requirements of the two CPP's.

¹⁰Can be found at <http://www.ebxml.org>

5. In this scenario company B communicates with company A directly and sends the newly created CPA for acceptance to company A. Upon agreement of the CPA by company A, both companies are ready for electronic business.
6. The companies then use the underlying ebXML framework and exchange business documents conforming to the CPA. This means that both companies follow the business processes defined in the CPA.

This scenario demonstrated the various ebXML components, which will be introduced in the next sections.

4.1 The ebXML Collaborative Business Processes

Its all about collaborative business processes. Business happenings *between* two companies is called a collaborative business process and has a central role in the ebXML framework. Internal, inhouse business processes are not part of ebXML but collaborative business processes which go beyond organisations boundaries are. Section 4.6.1 shows how collaborative business processes can be tied together with internal business process, but again, that is not part of ebXML.

An ebXML collaborative business process allows for binary collaboration and for multi party collaboration. In a multiparty collaboration more than two parties engage in a collaborative business process. A multiparty collaboration scenario might be where we have a buyer party, a seller party, an insurance party and a shipping party.

What is a collaborative business process? A collaborative business process provides the choreography of business documents, such as “Invoice”, “Purchase Order”, or a “Notification of a Purchase Order”. The choreography provides the sequencing, the timing of the business documents exchanged, describes the message flow between parties. Business messages encapsulate business documents and include further information, such as the receiver and sender of the message for routing purposes. Very often these business documents are the electronic version of real world business documents from the paper world.

Who creates these collaborative business processes? The ideal case would be when industries, such as the auto, retail, furniture, computer hardware, or computer software industries sit down and analyse their collaborative business processes within their industry. ebXML provides worksheets how to get from the first analysis of collaborative business processes to the fine grained collaborative business process XML instance documents. The ebXML Business Process Specification provides the XML Schema for collaborative business processes. Of course each industry does its collaborative business processes a little bit different. RosettaNet, for example, is a very successful vocabulary for collaborative business processes in the computer electronics industry. Once an industry, or individual companies have defined their common collaborative business processes, they store them in an ebXML Registry/Repository. All members of that industry then can reuse the concrete collaborative business process to engage in electronic business.

What becomes obvious here is that there is an analysing, modelling and design phase and a runtime phase. In the analysis, modelling and design phase the collaborative business processes have to be defined including their corresponding business documents. ebXML uses the UN/CEFACT Modelling Methodology (UMM) which uses UML (Unified Modelling Language) to model the collaborative business processes. To model the choreography of an ebXML collaborative business processes UML activity diagrams are used.

The ebXML Business Process Specification provides a UML class diagram of a collaborative business process to demonstrate how multiparty collaborative business processes are modelled. The ebXML Business Process Specification is a nice read as it provides a step by step description of the UML class diagram which is also represented as an XML schema. This means that if an industry, or an organisation model their collaborative business processes they will create a new XML ebXML Business Process Specification Schema instance document, such as a “Request Purchase Order” collaborative business process.

There are currently undertakings at OASIS and UN/CEFACT to enhance the ebXML Business Process Specification beyond version 1.1.

The business documents which are exchanged during a collaborative business process are of importance as they carry important business information. The next section introduces business documents or rather what business documents are composed of.

4.2 The Core Components

Once the collaborative business process with the choreography of business documents is analysed the analysis of the real business documents has to take place. Because ebXML provides a framework of electronic business for any industries, it has to cater for any different types of business documents. The idea of breaking down business documents into smaller parts is the idea behind the Core Components. The Core Components are the building blocks for every business document.

One layer above the core components are the aggregated components and one layer above are Business Information Entities (BIE), which are composed of core components or aggregated components. Finally a business documents is composed of BIE's. This layering encourages reuse of BIE's and their aggregated components and core components.

What is an "Address"? What is an "Invoice"? What is a "Purchase Order"? What does "delivery date" mean? These questions highlight a fundamental problem which is called the problem of semantics. Semantics, the science of meaning. The problem of semantics is not a problem of ebXML only but for most other projects as well. It is not easy to provide an answers to those questions.

So what is a "Purchase Order"? When asking 5 different people, you probably get 5 different answers. To agree on what a "Purchase Order" means is very important as a "Purchase Order" is used in different industries, in different countries and probably in different ways. The context and the localisation of an BIE adds information about the semantic. So an information system needs to know, not only the structure (XML schema) of a "Purchase Order" but also what it means.

If organisations are doing business they need to have the same understanding otherwise they run quickly into misunderstandings and misunderstandings in business can become very expensive. Nevertheless, business documents, such as invoice, have to be engineered. The various core component documents, such as "Core Component Overview", "Core Component Discovery and Analysis", "Guide to the Core Components Dictionary", "Naming Convention for Core Components", "Document Assembly and Context Rules", "Catalogue of Context Drivers", "Core Component Dictionary", and "Core Component Structure" ¹¹ promote for harmonisation between different "Invoices". Looking at the 5 different "Invoices" there will be commonalities among these 5 "Invoices". It is then important to localise those commonalities and provide general "Invoices". A context is associated with a core component and allows to distinguish the concrete usage of the core component in that specific context. This encourages reuse and should make it easier to engineer business documents. A business document creation tool might allow point and click to engineer a business document from a list of BIE's, retrieved from an ebXML Registry/Repository.

As the Core Components are the idea how to do it there was no concrete implementation available. The Core Components series of technical reports provide guidelines how to get to core components and how to engineer business documents. Thats were the OASIS Universal Business Language (UBL) Technical Committee stepped in.

4.2.1 The Universal Business Language

"The purpose of the UBL TC is to develop a standard library of XML business documents (purchase orders, invoices, etc.) by modifying an already existing library of XML schemas to incorporate the best features of other existing XML business libraries. The TC will then design a mechanism for the generation of context-specific business schemas through the application of transformation rules to the common UBL source library. UBL is intended to become an international standard for electronic commerce freely available to everyone without licensing or other fees.'

The UBL Library Content UBL Subcommittee, under Australian Chair of Tim McGrath, is providing the real bits and pieces for UBL and expects to provide a final version 1.0 in early 2004. The OASIS UBL Technical Committee is pushing for ISO¹² certification.

Once we defined the collaborative business processes and the business documents we are ready create a concrete organisation profiles.

¹¹ Can be found at <http://www.ebxml.org>

¹² Homepage at www.iso.org

4.3 The ebXML Collaboration Protocol Profiles and Agreements

In the ebXML framework two companies run business to business based on an ebXML Collaboration Protocol Agreement (CPA) but first each company has to provide an ebXML Collaboration Protocol Profile (CPP)¹³.

A CPP provides all necessary information how a particular party intends to do electronic business. It starts off with some general information about the party itself, such as its ID, a possible website for more information etc. The CPP then references a collaborative business process which the party supports. Along the reference of the collaborative business process it also has to provide the role it wants to play in the collaborative business process. In a “Purchase Order Request” collaborative business process for example, one party typically has to play the “buyer” role and one party the “seller” role. Because we are talking about electronic business the reference to the collaborative business process is not enough. For each business document which gets exchanged during the execution of the collaborative business process some additional information must be provided. Information such as the selected transport protocol (for example HTTP or SMTP), if a secure transport protocol is used (for example SSL), if encryption is necessary and if which encryption algorithm is used, including the necessary public keys or certificates. The CPP also references the packaging of business documents, typically their XML schema which allows to validate an incoming business document to a certain extend.

Once each party has such a CPP, each party registers it at an ebXML Registry/Repository. This allows another organisation to query for a potential new trading partner. The process to get from two CPP’s to a CPA is called the CPA formation process. In this process any problems of incompatibility between how two different parties intend to do electronic business have to be localised and fixed. For example, if the buying party intends to send the “Purchase Order Request” with HTTP as its transport protocol and the selling party intends to receive the “Purchase Order Request” with SMTP as its transport protocol, the two parties never will be able to exchange that business document.

Once such a Collaboration Protocol Agreement is formed both sides get such a copy. The CPA will be used to “install” a new electronic business agreement in their ebXML Message Service Handler. This message Service Handler, then knows, based on the information found in the CPA, exactly how business documents have to be exchanged, eg which transport protocol to use, which encryption to use, which key to encrypt a message, if SSL is necessary to establish a secure transportation and more. The big puzzle starts to make sense here, doesn’t it?

To create a valid CPP is not that easy and tools will be needed to make it easy, especially for SME’s.

So far we found collaborative business processes, core components, CPP’s and CPA’s. To support reuse of collaborative business processes, core components, these documents have to be stored in a repository with adequate access for all. CPP’s also will be stored in a repository for others to be queried. The ebXML project provides the necessary specifications for the ebXML Registry/Repository.

4.4 The ebXML Registry/Repository

The example at the beginning of this article showed how the ebXML Registry/Repository is used. The ebXML Registry/Repository does not only store data, like the core components, collaborative business processes, business documents, and CPP’s but also meta data about those data. It is then possible to query for a potential trading partner by defining, for example, that we are looking for a CPP which supports collaborative business process “Purchase Order Request” and the party is located in Australia.

The registry part of ebXML Registry/Repository provides an interface to query information of the ebXML Registry/Repository whereas the repository part of the ebXML Registry/Repository is in charge of storing data.

To encourage organisations to use the ebXML framework typically a government could host the ebXML Registry/Repository and thus providing certain services to its national business organisations. That would provide a certain level of trust in the ebXML Registry/Repository. This is combined with some maintenance work to keep the information in the database clean. The organisations themselves are responsible to keep their CPP’s up to date and the industries which define the collaborative business processes and core components have to maintain, enhance, improve their stored information.

Figure 2 points you to an open source ebXML Registry/Repository implementation.

¹³Theoretically the straight creation of a Collaboration Protocol Agreement for two concrete trading partners allows to get a CPA more quickly.

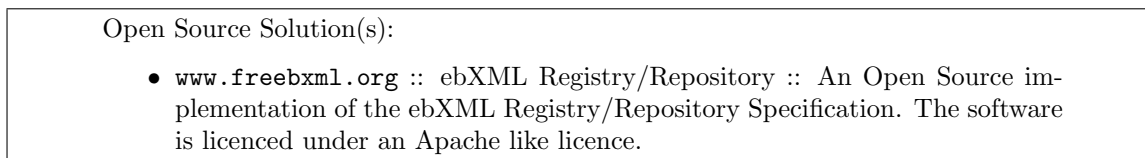


Figure 2: Open Source solution for ebXML Registry/Repository.

4.5 The ebXML Messaging System

To send ebXML messages back and forth an ebXML Message Service Handler (MSH) is necessary. The main functionalities of the ebXML Message Handler are messaging, reliable messaging, security (at various levels such as message level and transport level), persistence (for monitoring and auditing purposes) and more. Messaging is based on SOAP with Attachment over any given transport protocol. Any ebXML Message Service Handler has to be capable of providing the desired transport protocols, security capabilities, desired encryption algorithms etc.

Once two parties achieve to get a final Collaboration Protocol Agreement (CPA), both parties involved will get a copy of that CPA. The CPA then gets installed in a MSH. This is very useful if a party runs electronic business with many different parties. In such a case, the new CPA can be “just” installed in the MSH. The MSH then has to analyse an incoming message to which CPA it belongs to, once that is sorted out the MSH can then handle the message according to the information in the CPA the message belongs to.

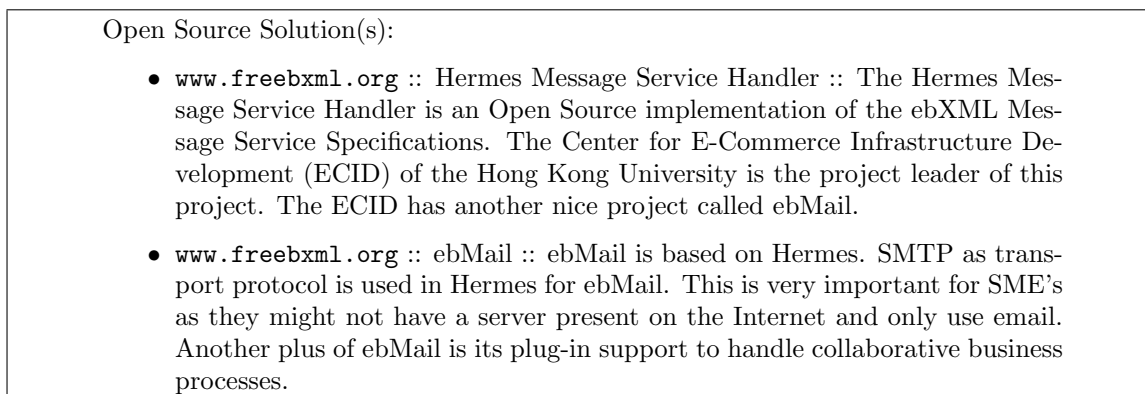


Figure 3: Open Source solutions for the ebXML Messaging.

Figure 3 points you to two ebXML Messaging Service implementations.

4.6 What is missing

To make it short: The full implementation from A to Z in all details of the ebXML stack is what is missing today. Preferable in an open source manner so open source developers can start to build next generation open source business applications, based on an open standard such as ebXML. Not only the execution engines, such as Messaging Service Handler and Business Service Interface have to be ready but also modelling tools to let business analyst model collaborative business processes and produce collaborative business process XML instance documents. There are various commercial applications which implement parts of the ebXML stack.

Integration and interoperability have to be seamless with existing infrastructures. Software vendors have to endorse ebXML in their product suites to allow its customers to be part of the ebXML global electronic business environment. How to bring SME's into ebXML? As soon as SME's software packages, often commercial products such as MyYOB, Quicken or QuickBook support ebXML SME's can start to engage in electronic commerce.

ebXML is not the only framework which wants to provide global electronic commerce but one which was developed in an open manner. Among competing frameworks are the various combination of Web-Services Specifications, such as Business Process Execution Language for Web Services (BPEL4WS), Web Services Description Language (WSDL), Web Services Orchestration (WSOC). Please check each Web Service specification for their usage as some are proprietary specifications.

There are several open source implementations of business applications, such as accounting, general ledger, accounts receivable, accounts payable, payroll and some inventory.

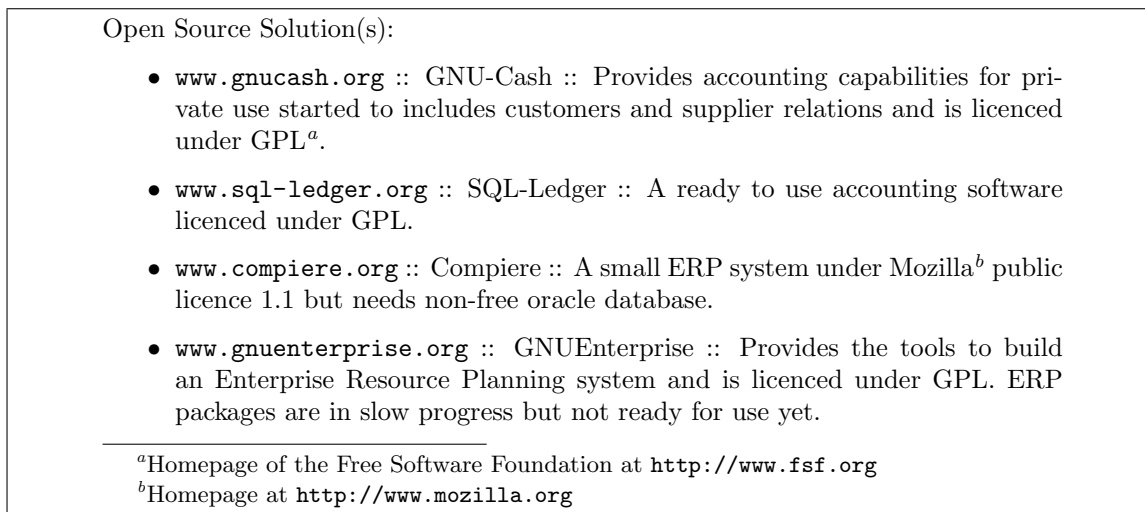


Figure 4: Open Source solutions for Small to Medium Enterprise organisations information systems.

Figure 4 points you to several open source business applications.

4.6.1 Enterprise Application Integration - MOM

Here a really short introduction about Enterprise Application Integration (EAI). Lets assume that a company has an open source Enterprise Resource Planning (ERP) system and the open source Hermes ebXML Message Service Handler. The ebXML Messaging System Hermes would know how to send and receive ebXML messages, how to provide reliability, how to encrypt/decrypt messages but would not know what to do with the business document, a “Purchase Order” for example. Message Oriented Middleware (MOM) is a technology which allows to publish and subscribe to messages, such as the “Purchase Order”. The ERP system is interested in “Purchase Orders”, so it subscribes to “Purchase Orders” at the MOM. Hermes on the other hand, which receives “Purchase Orders” publishes “Purchase Orders” to the MOM. So when Hermes receives a “Purchase Order” it publishes (sends) the message to the MOM and the MOM checks who is subscribed to a “Purchase Order” and thus forwards the “Purchase Order” to the ERP system. What a MOM also provides is a translation service. Business messages between two organisations, or two different applications might have a different document format. With message mapping rules a “Purchase Order” of the from *X* can be transformed to a “Purchase Order” of form *Y* (think XSLT (eXtensible Stylesheet Language Transformations)). Again, hopefully both formats are NOT proprietary but based on an open standard for seamless integration.

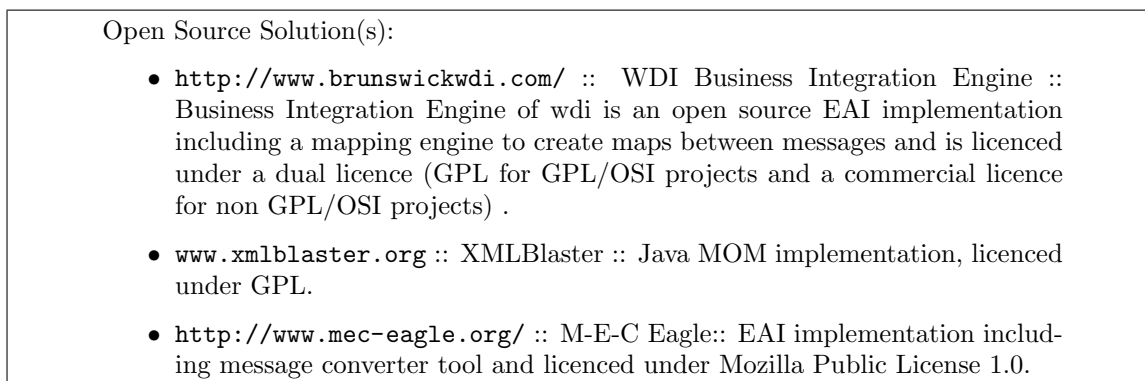


Figure 5: Open Source solutions for Enterprise Application Integration.

Figure 5 points to you open source Enterprise Application Integration implementations.

5 Conclusion

ebXML provides an open framework for global electronic business in the form of good specifications which still actively get updated and enhanced. This young technology needs time to mature. Productive live ebXML systems in the real world show that ebXML gets adopted by companies around the world and the few, very promising, ebXML open source implementations show the growing interest to provide ebXML technologies on an open source basis.

The loosely coupled ebXML components allows to integrate ebXML into other frameworks by deploying one component by one.

6 Further Resources

Here is a list for further web resources where you can get more information about ebXML.

- The ebXML website, of course: www.ebxml.org
- The ebXML Open Source website: www.freeebxml.org
- The ebXML Forum with articles about ebXML: www.ebxmlforum.org
- The ebXML Asia Forum: www.ebxmlasia.org
- The openebXML website: www.openebxml.org
- Critical comments about ebXML: www.rawlinseconsulting.com/ebXML/
- Some various ebXML open source projects at Sourceforge: www.sourceforge.net

If you want to get information about Business Processes have a look at www.ebpm1.org, which is a Website dedicated to the architecture and technologies of Business Process Management Systems.