3 International Data Standard for Information Exchange

Since the Internet explosion in the mid-90s and XML became a W3C standard in 1998, there's been a rush by standard organisations to develop a global XML-based e-business infrastructure. One such initiative is the ebXML project, jointly developed by UN/CEFACT and OASIS which started in 1999 and involves more than 2,000 participants from 100 organisations across 150 countries.

ebXML is an open XML-based infrastructure that enables global use of e-business information in an interoperable, secure and consistent manner by all parties. It comprises the Registry & Repository Specification, Business Process Schema Specification, Collaboration Protocol and Agreement Specification, Messaging Service Specification and the Core Component Technical Specification. Today, except for Business Process Schema Specification, the other specifications are now an ISO-15000 specification.

While there have been active developments in the area of messaging, and registry and repository, one area that is lacking, until quite recently, is the availability of business document schemas, which is a critical component for the success of ebXML. Realising this 'gap', OASIS started the UBL project in 2001, which attempts to fulfill the promise of XML for business by defining a standard cross-industry vocabulary. Based strictly on the ebXML Core Component Technical Specification (CCTS), it is the first such initiative to do so. The ebXML CCTS is the underlying methodology for developing a common set of semantic building blocks that represent the general type of business data in use today and provides for the creation of new business vocabularies and restructuring of existing business vocabularies.

The paper discusses ebXML in general, the underlying CCTS methodology, the UBL project, its development methodology, deliverables and the future direction of UBL.

Sivakumar Ganesan
Senior Software Engineer, CrimsonLogic Pte Ltd

Kamarudin Bin Tambi
Manager, CrimsonLogic Pte Ltd
Vice Chairman, XML Working Group of IETC

1 Introduction

Electronic Business eXtensible Markup Language, typically referred to as ebXML[1], is a family of XML-based standards sponsored by Organization for the Advancement of Structured Information Standards[2] (OASIS) and United Nations Centre for Trade Facilitation and Electronic Business[3] (UN/CEFACT). The mission of ebXML is to provide an open, XML-based infrastructure that enables electronic trading relationships between business partners in an interoperable, secure and consistent manner.

The vision and model for better e-Business using open standards was created by combining the business knowledge gained from 20 years of EDI-based interactions from UN/CEFACT, with the OASIS web commerce and marketplace expertise of Internet-based companies using XML. This model seeks to move from processes that are essentially paper based and highly labour intensive in configuring and deploying manually, to a world where trading partners can discover each other and then begin to do business electronically by linking their systems using ebXML and the Internet.

Each step of this process is supported and enabled by ebXML through the use of components based on discrete and loosely coupled specifications that are engineered to deliver specific functionality. Each component can be used individually or combined as needed to enable open, low-cost global commerce. The International Organization for Standardization[4] (ISO) has approved the following five ebXML specifications as the ISO 15000 standard, under the general title, Electronic business eXtensible markup language:

- ISO 15000-1: ebXML Collaboration Partner Profile Agreement;
- ISO 15000-2: ebXML Messaging Service Specification;
- ISO 15000-3: ebXML Registry Information Model;
- ISO 15000-4: ebXML Registry Services Specification; and
- ISO 15000-5: ebXML Core Components Technical Specification.

In addition to the five ISO standards, the original ebXML project envisioned one more specification for formally defining standard business processes, known as ebXML Business Process Specification Schema (BPSS). This specification focuses on the collaborations of trading partners and the business transaction activities they perform in the context of those collaborations. The BPSS has been strongly influenced by UMM, a modelling methodology developed by UN/CEFACT, but does not require it.

1 http://www.ebxml.org
2 http://www.oasis-open.org/home/index.php
3 http://www.unece.org/cefact/
4 http://www.iso.org/iso/en/ISOOnline.frontpage

2 Background

To understand the evolution of ebXML, it is necessary to take a look at EDI from which ebXML has borrowed many ideas. Electronic Data Interchange (EDI) is a technology that has supported a large-scale business-to-business e-business infrastructure for many years. EDI predates XML and even the Web. EDI is successfully used in production systems that support large-scale e-business, both in terms of transaction volumes and in economic value. Nevertheless, EDI has not managed to extend its reach outside a limited number of application areas; in particular, it hasn't extended its reach to small- and medium-size enterprises (SMEs). This is commonly attributed to a number of factors:

- The EDI message syntax is cryptic and hard to understand;
- EDI relies on expensive private value added networks (VANs);
- Software and services for EDI are very expensive;
- EDI is tied to batch applications; and
- The EDI world is fragmented in multiple communities.

EDI uses its own syntax to encode messages. The EDI message format uses short alphanumeric codes and has various ways to omit unused message sections to reduce the overall message size. It is necessary to have external code tables and knowledge of the message structure to interpret the message. Message size was certainly an important issue at the time EDI was designed, as bandwidth was expensive. This issue is less relevant nowadays because Internet bandwidth is much less expensive, and because compression technology can reduce message size at transport (rather than application) level.

The EDI syntax is undeniably terse and XML offered advantages over the EDI format because of the support for validation using standard parsers, availability of more and better conversion tools to translate EDI messages to other formats, and XML's superior support for character sets.

XML documents include their own meta-data, in the form of element and attribute names that label information. This allows information in a document to be obtained using meta-data, rather than using positional information. This offers increased flexibility to extend the XML document with additional information without breaking existing applications. Various experts in the EDI community have acknowledged these advantages and agree that, with XML, the EDI syntax has become obsolete.

Another reason as to why EDI has proved to be too expensive is the absence of support for 'common business scenarios' that causes the process of setting up partner agreements to be too expensive.

ebXML aims to overcome these limitations inherent in existing EDI frameworks, by harnessing the power of XML, providing 'common business scenarios' as formal specifications, with a common set of semantic building blocks for business information and a formal mechanism to generate partner profiles and agreements. It utilises the potential of the Internet for automated discovery of partners' capabilities through the registry model. ebXML has adopted platform independent technologies such as SOAP combined with standard security mechanisms, to make it a truly open, process oriented framework that enables low-cost global commerce.

3 ebXML Implementation

Typically, there are three key phases involved by which an organisation participates in business transactions with other partners, based on the ebXML specifications.

3.1 Implementation Phase

In the Implementation Phase, the key component is the ebXML Registry and Repository (ebXML RR). The ebXML RR contains industry defined Business Processes and Scenarios that are applicable to most business transactions. Organisations can choose to extend these processes and add scenarios of their own, if necessary. The ebXML RR also contains profiles for organisations that have already registered themselves for performing ebXML transactions with other trading partners.

For example organisation A, which is interested in doing electronic business as per the ebXML standard, consists of three steps as shown in Figure 1:

- Request Information;
- Implement ebXML System; and
- Publish Business Profile.

The first step is to request the ebXML Specifications (Business Processes, Business Scenarios) and understand them. Once the organisation has understood the specs, it decides which business processes it would like to implement, following which it needs to implement a system in-house based on those standards. It could either build a new system or build on top of an existing legacy system. The whole idea is to expose a system that understands and interacts in ebXML protocol. Once the system is built, the organisation is ready to conduct business with other organisations. To facilitate that, it needs to publish its profile known as a Collaboration Protocol Profile (CPP) to the ebXML Repository for other organisations to discover. A CPP thus enables any organisation to describe its profile i.e. which business processes it supports, its roles in that process, the messages exchanged, the transport mechanism for the messages etc. Once the CPP is published to the ebXML Repository, it will allow other organisations to access it and learn about the capabilities of Organisation A. At any time, Organisation A is free to access its own profile, review and make changes as necessary.

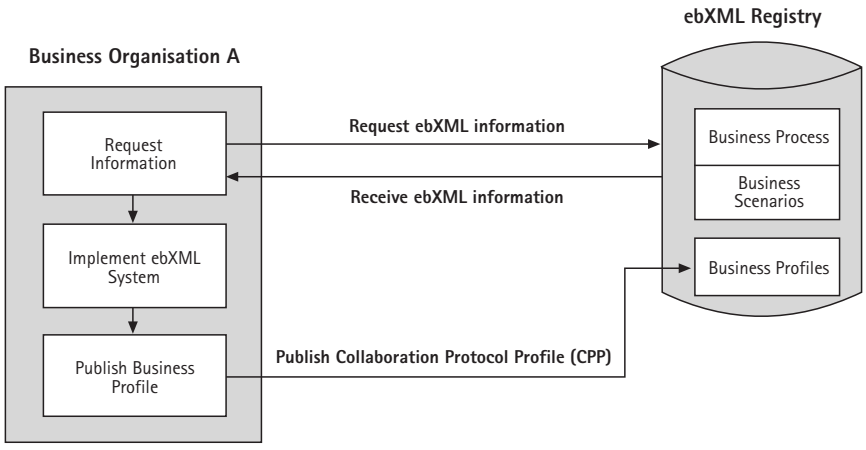

Figure 1: Implementation Phase

3.2 Discovery and Negotiation Phase

Similar to Organisation A, assume that Organisation B has also published its profile as described in the implementation phase. The second phase, Discovery and Negotiation Phase describes how Organisation A prepares to conduct electronic business with a partner Organisation B. The first step that Organisation A does is to retrieve Organisation B's profile information from the ebXML RR. Once Organisation A has the profile of Organisation B, it is in a better position to understand Organisation B's capabilities i.e. whether it supports the business processes that it is interested in, the messages to be exchanged, transport mechanisms, security and reliability of the process, etc. In the real world, businesses always negotiate terms and implement business contracts before conducting any business. ebXML B2B exchange is no different in this aspect. So, the next step for Organisation A is to send over a business contract called a Collaborative Partner Agreement (CPA), to Organisation B. The CPA will be a reflection of the profile (CPP) of both the organisations. Both the organisations can now collaborate on the CPA and refine it to meet the business needs of both the organisations. Finally, both parties accept the agreement. During this phase, it is very likely that key personnel from both organisations will meet in person and make assessments before committing to an eBusiness relationship.

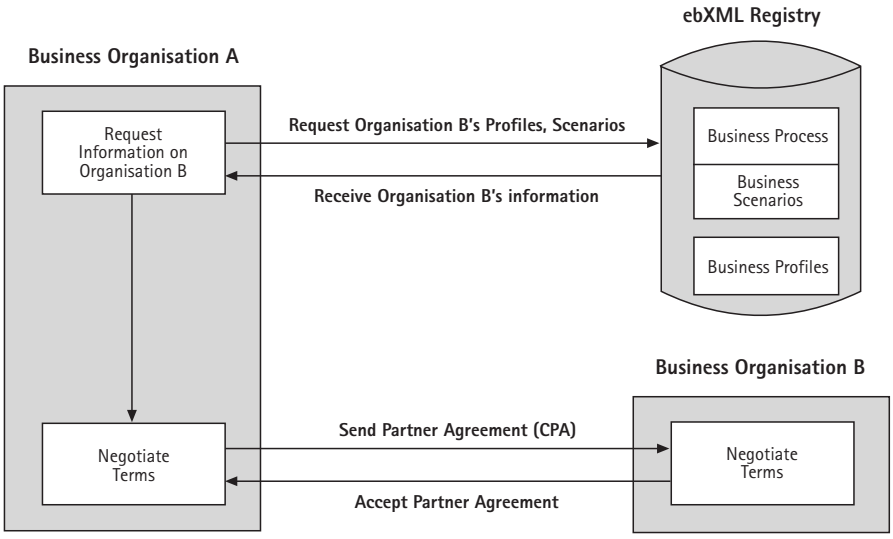

Figure 2: Discovery and Negotiation Phase

3.3 Transaction Phase

The partner organisations are now ready to conduct transactions. A CPA was accepted in the previous phase and the transactions can be conducted in a pre-defined fashion where each business organisation plays a pre-determined role in the transaction. The transactions consist of ebXML messages, which are sent over the standard ebXML Messaging Service.

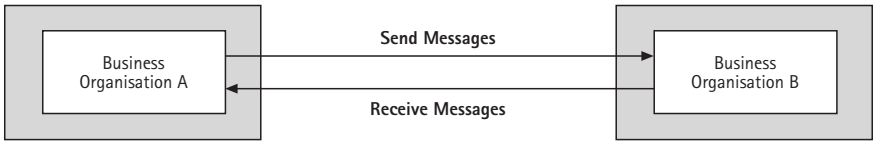

Figure 3: Transaction Phase

4 Core Component Technical Specification

Traditionally, standards for the exchange of business data have focused on static message definitions that have not enabled a sufficient degree of interoperability or flexibility. A new flexible and interoperable way of standardising Business Semantics is required. The Core Component Technical Specification (CCTS) presents a methodology for developing a common set of semantic building blocks that represents the general types of business data in use today and provides for the creation of new business vocabularies and restructuring of existing business vocabularies. It specifies a new approach to the well-understood problem of the lack of interoperability between applications in the e-business arena and provides a way to identify, capture and maximise the re-use of business information to support/enhance information interoperability across multiple business situations.

The CCTS key concepts cover two focus areas - Core Components and Business Information Entities, which are explained below.

4.1 Core Components

The central concept is the Core Component, which is a semantic building block and is used as a basis to construct all electronic business messages. Core Component is divided into four different categories: Basic Core Component, Association Core Component, Core Component Type and Aggregate Core Component.

Figure 4 shows two Aggregate Core Components, **Person. Details** and **Address. Details**. Each Aggregate Core Component has a number of Properties (i.e. business characteristics). The Aggregate Core Component **Person. Details** has four Properties, namely **Name, Birth, Residence** and **Official Address**. The Aggregate Core Component **Address. Details** also has four Properties, namely **Street, Post Code, Town** and **Country**. Most of these Properties are Basic Core Components and represents singular business characteristic and their set of allowed values is defined by a Data Type. The Data Types **Name, Street, Post Code** and **Town** are of the Data Type Text, Birth Date is of the Data Type Date and Country is of the Data Type **Identifier**.

Official Address and Residence are Association Core Components and represent complex business characteristics and their structure is defined by the Aggregate Core Components **Address**.

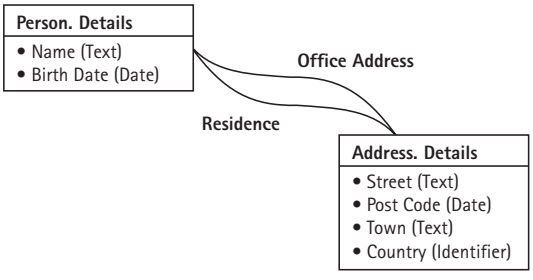

Figure 4: Core Component

Core Component Type forms the basis for all data types. The approved core component types are shown in Table 1 below.

S/No	Core Component Type	Primitive Type
1	Amount	Decimal
2	Binary Object	Binary
3	Code	String
4	Date Time	Date
5	Identifier	String
6	Indicator	String
7	Measure	Decimal
8	Numeric	Decimal
9	Quantity	String
10	Text	String

Table 1: Approved core component types

4.2 Business Information Entity

When a core component is used in a real business circumstance it serves as a basis of a Business Information Entity. Business Information Entity is the result of using a Core Component within a specific Business Context. There are three types of Business Information Entity, (1) Basic Business Information Entity; (2) Association Business Information Entity; and (3) Aggregate Business Information Entity.

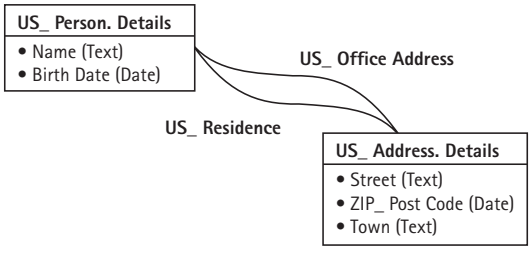

Figure 5: Business Information Entity

Figure 5 shows two Aggregate Business Information Entities, **US_ Person. Details** and **US_Address. Details**. **Name, Birth Date, Street, ZIP_ Post Code** and **Town** are Basic Business Information Entity and **US_ Official Address and US_ Residence** are Association Business Information Entities. In this example, the geopolitical business context has been applied to the core component to form the business information entity.

The diagram below shows the overall relationship between the core component and business information entity.

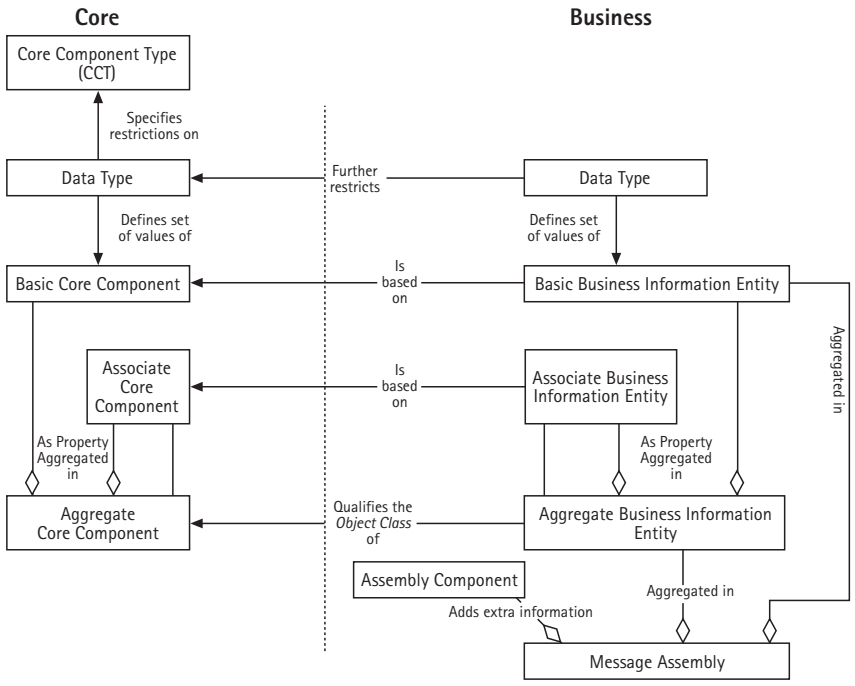

Figure 6: Relationship between Core Component and Business Information Entity

4.3 Naming Convention

CCTS Naming Convention is derived from the guidelines and principles described in document ISO 11179 Part 5 - Naming and Identification Principles For Data Elements. The data element definition should (1) state the essential meaning of the concept; (2) be precise and unambiguous; (3) be concise; (4) be able to stand alone; (5) be expressed without embedding rationale, functional usage, domain information; (6) avoid circular reasoning; and (7) use the same methodology and consistent logical structure for related definitions. The data element name, also known as the tripartite name should comprise the following:

a) Object Class Term - Represents the primary concept or Object Class of the data element, e.g. **Country**

b) Property Term - Represents the distinguishing characteristics or property of the Object Class of the data element, e.g. **Name**

c) Representation Term - Describes the form or Representation of the data element, e.g. **Text**

Using the examples in Figure 4, the data element names for the Basic Core Components are **Person. Name. Text, Person. Birth Date. Date, Address. Street. Text, Address. Post Code. Text, Address. Town. Text** and **Address. Country. Identifier.**

5 Universal Business Language

In 2001, when ebXML phase 1 was completed, work for ebXML phase 2 began. Both OASIS and UN/CEFACT discussed on scope of work and the area of responsibilities that each organisation would be responsible for. This is shown in Figure 7 below.

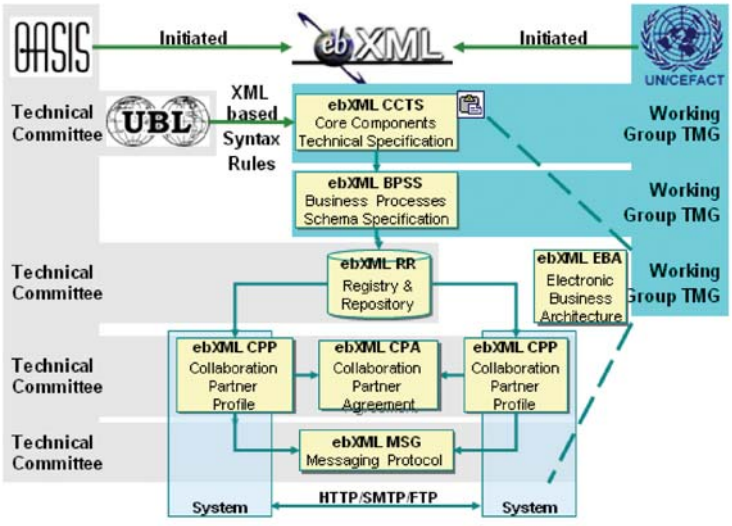

Figure 7: ebXML Phase 2

Business document schemas are not covered in both ebXML phase 1 and 2. While CCTS provides for standardisation, it does not address syntax specific expressions. Realising the 'gap', OASIS set up the Universal Business Language Technical Committee (UBL TC) in 2001 with a clear goal of developing a free cross-sector XML-based business language global standard which is non-proprietary and one that fully implements the CCTS. While UBL is built to support ebXML, it does not however mandate a particular framework. In short, UBL fulfils the promise of XML for business by defining a standard cross-industry vocabulary. It is the ebXML missing link. Together with ebXML, it enables the next generation of EDI which is (1) cheaper, easier and Internet-ready; (2) extends the benefits of EDI to small business; (3) fits existing legal and trade concepts; and (4) allows re-use of data.

5.1 UBL 1.0

UBL 1.0 was ratified as an OASIS Standard on 8 November 2004. It comprises eight XML document schemas covering the basic procurement process from order to invoice, as show in Figure 8 below.

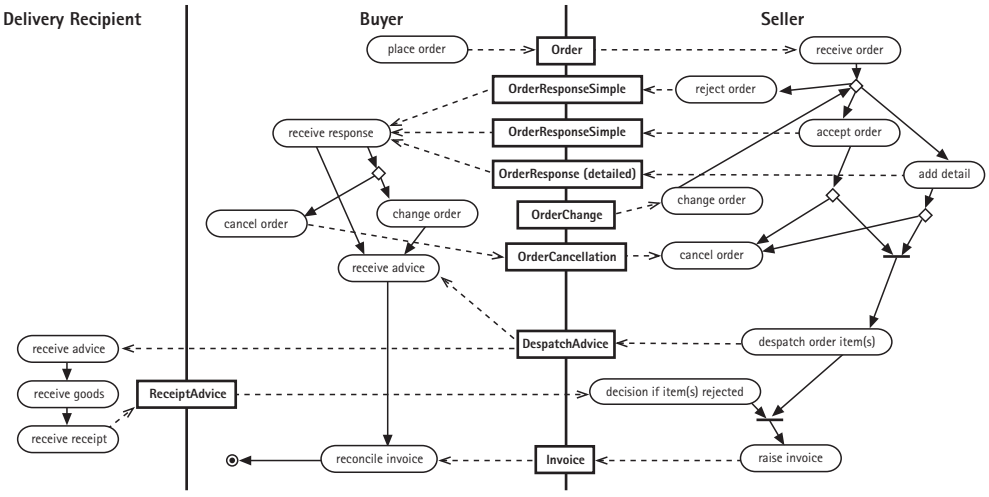

Figure 8: UBL 1.0 – Order to Invoice Business Process

Underlying the eight XML document schemas is a library of reusable Aggregate Business Information Entities (ABIEs) and Basic Business Information Entities (BBIEs). There are an estimated 104 ABIEs and 125 BBIEs in UBL 1.0.

5.2 UBL 2.0

In 2005, work on UBL 2.0 began. The basic procurement process has been extended to include catalog, quotation, credit note, debit note, statement and remittance advice. A new transport process containing packing list, forwarding instruction, bill of lading, waybill, certificate of origin, freight invoice and transport status documents are also added. In total, there are close to 30 XML document schemas in UBL 2.0. The underlying library has also grown significantly to 246 ABIEs and 414 BBIEs.

CrimsonLogic chairs the UBL Transport Subcommittee (TSC) and contributed the certificate of origin, packing list and freight invoice documents. Some of the ABIEs that the UBL TSC developed include the Consignment, Shipment, Shipment Stage, Goods Item, Port, Transport Equipment and Transport Means.

The first public review package of UBL 2.0 was made available for public review in January 2006. Some 160 comments were received during the public review. These comments have mostly been addressed. The second public review of UBL 2.0 is expected to be released in end July/early August 2006. Finally, UBL 2.0 is expected to be ratified as an OASIS Standard in the last quarter of 2006.

5.3 UBL Development Approach

The figure below shows the UBL development approach.

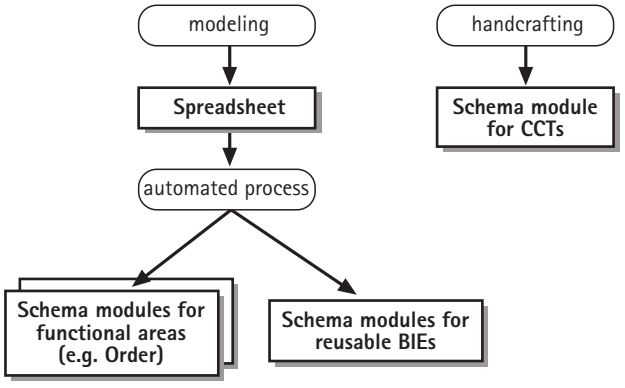

Figure 9: UBL Development Approach

Data models are first being done to identify the object classes. For each object class, the properties, together with the data type, occurrences are identified. The relationship between the object classes is also identified. This is shown in Figure 10.

After the completion of the data model, the object classes, together with the properties are then entered into an Excel file. The Excel files are then imported into an automated process to generate the XML schemas. The schema module for the CCTS is however handcrafted by UN/CEFACT and made available to UBL TC under license. Figure 11 shows a section of an Excel File for the Common Library.

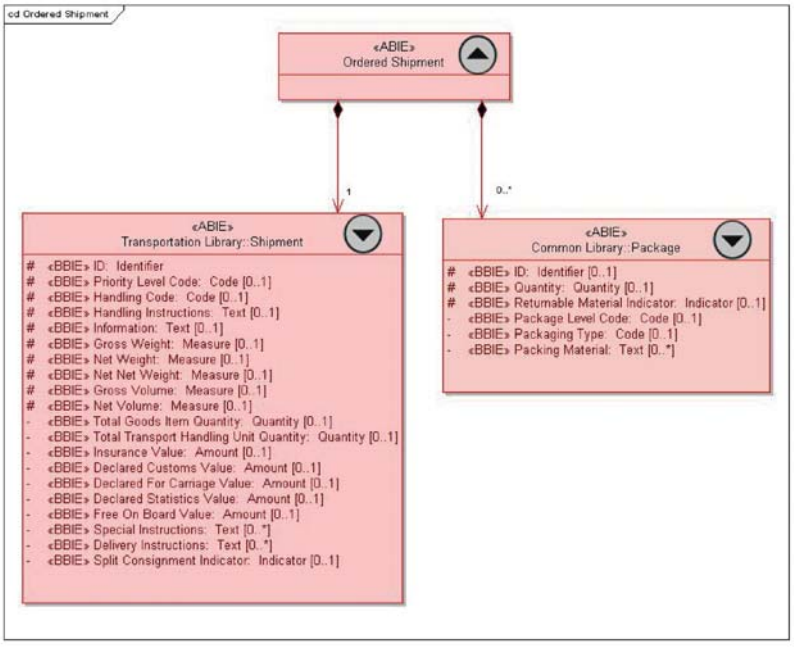

Figure 10: Ordered Shipment Data Model

A189 =SUBSTITUTE(SUBSTITUTE(CONCATENATE(IF(C189="","",CONCATENATE(C189,"")),"",D189)," ",""),"","")

	A	B	C	D	E	
1	UBL Name	Dictionary Entry Name	Object Class Qualifier	Object Class	Property Term Qualifier	Propert Possess
180	OrderReference	Order Line Reference. Order Reference		Order Line Reference		
181	OrderReference	Order Reference. Details		Order Reference		
182	ID	Order Reference. Identifier. Identifier		Order Reference		
183	SalesOrderID	Order Reference. Sales Order Identifier. Identifier		Order Reference		Sales O
184	CopyIndicator	Order Reference. Copy Indicator. Indicator		Order Reference		Copy
185	GUID	Order Reference. Globally Unique Identifier. Identifier		Order Reference		Globally
186	IssueDate	Order Reference. Issue Date. Date		Order Reference		Issue
187	IssueTime	Order Reference. Issue Time. Time		Order Reference		Issue
188	CustomerReference	Order Reference. Customer Reference. Text		Order Reference		Custom
189	OrderedShipment	Ordered Shipment. Details		Ordered Shipment		
190	Shipment	Ordered Shipment. Shipment		Ordered Shipment		
191	Package	Ordered Shipment. Package		Ordered Shipment		
192	Payment	Payment. Details		Payment		
193	ID	Payment. Identifier. Identifier		Payment		
194	PaidAmount	Payment. Paid Amount. Amount		Payment		Paid
195	ReceivedDate	Payment. Received Date. Date		Payment		Receive

Figure 11: Excel File for the Common Library

5.4 UBL Document Structure

Figure 12 shows the overall UBL 2.0 document structure.

Each document schema imports the common aggregate and common basic schema, which in turn imports the core component type schema.

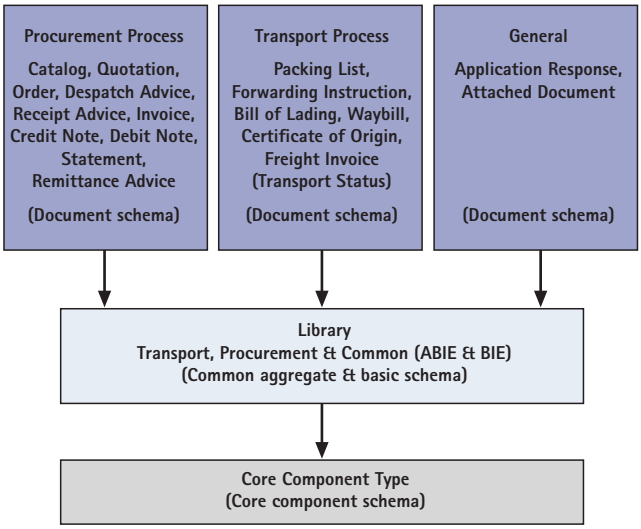

Figure 12: UBL 2.0 Document Structure

5.5 Beyond UBL 2.0

As mentioned earlier, UBL 2.0 is expected to be ratified as an OASIS Standard in the last quarter of 2006. When completed, UBL 2.0 is expected to be adopted worldwide, largely in Europe, North America and Asia.

Discussions have been going on between OASIS and UN/CEFACT to move UBL work to UN/CEFACT for the past two years. Both organisations have reached an agreement as follow:

1) UN/CEFACT recognizes UBL 2.0 as appropriate first-generation XML documents for eBusiness.

2) For OASIS and UN/CEFACT:
   (a) future UN/CEFACT deliverables constitute the upgrade path for UBL; and
   (b) the maintenance of UBL 2.0 remains with the OASIS UBL TC.

3) In the expectation that UN/CEFACT will produce its own integrated set of XML schemas within a period of three years, OASIS will produce no further major versions of UBL past UBL 2.0.

4) OASIS will grant UN/CEFACT a perpetual, irrevocable license to create derivative works based on UBL.

6 Conclusion

ebXML was developed out of the need to have a common XML-based infrastructure that enables global use of e-business information in an interoperable, secure and consistent manner by all parties. It encompasses all aspect of e-business; registering specification, process management, partner profile management, messaging specification and message component specification. The need arise due to issues faced with using EDI like (1) weak semantics; (2) lack of extensibility; (3) difficulty in aligning business process; (4) lack of registry; and (5) proprietary messaging.

Businesses conducting e-business transaction with one another need an internationally recognised message specification so that they don't have to deal with too many different message formats when exchanging messages with different parties. There needs to be a consistent way to define the message format, one that is flexible and allows a high degree of interoperability. CCTS provides the methodology to achieve this. However, CCTS does not address syntax specific expressions.

UBL was developed to address the business' need for an internationally recognised message specification, one that is cross-sector, non-proprietary and fully implements the CCTS. It is based on the '80/20 rule'; its goal is to identify and standardize the 20 percent of the possible data elements that will satisfy 80 percent of the usage scenarios. It is extensible, where sectors can customise (change) UBL following strict rules ensuring that basic data can still be read by the vendors that do not understand the sector specific extensions. As the world's first implementation of CCTS, UBL is the ebXML missing link.

After seven long years, the vision of ebXML has now been realised. Similar to the global EDIFACT standard which has been used for the past 25 years, e-business based strictly on XML infrastructure can now move full-steam.

7 References

[1] Eric Chiu, ebXML Simplified-A Guide to the New Standard for Global E-Commerce, John Wiley & Sons, 2002.

[2] David A. Chappell et al, Professional ebXML Foundations, Apress, 2004.

[3] David Webber, The Benefits of ebXML for e-Business, XML 2004 Proceedings.

[4] Mark Crawford, UBL: Realizing eBusiness XML, 2003.

[5] Jon Bosak, Denmark leads the way with electronic billing, August 2005.

[6] UN/CEFACT, Core Components Technical Specification - Part 8 of the ebXML Framework, November 2003.

[7] Universal Business Language (UBL) v1.0, OASIS Universal Business Language Technical Committee, November 2004 - http://docs.oasis-open.org/ubl/cd-UBL-1.0.zip.

[8] Universal Business Language (UBL) v2.0 CD1, OASIS Universal Business Language Technical Committee, January 2006 - http://docs.oasis-open.org/ubl/prd-UBL-2.0.zip.