ebXML

Applying the Unified Modeling Language
to reuse existing business knowledge

The Vision

ebXML's vision is revolutionary and ambitious at the same time: dynamically formulating trading partner agreements -containing everything from security and transport protocols to message formats for doing electronic business. In a world where face-to-face contact is still considered crucial before signing an agreement however, some of ebXML's ideas could be considered far-fetched.

This 18 month initiative, started in 1999, recently came to an end with 2001's May meeting. The deliverables presented at this closure meeting give a comprehensive view of how things should work, but lack implementation guidelines that could get you started with ebXML right now. The ebXML documents are a tad vague and subject to interpretation. This is why the ebusiness Transition work group (ebTWG) was brought to live to fine tune and subsequently finish the work started by ebXML.

Mechanism

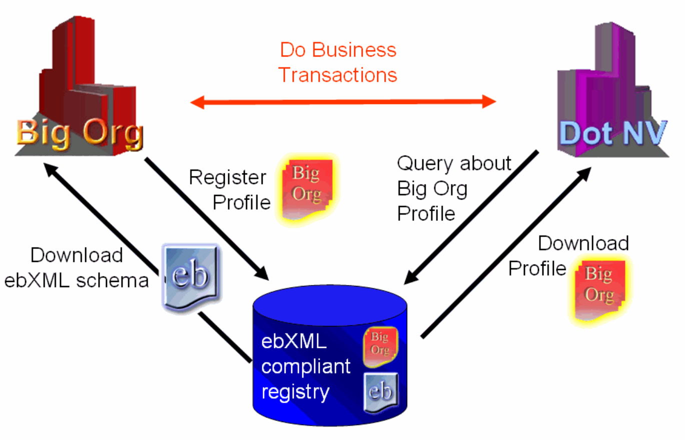
Three important mechanisms are needed by ebXML: firstly, a methodology to describe our business processes (or services) so that others can understand what exactly it is we are offering; secondly, you'll want to register your services in a central registry. Finally a mechanism is needed that allows for the discovery of the registered services. (Does this sound like UDDI to you? -it actually does, we'll come back to this in another paper)

Let's start by examining the first step: explaining what your business does in an industry-neutral language. Everything points in the direction of UML as the language of choice. UML is just a notation however, so we also need some kind of methodology (a way of working). ebXML suggests using the United Nations Modeling Methodology (UMM). UMM offers a number of best practices for modeling the business process using UML.
Let's say we've made our UML model, and are ready to communicate it to the world. But hold on -you're not going to disclose the cookie production process that makes your business stand out of the crowd now are you?

Registering your services

Do not worry; only that part of the business process responsible for communication with the outer world needs to be transmitted to the publicly available registry.

The complete process of creating the trading partner agreement involves a series of 12 steps (not all mandatory), of which the most important are shown in the following illustration. As you can see (starting from the left) we'll first need to download an ebXML schema.



Tools and standards

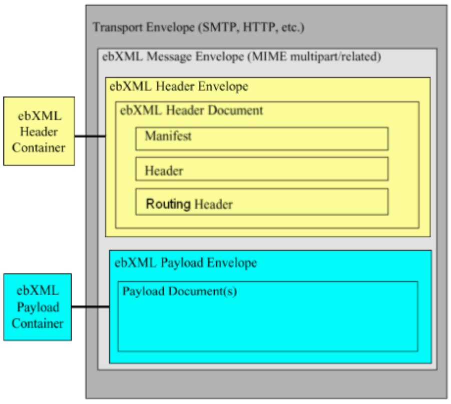
Schemas contain the way in which data is structured in an XML document. Or in other words, schemas contain data models that a parser can read to determine if the XML file is structured accordingly. (A parser uses the schema to effectively validate the XML file)

The next step will thus imply the translation of the UML model into an XML schema. The exact way in which this is done goes beyond the scope of this discussion, but it is useful to know that it can be done by some UML to XML tools. The obtained schemas, bundled with some practical information concerning your company (address, email, etc…) will compose the so-called CPP (collaborative partner profile). This profile will need registration in the registry so that others can find it.

Creating the agreement

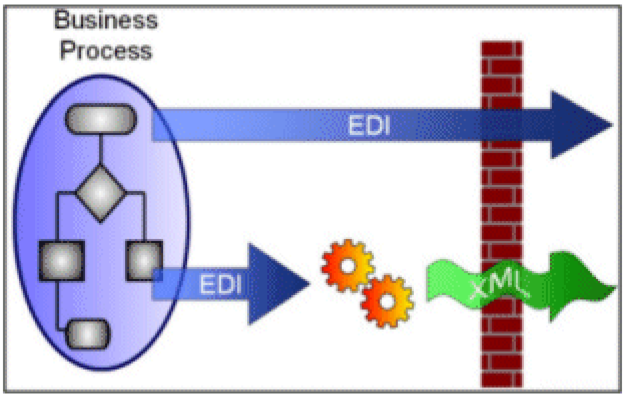
Our company's services are now available for discovery by others. The process of discovery should ideally be automated in such a way that an application can be scheduled to browse the registry after a predetermined timeframe, looking for businesses that offer a certain service at the best conditions. It goes without saying that this automated process will rely on trusted third parties issuing quality labels to the services offered.

When a service is found that fulfills all the necessary preconditions, both companies' CPP's are combined to produce a collaborative partner agreement (CPA). This CPA contains information about the exact way in which applications on both sides of the trading agreement can interface with each other. As soon as the CPA is produced, both trading partners can start doing business electronically, by sending electronic messages to each other. ebXML defines a framework for structuring the documents that are sent between ebXML compliant partners. The message structure is based on SOAP (Simple Object Access Protocol) and uses digital signatures for authentication and SSL for network transmission encryption. The following illustration shows the ebXML message structure.

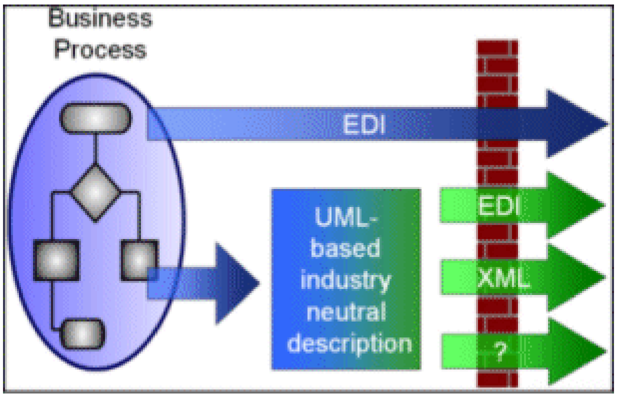


Re-use

One of the primary goals of ebXML is reuse of the existing business knowledge. As a useful example, let's take a company currently using EDI messages, but considering the move to XML. EDI documents contain loads of business knowledge due to the very strong semantic meaning of the different fields in the message. Now, instead of trying to 'map' the EDI message fields to XML. elements (which incidentally also implies a complete loss of the business semantics which were available in the EDI message), wouldn't it be wiser to take a completely different approach that does preserve the business semantics? Following illustration shows how not to proceed:



The solution is quite obvious really, and once again calls upon the UML notation. As the illustration below indicates, modeling your business with UML before producing electronic messages is an ideal way of preserving business knowledge. It is also a great way to make sure your services can be communicated to the outer world even if two years from now, another 'revolutionary' language takes the place of XML.



ebXML is an initiative which has massive backing from vendors and standardization organizations alike, and harbors some very interesting concepts and ideas that could finally put an end to the ever extending proliferation of XML dialects.