Core Components: Delivering on the ebXML Promise

Alan Kotok

mightywords™

Core Components: Delivering on the ebXML Promise

by Alan Kotok

Published by MightyWords, Inc.
2550 Walsh Avenue
Santa Clara, CA 95051
www.mightywords.com

ISBN 0-7173-1939-3

Table of Contents

Executive summary **1**

1. Introduction **3**

E-business: It's bigger and sometimes better
Business semantics and its role in e-business
The rest of this briefing

2. A Look at ebXML **10**

First came EDI
Learning the lessons from EDI
ebXML infrastructure
Business content in the ebXML architecture
How ebXML works
ebXML development and status

3. The Process of Doing e-Business **19**

Business process models and analyses
Business process catalogue
Translating business process models to XML

4. Core Components: Definition and Discovery **28**

Core component structure
Core component naming conventions
Analysis and discovery of core components
Reusability, the purpose of core components

5. Context and Assembly in Business Messages **31**

Context gives meaning to core components
Context categories or drivers
Assembling business documents with core components and context

6. Current Work on Core Components **43**

7. Core Components Example **47**

Appendix 1. Names and identifiers of current core component **57**

Appendix 2. EbXML specifications and technical reports **64**

About the Author **66**

Executive summary

This briefing discusses Electronic Business XML or ebXML, a new global standard for conducting e-business, and focuses on the part of ebXML that deals with business semantics, called core components. It provides an overview of this innovative technology, to help business people begin planning for its adoption and benefits.

Once implemented in business software and services, ebXML will make it possible for any company of any size in any industry to do e-business with any other company in any other industry, anywhere in the world. To achieve this ambitious goal, ebXML sets out to do what no other business standard has tried to do, namely develop a way for companies to talk business systematically and accurately across industry or linguistic boundaries.

Core components are the basic data items that business documents can use and reuse from one document to the next. Many common business documents have the same basic structure and underlying data content. However, different industries use different terms for the same ideas, and for businesses to communicate with each other, they need a way of breaking through these semantic barriers, without asking companies to change their long-standing business practices.

With core components, companies can take e-business to an entirely new level, and encourage more innovative business collaborations and improvements in business processes. The ability to relate data from one business document to another makes it easier for companies to share their data in electronic messages with more business partners. This increased data sharing can expand the use of collaborative business planning and supply chain integration. Also, core components can help integrate current company data with systems and vocabularies used in related fields and thus make it easier to expand into new lines of business.

With core components, electronic business messages can identify these underlying common data items, and relate them not only to their counterparts in other business documents, but to earlier e-business technologies (such as Electronic Data Interchange), and the data stored in company business systems. Core components have unique identifiers and a neutral syntax that makes them interchangeable, machine readable, and independent of any vendor's software or network.

The work on core components includes not only the identification of these interchangeable parts in business documents, but the systematic and consistent definition of the business context – the substance of industry business practices and terminology – that give the components their precise meaning in business documents. The systematic combination of core components with context allows for the automatic assembly of e-business documents exchanged with trading partners.

EbXML is still a work in progess. The development phase ended in May 2001,ethe core components team having completed its basic methodology and proposed a starter set of core components.. Since then, the two leading e-business standards organizations have joined to continue the work. Industry organizations and individual companies should consider taking part in this important exercise, since it will likely determine the language of e-business for many years to come.

1. Introduction

This briefing is about doing business electronically, combining the domains of business and technology in equal parts. The briefing discusses a new global standard for e-business called Electronic Business XML (ebXML)and one of its key features, which facilitates doing business across traditional industry boundaries. This feature, called core components, makes it easier to generate messages and interpret incoming electronic data used in business, and thus encourages millions of enterprises worldwide to take part in the e-business boom.

E-business: It’s bigger and sometimes better

To most people, e-business means selling goods and services to consumers over the World Wide Web, and for good reason: The online retail market is big and getting bigger. The Boston Consulting Group expects the online retail market in the U.S. to grow some 45 percent in the year 2001, to $65 billion.1 Another research firm, Jupiter Media Metrix, expects the retail market to grow to $130 billion by 2006, with 63 percent of the online population making purchases by that date, up from about half (52 percent) in 2002.2

But the big returns – and perhaps the secret of success – in e-business come not from retailing but from business-to-business commerce. Giga Information Group expects the value of electronic business-to-business commerce to reach $5.2 trillion by the year 2004, up from $3.3 trillion in the year 2000. Giga attributes this growth to the availability of different channels for e-business by 2004, most of which result from the application of Internet-based technologies. [3]

[1] “Online Retail Market in North America to Reach $65 Billion in 2001,” Boston Consulting Group, 2 May 2001, http://www.bcg.com/media_center/media_press_release_subpage44.asp .

[2] “Reports Of The Death Of Online Retail Are Greatly Exaggerated, Says Jupiter Media Metrix,” Jupiter Media Metrix, 22 May 2001, http://www.jup.com/company/pressrelease.jsp?doc=pr010522 .

[3]“ Multi-channel B2B E-commerce to Exceed $5.2 Trillion in 2004,” Giga Information Group, 9 January 2001, http://www.gigaweb.com/Marketing/home.asp?intGContextID=15&strMode=newsroom1 .

What makes e-business with other companies [4] so lucrative? Commercial interactions among businesses go well beyond trade in goods and services. The big payoffs come from improved business processes that create more opportunities and reduce overhead. These collaboration opportunities contribute to the bottom line not only today, but keep paying off well into the future. In fact, Jupiter Media Metrix predicts these collaborative applications will grow faster than the routine commercial uses of private trading networks during 2001 and 2002.[5]

To show how collaborative commerce differs from routine trade, consider the following example. I love coffee, and not just any coffee, but those high-test coffees you get in coffee bars. At a coffee bar I frequent, the manager often needs to take inventory of the perishable sandwiches, baked goods, cold drinks, teas, and, of course, the many kinds of coffee. With paper and pencil, he compiles that inventory, which tells him the quantities of various goods in stock, from which he can order more if needed.

If the store's suppliers have online ordering, the manager could take the paper inventory tabulations, log on to the suppliers' web sites and order the goods needed. This is e-business to most people, including most business people.

Now consider this hypothetical scenario: The manager notices that most of his suppliers print standard bar codes on the items delivered to his store. The standard bar code used in the North America for grocery items is called the Universal Product Code or UPC; outside North America it is called EAN (for European Article Number, although it is used in Asia, Africa, and South America).[6] You see these bar codes on most products sold on supermarket shelves, since nearly all large chains now require their suppliers to put bar codes on stock items, but they work just as well in retail stores of any size. With a minimal investment -- a simple bar code reader can attach to a standard PC's serial or USB port and cost under $US 500 -- the coffee bar can keep an accurate running count of the items in stock, and save the manager a ton of time now spent with paper and pencil.

But the scenario gets even better. With the store's inventory online, the coffee bar's system can generate purchase orders automatically once the inventory levels reach the reorder point. Why should the manager type out the information already sitting in the store's systems? The manager already has the suppliers' prices, the system has the quantities needed, and the location for

[4] We use the term 'company' here to refer to any enterprise involved in the production or marketing of goods and services, including those in the public sector and not-for-profit organizations.

[5] "B-To-B Collaboration Will Be The Fastest Growing Use Of Private Trading Networks As Companies Shift Their Focus Beyond Transactions, Predicts Jupiter Media Metrix," Jupiter Media Metrix, 23 May 2001, http://www.jup.com/company/pressrelease.jsp?doc=pr010523

[6] "Getting Started ID Numbers and Bar Codes," Uniform Code council (undated), http://www.uc-council.org/id_numbers/id_getting_started_with_id_num.html

delivery does not change. The store and the suppliers should use the power of computer systems to generate that information automatically in electronic business documents, rather than manually re-entering the data into computers, with the problems of time and errors that causes. Once the suppliers get this information, they can fill the orders, ship the goods, and invoice the store for the costs..

The history of e-business, starting with electronic data interchange (EDI), shows that something interesting happens when companies begin exchanging data in this way. Companies begin to look at the information they generate and receive in a whole new way. The exercise often helps point out redundancies and bottlenecks, and inspires companies to work more closely together to re-engineer business processes that benefit all parties.[7]

For example, in the scenario described above, we discuss how the coffee bar can automatically generate purchase orders and invoices. But why even bother with purchase orders and invoices? The coffee bar can send the suppliers automated daily inventory reports of the goods provided by the suppliers, and the suppliers can then feed the data directly into their production and delivery systems, as long as they can agree on prices in advance. The suppliers can ship daily the goods needed to keep the store stocked but without paying for too much inventory that can go stale and tie up capital. As a result, the purchase order becomes redundant.

- And what about invoices? Remember that the coffee bar is scanning the bar codes on items as they enter into inventory, so it has a running count of new items delivered and accepted. Remember also that the coffee bar and the suppliers have already agreed on pricing. If the coffee bar accepts the items into inventory, and the supplier gets inventory records every day, both parties know the precise items and quantities accepted. And since they both have agreed on prices, why bother with a separate document called an invoice? Upon receipt of the items, the manager of the coffee bar can authorize payment to the suppliers, and skip the invoice, and its processing at both ends, completely. (In some places in the world business regulations require an invoice, but not in this case.)

The benefits to the coffee bar from these new arrangements begin with relieving the owner of the timeconsuming drudgery of administrative tasks, which lets him concentrate on the real reason for going into business: providing better quality services for the customer (like getting an extra shot into my Café Americano). But the benefits show up directly on the bottom line as well. These innovations can reduce inventories to a bare minimum, reduce or eliminate spoilage, reduce finance costs for the inventories, and improve cash flow for both the coffee bar and the suppliers. Figure 1 summarizes the differences between these routine electronic trade and collaboration scenarios.

[7] EC/EDI Insider from Washington Publishing Company has compiled a number of these case studies. See http://www.wpc-edi.com/Insider/Articles/Categories/CaseStudies.html

Figure 1.

Trade or Collaboration?

Trade scenario:	Collaboration scenario:
Supplier posts catalog on web site	Customer forecasts quantities of items needed over period of time
Customer logs into supplier site	Supplier and customer negotiate prices, quantities, and terms
Customer selects goods from catalog	Customer system sends periodic inventory reports to supplier system
Customer provides credit card or arranges terms	Supplier ships goods to maintain agreed-upon quantities
Supplier ships goods	Customer scans bar codes to enter replenishment items into inventory
Customer keys data into inventory system	Customer authorizes electronic payment to supplier for replenished quantities
Supplier processes credit card or sends invoice	

The collaboration scenarios may seem far-fetched at first, but these kind of collaborative trading arrangements have become common at many companies involved in manufacturing and distribution. The practice of vendor-managed inventories, for example, has been going on in various forms for decades. In vendor-managed inventories, like those described for my favorite coffee bar, the suppliers take responsibility for keeping business customers stocked, within limits agreed upon in advance. [8]

What is different and perhaps striking about this hypothetical case are the parties themselves. Both the coffee bar and the suppliers are small companies. Up to now, only big companies could consider these kinds of business process improvements. The benefits to these companies may be significant, but can they afford the investment needed in these systems? Up to now, the answer was probably no. But ebXML is designed to extend the benefits of collaborative e-business to companies of this size.

Business semantics and its role in e-business

In this scenario, we talk about the interactions between the coffee bar and its suppliers, but there are other interactions going on as well. If a supplier does not have its own trucks, a delivery service needs to deliver the goods. In that case, the delivery service will need certain pieces of information from the supplier (about the goods being delivered, need for refrigeration, shipping location, delivery location, and desired delivery times), as well as information from the coffee bar (name and telephone number of contact, and hours open).

Providing this information seems simple enough, but in an automated environment simple can be a relative term. For example, the coffee bar's inventory system may call its items in inventory 'items,' while the supplier's system may call them 'units,' and the delivery company system calls them 'packages.' Even the names that the different parties call each other will differ. The supplier system may call itself 'vendor', while the coffee bar system may refer to the supplier as 'seller.' And the delivery company uses the terms 'shipper' and 'consignee.'

In human communications, we can adjust to such inconsistencies in language easily, but with machines, we must define our terms carefully. To achieve the kind of results from e-business collaborations discussed above – generate new income-producing opportunities, reduce inventories to a bare minimum, reduce or eliminate spoilage, reduce finance costs for the inventories, and improve cash flow – the systems must be able to talk to each other. In a nutshell, this is the problem of business semantics. Up to now, e-business has not addressed this issue in a

[8] Alan Saipe and Jeremy Geiger, "Global Brief on Vendor Managed Inventory," KPMG Consulting , Toronto, Canada, August 1996, http://www.vendormanagedinventory.com/article3.htm

systematic way. EbXML has proposed methods of relating different industry terms to each other, which this briefing discusses in some detail.

Up to now, individual industries organized their own e-business initiatives that addressed issues of semantics within the industries. The paper industry, for example, calls wholesalers (who play a vital role) 'merchants.' Most printing companies are small, local operations – in the U.S. most printers employ fewer than 20 employees[9] – and will buy all or most of the paper they use for print jobs from merchants. Likewise, paper mills will sell most of the printing grade paper they produce to merchants.

But these printers use other materials in their work – ink, plates, and chemical solvents – acquired through different supply chains than those used for paper. For these other products, printers normally work through companies called 'distributors.' The processes and data exchanged for paper with merchants are similar to those exchanged with distributors for other materials. However, the systems used by printers need to relate these processes and terms if they hope to be effective. As a result, for printing companies to enjoy the full benefits of e-business, just within their own industry, they need to solve this problem of business semantics.

Yet businesses today rarely have the luxury of working just within their own industry. Companies often need to expand into new markets and take advantage of opportunities as they arise. Businesses need systems with the flexibility to quickly and easily relate the data in their used in one industry to the rules and terms used in other industries. This is the challenge and opportunity of interoperable business semantics that ebXML addresses.

Even if you do business in a stable industry and have no plans to expand into new lines, you still have regular dealings with companies outside the industry boundaries. In the coffee bar example, consider the role of delivery companies, which often have their own vocabularies. Businesses of all sizes have regular interactions with banks, accountants, utilities, and government agencies. Why not make these interactions with these enterprises electronic, seamless, and integrated with a company's business systems?

The rest of this briefing

In this briefing, we next discuss the ebXML initiative, explain why it attracted a groundswell of support worldwide, and describe both its business objectives and technical architecture. We then delve deeper into the business content aspects of ebXML, namely the business process models and core components. With core components, we discuss the methods for identifying and naming core components, their generation and assembly into

[9] "Industry Trends," Graphic Arts Information Network," PIA/GATF, undated, http://www.gain.org/servlet/gateway/industry/index.html .

electronic business documents, and the ongoing work to define core components across business domains. Finally, we present a real life example of common business messages, show how core components fit into these messages, and suggest ways that the companies involved can expand into new lines of business as a result of core components.

One word of warning: This briefing discusses ebXML and specifically its work on core components, but it is NOT a technical specification that developers can follow for building business systems. As noted throughout the text, at the time of writing this briefing, the work on core components continues, but is not yet completed as an ebXML specification. We encourage readers to visit the ebXML web site (www.ebxml.org) or, better yet, take part in the work of the working groups either conducting the basic work or defining core components themselves. In chapter 6 of the briefing, we provide web addresses to find more details about these activities.

2. A Look at ebXML

At its core, the ebXML initiative is an attempt to make e-business easy, ubiquitous, and inexpensive for the millions of smaller companies not yet part of the e-business experience. This project, started in November 1999, aims to make it possible for any company in any industry to conduct business electronically with any other company in any other industry anywhere in the world.10

First came EDI

The opportunities for companies to profit from this e-business experience, especially collaborative e-business, raised the expectations of many managers. In the later part of the 1990s, executives wishing to move into the world of collaborative e-business soon discovered that the technology was not available to make it happen for most companies. The only technology available for the exchange of standard business messages between companies was electronic data interchange (EDI).

EDI technology, based on standards written mainly in the 1980s and updated annually, worked well (and continues to work well) for a good number of companies. Many of the productivity improvements generated during the later part of the 1990s that propelled the North American economy to unprecedented levels came about as a result of companies sharing more data, and using that data to make fundamental changes in the way they do business. Manufacturers, distributors, retailers, transportation companies, financial institutions, and government agencies (among others) use EDI to better coordinate their activities, reduce inventories, and improve service to customers.

As defined by the Accredited Standards Committee X12 (ASC X12), the standards body for e-business messages in North America,

> EDI is the computer-to-computer exchange of business data in standard formats. In EDI, information is organized according to a specified format set by both parties, allowing a "hands off" computer transaction that requires no human intervention or rekeying on either end. The information contained in an EDI transaction set is, for the most part, the same as on a conventionally printed document.[11]

[10] "Enabling Electronic Business with ebXML," ebXML White Paper, December 2000, http://www.ebxml.org/white_papers/whitepaper.htm.

[11]ASC X12, "What is EDI." http://www.x12.org/x12org/about/index.html?whatis.html

The basic message formats in the ASC X12 standard are called transactions sets and messages in the UN/EDIFACT standard used outside North America. As the definition suggests, most of the EDI messages are designed as generic electronic business documents, such as invoices, ship notices, and purchase orders. Some of the newer messages serve specific functions in specific industries, such as health care claims.

EDI is a powerful technology, and provides benefits for companies with high volumes of transactions supplied by many trading partners, which generally means larger enterprises. But for smaller companies (except for those acting as suppliers for big customers), EDI is just too complex and too expensive to make economic sense.

EDI software often comes with a high price tag. Because of the size and complexity of EDI standards, the software has to support each of the 300-plusseparate messages in the latest version (version 4040, December 2000) of the X12 standard, plus thousands of data segments and elements, and tens or hundreds of thousands of code entries. Also, the standards change annually, which means packaged EDI software often needs to support multiple versions of the standards going back to the 1980s.

To compound the problem, each industry implements EDI differently. If a company implements EDI in, for example, the aerospace industry, it often finds it cannot transfer much of that investment to conduct EDI in the petroleum business. The EDI standards may offer individual messages for specific business documents, but they provide only a general template and much more functionality than most individual companies and industries need. The different industries pick out the messages that companies use in that line of business, then tailor the transaction sets by selecting the data elements and segments (collections of related elements) needed. While this approach is efficient for each industry, it isolates each industry flavor of that message in a silo, making it difficult to relate the experience of one industry to another.

EDI has another problem, namely the lack of integration with companies' basic management systems, such as manufacturing, inventory, and accounting. EDI messages are designed to go door-to-door, that is, from one mailbox to another, using an efficient but cryptic syntax. Receiving an EDI message does a company little good, until the company maps each of the elements in that message to its databases. For example, in the case of high volume transactions in a structured supply chain, the hundreds of suppliers serving automobile manufacturers, the mapping exercise is worth the effort. Where the interactions are less frequent and do not easily match up with company databases, EDI becomes much less attractive.

These and related issues made EDI too expensive and difficult for most small companies and thus had the effect of shutting out most companies from the benefits of EDI. The U.S. Census Bureau, in its most recent economic survey (1997), found that the number of companies with more than 20 employees represents just over 10 percent of the 5.5 million businesses in the

United States.[12] In just the sheer numbers of small companies, therefore, the opportunities for expanding e-business are significant.

The EDI standards bodies, particularly the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) that manages the UN/EDIFACT standard, recognized the problem and have looked into making basic changes in the way they approach e-business to expand its scope to smaller companies. This organization developed a business process modeling methodology that lets industries and companies represent the interactions among businesses in business terms, yet still systematically translate those processes into precise messages and data elements sent from one company to another. We discuss this methodology later on in the briefing.

Learning the lessons from EDI

Despite the drawbacks of EDI, two decades of experience in defining business semantics and successfully exchanging data among companies provide some valuable lessons for ebXML and other e-business initiatives. In EDI, all of the messages, the parties sending and receiving the messages, and many of the data items in the messages (such as shipments, locations, and products) have unique identifiers. Some of these identifiers, like the UPC/EAN bar codes found on retail goods, are so common that they have become every day features of modern Western civilization. Other identifiers, such as Standard Address Numbers used in the book publishing industry, are confined to particular industries. But the use of electronic rather than hard copy documents creates an urgent need for unique identification of these objects. Without them, systems used in e-business cannot tell one message, party, product, or location from another.

Another contribution of EDI is the use of acknowledgments for messages. Getting acknowledgments for important messages is just good business practice. When sending a hard copy document through the mail, you can buy a certified mail receipt to verify that the receiving party actually received the document. Package delivery services also can provide signed receipts for deliveries. Fax machines send a standard acknowledgment message when the receiving machine accepts the transmission or an error message if encountering problems. Likewise, e-business messages need to provide acknowledgments of receipt to indicate that the receiving party has received the message. The X12 standard has a set of common acknowledgment messages, with the functional acknowledgment (transaction set number 997 in the X12 array) being the most-often used.

[12] U.S. Census Bureau, “Statistics about Small Business and Large Business from the U.S. Census Bureau.” http://www.census.gov/epcd/www/smallbus.html .

However, an EDI acknowledgment indicates only that the mailbox on the other end received the message, in much the same way that the fax machine acknowledgment says that the machine received the fax. It does not imply understanding of the message contents by the receiver nor does it indicate a commitment to act on the contents. A number of EDI messages are designed to serve as responses to request messages that tell explicitly how the receiving organization acted or intends to act. The meaning of these responses goes well beyond a simple acknowledgment.

For example, the ship notice transaction set in the X12 standard is sent by a vendor to a customer or intermediary (third-party warehouse) telling that the goods previously ordered are en route, when they are to be delivered, and sometimes describing the means of transportation. Upon delivery, the receiver of the goods can send a receiving advice message as a response that indicates which goods were received and accepted into inventory, and which goods were not received and the reasons for their rejection. If the supplier is managing the inventory of the customer, the receiving advice tells the supplier the units added to inventory. It also alerts the supplier for which goods the customer will authorize payment.[13]

The designers of ebXML used as one of its guiding principles to learn from these lessons and, to the extent possible, make the transition or migration from EDI to ebXML as seamless as possible. The ebXML requirements document included many of these lessons, as well as incorporating the work done on earlier initiatives to simplify EDI for smaller businesses and business process analysis developed by UN/CEFACT. The ebXML requirements also called for making as much use as possible of existing standards and specifications developed by the World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF), rather than building entirely new solutions. [14]

EbXML infrastructure

The ebXML technology uses a modular structure that divides the architecture into a set of building blocks. Because of this modular structure, industries or companies can implement as much of the technology as they need, and build on the first efforts later.

[13] Alan Kotok, “XML and EDI, Lessons Learned and Baggage to Leave Behind,” XML.Com, 15 August 1999, http://www.xml.com/pub/1999/08/edi/index.html

[14] “General ebXML principles,” electronic business XML (ebXML) Requirements Specification Version 1.06, May 2001, pp. 10-11.

Trading partner profiles and agreements

Important and unique features of ebXML are the ability to find new trading partners and to agree on the technical aspects of the overall trading partner relationship. For the discovery of new trading partners and their capabilities to conduct e-business, ebXML has collaboration protocol profiles (CPPs). With CPPs, companies can indicate in a standard XML document the lines of business, processes, individual messages, exchange technologies, and security features they support. Companies can then merge their CPPs and agree with each other on the business processes, messages, and technologies used for e-business in XML documents called collaboration protocol agreements (CPAs). Please note that CPAs apply only to the technical aspects of the relationship. The parties still may need documentation to cover other legal and business issues.

Registries and repositories

Most companies' first encounters with ebXML will probably come through the registries where companies can locate the industry processes, messages, and semantics used to define the interactions with other parties. Industry groups will list in the registries their processes, messages, core components, and other business objects, such as industry-defined code lists, that companies need to do e-business. Companies will file their CPPs in the ebXML registries and search the registries for CPPs of other companies with which they can do business. Because of these functions, registries and corresponding repositories will play a critical role in getting companies started with ebXML (see Figure 2 below).

Messaging

For companies interested in ebXML, the messaging functions will probably be the first specifications they implement. These functions enable companies to send and receive data in specified envelopes and message format. The ebXML messages are based on the Simple Object Access Protocol (SOAP), an XML application for messaging that defines a basic format with headers to indicate sending and receiving parties, as well as routing and security. EbXML uses an enhancement to SOAP that allows for attachments of digitized content other than text. This enhancement lets ebXML messages include objects such as software code or computer-generated images. The specifications also recommend acknowledgments and define the requirements for reliable messaging to ensure that companies receive a message no more than once.

Business content in the ebXML architecture

We address business process and core components more fully in later sections of this briefing. Here, we discuss the role of business process and core components in the overall ebXML architecture.

Business processes

The emphasis in ebXML on defining and analyzing business processes is a basic feature of ebXML, and one that separates it from most other e-business standards or specifications. This emphasis on business process, however, did not begin with ebXML. RosettaNet, an XML vocabulary for the computer industry (component and equipment manufacturers, and software companies) devoted most of its early work to developing Partner Interface Processes (PIPs), that define and describe the processes in that industry. 15 But ebXML is the first large-scale e-business initiative to apply business modeling across industries.

Business processes define the relationships, called collaborations, between business partners. They identify the parties in the transactions, the messages exchanged between the parties, the sequence of the messages, and the data elements in the messages. Industry groups list in ebXML registries the processes they use to conduct e-business. Registries also have companies' profiles (CPPs) that list the processes they support. Trading partners combine the CPPs into technical agreements (CPAs) that indicate the processes supported by both companies.

Core components

In ebXML, business processes offer interoperability at the message level, while core components provide interoperability at the more detailed level of individual data elements. Core components indicate the data elements used most widely by businesses and in multiple industries, and enable companies in one industry to relate these common data items to their counterparts in other industries.As defined by ebXML, core components have a neutral syntax and are assigned a unique identifier. The neutral syntax lets companies relate data in ebXML messages to their counterparts in industry XML vocabularies, as well as data elements in EDI messages. The unique identifier lets systems access specific core components to assemble messages during the design phase.

Industries – usually standing industry organizations or consortiums of companies interested in industry-wide solutions – will identify the core components in their business processes and indicate the precise location of the components when they are assembled into messages. These

[15] See www.rosettanet.org for details of this initiative, RosettaNet is an ebXML participant.

components are part of the processes listed in industry registries and identified in company CPPs. The agreements between two companies doing e-business, known as CPAs, include core components as part of the configuration defining the precise relationship between the two companies.

How ebXML works

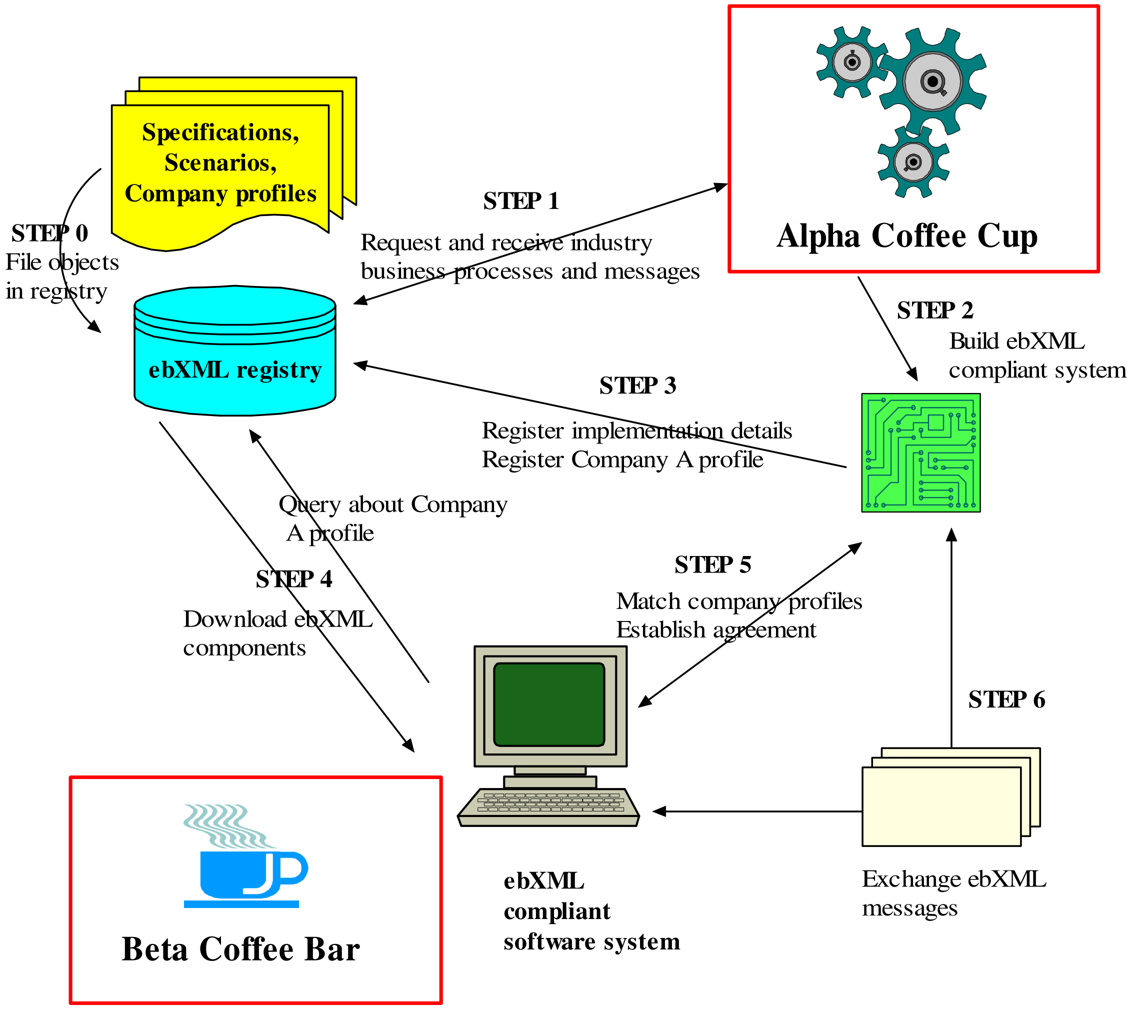
Figure 2 offers a typical scenario showing how industries and companies will likely interact with ebXML. Before anything can happen, shown here as step 0, industries file in an ebXML registry their business processes, message schemas, core components, and special industry objects (such as standard product codes or hazardous materials lists) that companies need for e-business in that industry. Our scenario shows two companies from our earlier example, Alpha Coffee Cup Co. and Beta Coffee Bar Inc., although the scenario could just as easily involve a company and a third-party message hub or Web-based application service provider.

Alpha Coffee Cup, a manufacturer of paper board coffee cups and sleeves, plans to expand its operations to take advantage of the boom in coffee bars serving coffee fanatics like myself. The company goes to the grocery industry’s registry and finds the processes, schemas, core components and other objects it needs to conduct e-business with others in the industry (step 1). After inspecting these materials, Alpha Coffee Cup finds it needs to upgrade its systems to do handle these new requirements, and through a combination of new software and home-grown upgrades, makes the enhancements (Step 2).

For potential trading partners to know that Alpha Coffee Cup ready to do ebXML e-business with other companies, it prepares and files a collaboration protocol profile (CPP) with the same registry it found the earlier materials (step 3). The CPP spells out the processes and messages Alpha Coffee Cup supports, as well as its technical capabilities for e-business. The good news here is that Alpha Coffee Cup only has to perform steps 1 through 3 just once to get started, although if it makes any further enhancements to its systems it may need to revise its CPP accordingly.

Because Beta Coffee Bar has already started doing e-business with ebXML, it would like to use as many ebXML-compliant suppliers as possible. After Alpha Coffee Cup files its CPP, Beta Coffee Bar does a search of the registry, and sees that Alpha Coffee Cup is registered with this industry. Beta Coffee Bar already is listed in the industry registry, and verifies Alpha’s capabilities in Alpha’s CPP. Because the purchasing and shipping of coffee cups use somewhat different business processes than other consumables, Beta Coffee Bar needs to retrieve a few new objects from the industry registry to do e-business with Alpha Coffee Cup (Step 4).

Figure 2. Typical ebXML business scenario


Specifications, Scenarios, Company profiles
STEP 0
File objects in registry
ebXML registry
STEP 1
Request and receive industry business processes and messages
Alpha Coffee Cup
STEP 2
Build ebXML compliant system
STEP 3
Register implementation details
Register Company A profile
Query about Company A profile
STEP 4
Download ebXML components
STEP 5
Match company profiles
Establish agreement
STEP 6
Beta Coffee Bar
ebXML compliant software system
Exchange ebXML messages

The two companies merge their respective CPPs into a collaboration protocol agreement or CPA that details the technical aspects of the business relationship (step 5). Once the CPA is in place, the two companies can begin exchanging ebXML messages, using transactions specified in the business process, shown as Step 6 in the Figure 2.

Figure 2 shows the important role played by registries in the ebXML model. Companies will have at least several interactions with registries as they get started with ebXML. When actually exchanging messages (Step 6 in Figure 2), however, the parties deal with each other, not the registries. [16]

EbXML development and status

EbXML is a joint initiative of the United Nations (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS), with participation from over 2,000 participants in 30 countries. Participants include standards bodies, industry groups, business consortiums, and individual companies, many of them IT vendors. The development phase ran from November 1999 to May 2001. During that time, several technical project teams wrote the specifications and technical reports that developed ebXML's technical infrastructure and business content.

With ebXML, the process of developing the specifications was almost as remarkable as the final documents themselves. In the 18-month development phase any person could sign up for the project team list-serves, with no membership fees or restrictions. The only time ebXML collected any fees was during the quarterly in-person meetings, to help defray costs of those meetings. The entire operation relied on volunteers from the participating companies.

After May 2001, the work of ebXML divided up between OASIS and UN/CEFACT, with OASIS taking over the technical infrastructure teams – messaging, registry/repository, collaboration partners, and conformance – and UN/CEFACT taking over the business content teams: business process and core components. A joint coordinating committee looks after overall management and marketing.[17]

[16] David Webber and Anthony Dutton, "Understanding ebXML, UDDI and XML/edi," XML Global, October 2000, http://xml.org/feature_articles/2000_1107_miller.shtml.

[17] Alan Kotok, "ebXML: It Ain't Over 'til it's Over," XML.Com, 16 May 2001, http://www.xml.com/pub/a/2001/05/16/ebxml.html

3. The Process Of Doing e-Business

EbXML begins with the business process. As discussed earlier, business processes define the relationship between business partners. They identify the parties in the transactions; the messages exchanged between the parties, the sequence of the messages, and the data elements in the messages.

EbXML chose to put a major emphasis on business process for three reasons:

1. It offers a way of recording business needs and interactions separately from technology, so business people can talk business in their own terms without getting caught up in the technical implementation.

2. The approach provides a reference for finding commonalities among industry e-business implementations and between different technologies, to encourage interoperability.

3. It allows for the use of modeling tools to systematically describe business processes in succeeding lower levels of detail that can then be translated directly into technical implementations.

EbXML made business processes the starting point in defining electronic business messages and the data items they contained. In the ebXML specifications, companies identify the business processes they support in their collaboration protocol profiles, and later in the collaboration protocol agreements undertaken by trading partners. EbXML registries list an industry's business process models, as well as individual message schemas, and core components identified in those processes.

Business process models and analyses

The business process parts of ebXML are based on previous work done by UN/CEFACT that developed a unified modeling methodology (UMM) for business processes. The UMM defines a meta-model, literally a model of models, that provides high-level views of interactions among parties and systems engaged in e-business. UN/CEFACT, in turn, based its UMM on earlier research and development work called the Open/EDI Reference Model, which became ISO standard 14662. A reference model acts as a guideline for further standards work.[18]

One of the views that the UMM and ISO14662 presents is a business operational view or BOV that describes the interactions among parties in strictly economic terms. The BOV covers the terminology used in an industry, known as business semantics, as well as its common practices. For example, some industries (e.g., raw materials) may not have standard package configurations for their shipments, and thus do not have a standard industry-wide definition of net weight. Business interactions in these industries involving shipments therefore will include a tare or package weight that the parties need in order to determine the net weight of the goods shipped. As a result, business messages in these industries for purchasing, shipping or inventory control, will include a tare or package weight measurement, while comparable messages in those industries having standards for determining net weight will have little need for it.

Another view represented in the UMM is the functional service view (FSV), which shows the technology end of the interactions among companies, such as networks, software, and protocols used to exchange data. The main purpose of business process modeling is to let business people concentrate on defining the BOV, and let the BOV drive development of the FSV (or, in some cases, multiple FSVs). In either case, one needs a systematic and consistent way to connect the BOV to the FSV.

The way recommended by the UMM to connect business models to technology is the Unified Modeling Language (UML), a graphic representation of business interactions used for several years for software design. UML provides tools for analysts to describe processes, interactions, and properties of objects, down to fine levels of granularity. With these succeeding levels of detail, it is possible to directly relate the behavior and properties of objects to pieces of code, both software code and XML, but without resorting to the syntax of those technologies. As a result, UML holds a great deal of promise for business process analysis and systems design, where interoperability is a key requirement.[19]

For analyzing business processes themselves, to better understand the way businesses work and to document the interactions, ebXML offers a technical report (not a full-fledged

[18] Information technology -- Open-edi reference model, ISO/IEC 14662:1997. International Organization of Standards, 1997.

[19] A complete or detailed discussion of UML is beyond the scope of this paper; readers should visit the Web site of the Object Management Group (www.omg.org/) for more information on the subject.

specification, but still considered helpful for implementers) with worksheets to analyze business processes. The steps and worksheets spelled out for the analysis have corresponding steps in the UMM. These steps include:

1. *Business reference model* – provides a frame of reference for the overall process and can refer to other authoritative process models.

2. *Business process identification and discovery* – offers a high level inventory of processes, with little beyond a simple listing.

3. *Business process elaboration* – adds a little more detail to the processes, including the actors and pre- and post-process conditions.

4. *Business transaction definition* – defines the actual activities and the parties within the organizations engaged in the process.

5. *Business information definition* – describes the information contents of the messages exchanged, including the data items, field sizes, data types, and requirement references. This part can also include a context description needed for assembling the message using core components.[20]

At this finest level of detail, the process models connect the interactions among companies with the contents of the messages, or the business semantics. The model breaks down electronic business messages into business information objects, the chunks of information conveyed from one party to another within the document. Business information objects can include items unique to that industry, called domain components, or items contained in common processes found across industries, called core components. Or they can include other business information objects. Figure 3 shows the relationship of these components in business messages and models.

At this point as well, the model makes a distinction between the data items common to the business processes that cut across industries, and the industry context that gives the data the richer and specific meaning to that line of business. The context of the components provides the glue for connecting the components in electronic business messages.

[20] *"Worksheet Based Analysis Overview," Business Process Analysis Worksheets & Guidelines v1.0, ebXML Business Process Team , 10 May 2001, pp. 12-13.*

To provide examples and illustrate the differences between core component, domain component, and context, we go to the paper industry. The standard cut size for office paper in North America is 8 ½ by 11 inches. Outside North America, the industry measures paper according to sizes dictated by ISO standard 216, which stipulates that the width to height ratio is 1:1.4142, with the normal cut size for office paper, known as A4, being 210 by 297 mm.[21] The paper industry in North America and elsewhere uses many of the same business processes, but the industry context requires applying different sizes depending on the location of the customer: North America or elsewhere.

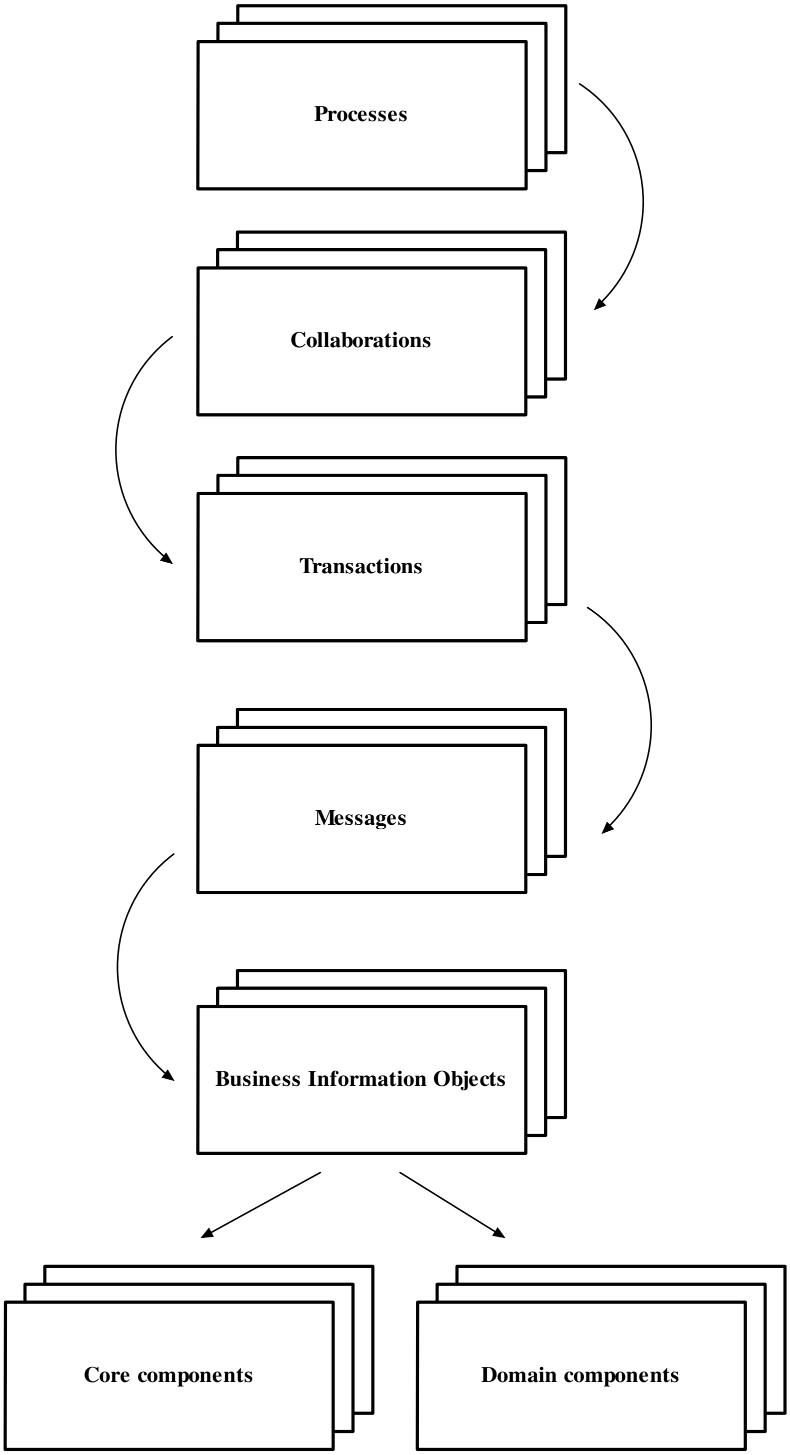
Size and weight are measurements obviously found in many industries, and while measurements can vary depending on the context (as described in the previous paragraph), we can identify these core components and apply them across industries. The paper industry, however, measures weight in a unique way. It uses the concept of basis weight, with the weight of paper defined in North America as the weight in pounds of a ream (500 sheets) cut to a basis size for a particular grade. Paper for newspapers uses a sheet 24 x 36 inches as the basis size, while paper used to print books and magazines use a sheet 25 by 38 inches. Outside North America, the industry uses grams per square meter for basis weight calculations. [22]

See next page for Figure 3.

[21] For the curious, ISO 216 derived that ratio from 1:Square root of 2 = 1:1.4142

[22] Again for the curious, basis weight is used in standard industry formulas to measure the usable yield from a roll or skid of sheeted paper, an indicator of paper quality.

Figure 3. Relationship of core components to business information objects and business processes



Documentation for paper shipments still must have gross weights that shippers, transportation companies, and customers need to plan for delivery and handling. These gross weights will be core components, with the context determining if the shipments use English or metric measurements. Basis weight however, is used only in the paper industry and will therefore be a domain component in electronic business documents.

With Figure 3 as a model, we can use our paper industry example to show the relationship between processes and core and domain components in a business message.

Process: Inventory shipment

Collaboration: Paper mill, newspaper

Transaction: Shipping/receiving

Message: Ship notice, sent from the mill to newspaper printing plant

Business information object: Newsprint roll shipped (a uniquely identified object)

Core component: Gross weight
Core component context: Geographic location/USA - English measurement, pounds or short tons (2000 lbs).

Domain component: Basis weight
Domain component: Basis size (24 x 36 inches)

Business process catalogue

The analyses of business processes in ebXML led to a cataloging of common business process patterns, independent of specific industries, and thus available for use across various lines of business. The collection of processes proposed by the business process team in ebXML represents at this stage a first pass at a catalogue, and therefore is not considered a complete list. EbXML published this first catalogue as a technical report in May 2001. The processes listed below provide examples of the kinds of business activities and the granularity of detail at which business processes are defined.

The ebXML business process team created the catalogue to encourage development of templates for use by different industries. The catalogue should also help in the analysis and discovery of collaboration patterns within those processes, and grow into a global tool to which all industries can contribute and from which they can derive benefits.[23]

Part of the method for discovery of business processes and collaboration patterns within those processes, is the discovery of core components in business messages. The catalogue includes cross-references to processes and messages from the EDI standards (ASC X12 and UN/EDIFACT) and leading XML-based specifications, such as RosettaNet and Open Applications Group. The repeated appearance of components across those standards and specifications suggests that the data item is indeed a core component available for re-use. Business process is also one of the important context categories for core components in a business document, which can change the precise meaning of the data item, sometimes dramatically.

The report notes that business process analysis and discovery can become complex and detailed. The development of a more complete business process catalogue will help encourage development of software tools, such as a business process editor, that can help companies and industries do this job more easily.[24]

Business processes in the catalogue are categorized according to a system proposed by Michael E. Porter in his 1998 book discussing the idea of a value chain. According to Porter, the value chain takes a set of inputs from a business process and increasingly generates outputs of greater value for customers. The value chain identifies the processes that transform capital, labor, and raw materials into finished goods and services of value to customers. The categories recommended by Porter and adapted by ebXML include:

- Financing
- Administration
- Procurement
- Human resources
- Transportation
- Manufacturing
- Customer service
- Marketing and sales [25]

[23] "Design Objective," ebXML Catalog of Common Business Processes, v1.0, ebXML, 11 May 2001, p 4-5.

[24] "Business Process Catalog Use Cases," ebXML Catalog of Common Business Processes, v1.0, ebXML, 11 May 2001, pp. 5-6.

[25] Michael E. Porter, Competitive Advantage : Creating and Sustaining Superior Performance, 1998, Harvard Business School Press

The catalog has 247 cross-industry processes. Many of the processes can be applied to a wide range of businesses, while others cross only a few industry lines. A number of examples are given in Table 1 below:

Table 1. Examples of ebXML business processes

Business process	Value-chain category
Request quote/Response	Procurement management
Query price	Procurement management
Query price and availability	Procurement management
Price catalog	Procurement management
Establish auction offer	Procurement management
Check credit	Procurement management
Establish payment terms	Procurement management
Create invoice (products)	Financial
Create invoice (services)	Financial
Receive payables	Financial
Originate payables	Financial
Originate child support payments	Financial
Originate payroll	Financial
Confirm/verify payment orders	Financial
Transmit mortgage loan info	Financial
Transmit a debit/credit adjustment	Financial
Receive receivables	Financial
Customs declaration (shipper)	Transportation and logistics
Export and import declaration	Transportation and logistics
Voyage itinerary/schedules	Transportation and logistics
Entry of cargo	Transportation and logistics
Departure of cargo	Transportation and logistics
Vessel/vehicle departure plan	Transportation and logistics
Vessel/vehicle load operation	Transportation and logistics
Cargo relay operation	Transportation and logistics
Manage sales lead	Marketing management
Query sales lead status	Marketing management
Distribute marketing activity info	Marketing management
Notify of marketing opportunity	Marketing management
Distribute product list	Marketing management
Initiate manufacturing order	Manufacturing
Setup production line tooling	Manufacturing
Produce product	Manufacturing
Inspect work in progress	Manufacturing
Inspect finished goods	Manufacturing
Provide and administer asset mgmt	Service support
Technical support and service mgmt	Service support

The catalog also includes specific process from the insurance and education business domains.

Translating business process models to XML

Identifying and classifying business processes has value for development of e-business collaborations, but in the end, the systems designers still need to turn those abstract models into working systems. EbXML wrote a schema for the specification of business processes to serve as a bridge from the process models to the software design. This Business Process Specification Schema comes in two versions, UML and XML, each directly mapped to the other.

The UML version is written as a class diagram, with all of the specification elements and relationships needed to create an ebXML business process specification. The XML version is presented both as a document type definition or DTD used in XML 1.0 and meeting the W3C's XML Schema specifications.[26]

A company or industry can write the XML version of the specification schema without having to develop a UML model. The output of the of the XML specification schema goes directly into the Collaboration Protocol Profile (CPP) that describe the technical e-business capabilities of companies and the Collaboration Protocol Agreement (CPA) that serves as the configuration files for the ebXML business service interface software that generates the ebXML messages.[27]

While the business process specification schema serves as a direct input into much of the rest of the ebXML architecture, it does not directly define the precise contents of the business documents. Business documents are defined using an external document specification or assembled from core components. However, business processes provide much of the context or meaning for the core components. Therefore the business documents exchanged in ebXML are formed as a partnership between the business process specification and core components.[28]

[26] "Goals/Objectives/Requirements/Problem Description," ebXML Business Process Specification Schema Version 1.01, 11 May 2001, page 2

[27] "System Overview," ebXML Business Process Specification Schema Version 1.01, 11 May 2001, pp. 5-6.

[28] "How ebXML Business Process Specification Schema is used with other ebXML specifications," ebXML Business Process Specification Schema Version 1.01, 11 May 2001, page 14.

4. Core Components: Definition and Discovery

The ebXML business messages sent between companies contain vital information for performing the services or delivering the goods that meet the needs of the customers for whom those goods or services are designed. Messages contain business information objects, the chunks of data with information needed to conduct the business in question. The business information objects, in turn, can contain pieces of data unique to that line of business, or more generic data items used by many industries.

Those generic pieces of data are called core components, and are the key to achieving interoperability among business messages. They serve as interchangeable building blocks that can be used across business messages. Each core component has a neutral syntax and unique identifier. These two properties make it possible to relate the data item used in a specific message to its counterpart in other industries, as well as to EDI messages.

This chapter provides a better understanding of the origin and identification of core components. It offers a background into the structure of core components and the naming conventions that reflect that structure. It also discusses the methods used to discover and derive core components. Please note that most businesses will not need to perform the analysis and discovery steps, nor will they need to identify and name their own core components. Industry groups and standards bodies will do that work, as reflected in the ebXML-compliant specifications developed by those organizations.

Core component structure

As discussed in Chapter 3, core components work in a larger structure of business information objects and business documents, fitted together with the industry context and industry-specific domain components. Core components themselves have a structure and hierarchy as well. At the lowest level is the basic component (the formal name is basic information entity) that expresses a single piece of data in a message. Related basic components can also be connected together in aggregates that express more complex data items.[29]

[29] “Conceptual Picture of Core Components,” Core Component Overview Version 1.05, ebXML, 10 May 2001, pp. 8-9.

An example will illustrate this structure. Financial account identifiers used by financial institutions, such as banks, are actually a collection of data strung together in an aggregate component. Business documents that direct or authorize funds transfers need an account number to identify the bank accounts from where the funds originate or to where they will go. But the financial institutions often need other related pieces of information besides the account number to take any action.

One of the ebXML core components collects these related data items together into an aggregate component called financial account details. This aggregate core component is made up of the following basic components:

- Financial account.identifier, such as bank account number
- Financial account.name, the full name on the account used for complete account identification
- Financial account.country.code
- Financial account.currency identification.code
- Financial account.product type.identifier, such as checking or investment account
- Financial account.nickname.name, familiar name given to the account[30]

Basic and aggregate components contain some of the business semantics that give the components meaning. In the above example with financial account data, the term financial.account precedes each of the core component names, and represents the object class in those names. The components also have other terms such as identifier, name, and code. This second group of terms, called representation types, indicate the form that core components take in business documents.

Representation types are derived mainly from core component types (CCTs) that provide a classification method useful in identifying common data items across industries. In most cases, the CCT corresponds directly to a single representation type. In a few instances, however, a CCT can be represented in more than one way. The CCTs and representation types include:

[30] ebXML Core Components Structure v1.04, pp. 18-19

Core component type	**Representation type**
Code	Code
Identifier	Identifier
Date time	Date Time Date and time
Amount	Amount
Quantity	Quantity
Text	Text Name
Measure	Measure

EbXML also identified several representation types related to common data types used to describe the format of data items:[31]

Data type	**Representation type**
String	Content Value
Boolean	Indicator
Decimal	Percent Rate

Representation types are used in the core component naming conventions discussed below. The definitions of the representation types follow:[32]

[31] "Result of Analysis," Guide to the Core Components Dictionary, ebXML, 10 May 2001, pp. 8-9.

[32] "List of Representation Types," Naming Convention for Core Components, ebXML, 10 May 2001, page 8.

Representation type	Definition
Amount	Number of monetary units specified in a currency where the unit of currency is explicit or implied.
Code	A character string – letters, figures or symbols -- that for the sake of brevity and / or language independence may be used to represent or replace a definitive value or text of an attribute. Codes usually are maintained in code lists for each attribute type (e.g. color).
Date	A day within a particular calendar year, using the format specified by ISO standard 8601.
DateAndTime	A particular point in the progression of time.
Identifier	A character string used to identify and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme.
Indicator	A list of two, and only two, values which indicate a condition such as on/off; or true/false. (synonym: boolean)
Measure	A numeric value determined by measuring an object. Measures are specified with a unit of measure. The applicable units of measure are taken from UN/ECE Recommendation 20 document.
Name	A word or phrase that constitutes the distinctive designation of a person, place, thing, or concept.
Percent	A rate expressed in hundredths between two values that have the same unit of measure.
Quantity	A number of nonmonetary units, associated with the indication of objects. Quantities need to be specified with a unit of quantity
Rate	A quantity or amount measured with respect to another measured quantity or amount, or a fixed or appropriate charge, cost or value, such as US Dollars per Euro.
Text	A character string generally in the form of words of a language.
Time	The time within a (not specified) day, using the format specified by ISO standard 8601.

Core component naming conventions

EbXML adopted a consistent and systematic format for core components dictionary entry names, based on ISO standard 11179, Guidelines for Structured Naming Conventions, and that reflect the structure of core components described above. These naming conventions help in understanding the purpose of the components.

Names for basic components consist of three parts:

1. Object class, the logical data group in which the data element belongs

2. Property term, distinguishing characteristic of the business entity

3. Representation type, the form of the set of valid values for the element; see the list of representation types given above.

In the examples given above, for financial account details, most of the entries have three parts separated by a period (.). The term financial account is the object class and used in all entries. The second part of the entry name is the property term that describes the specific property that distinguishes it from the other entries in the class. In the earlier example, identifier, name, country, currency, product type, and nickname are all properties of a financial account, thus those terms are used in the second part of the core component name.

The third part of the name is the representation type, and the financial account examples use the terms identifier, name, and code from the list of representation types given above. However, the first two entries in the group of financial account details have only two parts, not three:

1. Financial account.identifier
2. Financial account.name

When the property term is the same as the representation type, the conventions recommend using the term only once rather than repeating the term in the name.[33]

[33] "Basic Information Entities – data element level," Naming Convention for Core Components, ebXML, 10 May 2001, pp. 5-7

Aggregate components have two parts, an object class and a property term. All aggregate components use the property term details. Thus the name for the group used in our examples is financial account.details.[34]

Analysis and discovery of core components

In most cases, the discovery and analysis of core components is a process that industry groups or standards bodies will perform, not individual companies. However, this briefing gives an overview of the process to provide some background on their derivation and maintenance.

The ebXML team working on core components recommended a series of systematic steps for their discovery and analysis, but these steps first require a set of conditions in place and accepted by all interested parties:

- Rules for definition of core components
- Procedures and authorities for resolving conflicts on industry-specific components
- Processes and organizations across industries for the addition of new or updated items listed in registries
- Registries for ebXML business process models and core components

Because of the close alignment of core components with business processes, the discovery of core components starts with business processes. The steps begin with a comparison of business processes and if the processes differ, they then move on to a review the individual core components. Where entire new processes or variances from existing processes are found, the reviewers need to drill down and find new or updated business documents. These new or revised documents will be one source for new core components.

For the individual core components generated in this exercise, reviewers need to compare the components against those already listed in a registry. For those components already listed, the procedures recommend documenting the new users of those components in the businesses, industries, or regions being introduced. Where the components are similar but not identical, the analysis shifts to the context. If the context is different, then the reviewers need to document that context or (if no difference in context) then resolve the conflicts between components. Where

[34] "Naming of Aggregate Information Entities," Naming Convention for Core Components, ebXML, 10 May 2001, page 9.

both the component and context are different, then the reviewers have a candidate for a new component added to the registry.35

The ebXML core component team also prepared recommendations for handling requests for new business processes and core components by standards bodies, referred from the industry groups conducting the discovery of processes and components described above. The recommended procedures address different contingencies, such as the need to reuse or extend processes and components, add new contexts, or create entirely new items.36

It pays to remember that these procedures are only proposed and have yet to be field-tested. However, having even these proposed steps will help get the organizations geared up and processes in place for handling these issues.

Reusability, the purpose of core components

Once discovered and catalogued, core components make it possible to reuse data objects with the same meanings in different messages, processes, and industries, with only the business context changing. This reusable property – perhaps interchangeability is a better word – encourages interoperability at a detailed semantic level, which makes it easier to move data both between and within businesses.

Without this interoperability, companies must physically map the data from one electronic document to another, tracing the data back to the company business systems that use the data. This mapping process provides a level of integration within companies, but at a steep price. Maintaining these inter-document maps is an arduous task, one not made any easier in this time of rapidly changing business conditions.

With core components, however, each of these interchangeable and semantically equivalent parts is already identified and connected in the business document, as well as to the larger business process. Companies need to map their internal data to the core components, a one-time exercise, not to each data element in each document. Because core components have unique identifiers and a neutral syntax, the company can generate the map with a great deal more ease, since the core components are already identified in the business documents. As companies add or change trading partners or even take on new lines of business, they can extend the use of these components as they occur in new documents.

[35] "Discovery and Analysis," Core Component Discovery and Analysis, 10 May 2001, pp. 8-10.

[36] "Harmonization Analysis Activity," Core Component Discovery and Analysis, 10 May 2001, pp. 11-13.

While the core components stay the same, the context changes from one document, process, industry, and role to the next. This context adds the specificity to core components and makes them meaningful in business documents. 37 The next chapter discusses the role of context in core components, and the assembly of components in business documents themselves.

Chapter 7 gives a real-life example of core components in common business documents and in particular shows how the reusability of core components opens up new business opportunities and provides better service to the customers.

37 "Promoting Interoperability," Context and Re-Usability of Core Components, 10 May 2001, pp. 12-13.

5. Context and Assembly in Business Messages

Core components have the curious distinction of doing much, but by themselves meaning little. As discussed at several points in this briefing, they play an important part in encouraging interoperability in business semantics, with benefits to businesses that better integrate the data in e-business messages in their company systems and throughout the supply chain. But by themselves, core components have little meaning, and that is by design. The real meaning of these components in business messages comes from the business context. This chapter describes the important role played by context, the systematic definition of context, and the assembly of components and context into business documents.

Context gives meaning to core components

Core components need the addition of context – the substance of business activity – to provide the real meaning to the parties in the transactions. This substance, however, can be expressed in a number of different ways, and in almost any combination. As a result, business documents need to consistently and systematically identify that context if they hope to express the meaning of the business information objects contained in those documents accurately.

EDI messages also express business context, but in a different way. The EDI standards define business documents, called messages (UN/EDIFACT) or transaction sets (ASC X12), as made up of data segments, which in turn are composed of related data elements. Both data segments and data elements are used in many documents as interchangeable parts. EDI adds the specific industry knowledge in code lists used with many data elements. In some cases, industry groups add entire lists of codes directly to the standards. In other cases, they reference external code lists used in a specific industry. In recent years, the ASC X12 standard introduced entire transaction sets for single industries, the most notable for business messages used in health care.

For example, in the ASC X12 standard, the Date/Time (DTM) segment is made up of several data elements expressing date (year, month, day) and time (hours, minutes, seconds, time zone). The DTM segment also includes qualifiers with long lists of codes expressing the many purposes a business message can express date and time. In some cases the codes expressed generic business uses of date and time, e.g., date of invoice, scheduled date/time of delivery, and contract start date. In other cases, the codes express uses in particular applications, such as accounting period start and end.

EbXML takes a different approach. EbXML defines the core component separate from the context, then adds the context as needed to define the content of the information object in the business document. The addition of context takes place when assembling the components plus contexts into information objects and then into electronic business documents.

This process, however, is not always simple and straightforward. As discussed in Chapter 4, basic components can join together to become aggregate components, providing at least some meaning to the component even before bringing context into the picture. Therefore, the precise way context gets added to components will vary among industries, processes, and messages.[38]

EbXML lists several scenarios where the identification of context can encourage interoperability:

- *Same data, varying labels.* This scenario occurs whenever industries develop their own terminology to express the same ideas. As discussed in the Introduction, different industries can use the terms vendor or supplier or manufacturer interchangeably, and relating these terms accurately is one of the objectives of ebXML

- *Same data, different position in the document.* In this situation, the same data appear in different messages, or even in the same message, but used in a different way. In a purchase order, for example, the buyer may have the same bill-to as ship-to address, but the data are used by different parts of the vendor company and generate different actions.

- *Same data, varying processes.* In this scenario, the meaning of the data varies because the process makes different use of the information. For example, where a customer orders some goods and has the shipment delivered to a third-party warehouse, both the purchase order and the ship notice will identify the customer as the buyer. In this scenario, however, the buyer plays a much more important role in the purchase order than in the ship notice.

- *Same data, different culture.* This scenario can occur where different laws or traditions require different ways of expressing data. For example, the display of currencies, prices, dates, and times, can vary due to local requirements or practices.

[38] "Using Context Descriptors," Context and Re-Usability of Core Components, 10 May 2001, pp. 8-9.

Context categories or drivers

EbXML has proposed an initial set of six categories of context that can add to the meaning of core components. Because they generate the context or meaning added to the component, these categories are known as context drivers, each of which is discussed below.

Business process

Business processes are behind much of the definition of business content in ebXML, and as discussed earlier, they contribute significantly to the definition of core components, especially in the definition of context. Business processes identify the parties, collaborations, transactions, messages, and business information objects in the messages. As the process changes, the purpose of the object and the actions generated as a result of the object, will change as well.

The financial account.details component discussed earlier offers an example of the business process context. This component plays an important part in the processes of checking a company's credit and collecting on receivables, both listed as core processes in the ebXML catalogue. The use of the component in those processes, however, differs markedly. In the case of checking credit, it verifies to the vendor company that the customer has an account with this name and number. In the case of collecting receivables, it provides an address from where the customer transfers the funds.

In this and other contexts, complications can arise. The overall process may give indicators, not precise meanings to the components, where the trading partners need to go to a finer level of detail. For example, a company may have the same ship-to as bill-to address, but the ship-to address used in a purchase order will likely not have a post office box number, and may include precise locations, such as loading dock number.[39]

Geographic

The geographic context, called regional context in the ebXML documentation, explains differences in meaning of components from one part of the world to another. In this context, geography can be expressed at one or more levels:

- Global
- Continent
- Economic region, e.g. NAFTA, EU

[39] "Business Process Context," Catalogue of Context Drivers version 1.04, 10 May 2001, pp. 6-7.

- Country
- Sub-national region

This type of context also has complexities. Geographic context can be expressed as a single value or in combinations. The combinations can include established aggregates, such as Benelux (Belgium, Netherlands, Luxembourg), or cross-border arrangements, such as at Niagara Falls, New York and Ontario. For identifying national and sub-national contexts, ebXML recommends the codes listed in ISO standard 3166, the recognized international country code list.[40]

Official constraints

This type of context refers to external documented sets of rules, ranging from legal mandates to industry best practices. The types of official constraints include:

- National laws and regulations, including customs regulations
- Treaties and international conventions, which usually carry the force of law
- Standards, such as those issued by ISO and ANSI
- Guidelines, including defacto standards and industry best practices
- Contracts, including trading partner agreements[41]

For example, a product identifier, as expressed in the core component product.service.details (000155), will have a much different meaning when represented in a industrial product catalogue, as opposed to a hazardous materials list required by local or national regulations.

Product

The use of core components in messages involving different products and services can generate significant differences in the meanings of those components. For example, the component transport means.details (000116) will have a different meaning in a purchase order for goods, where the means of transport is requested by the buyer, as opposed to an oder to a trucking company for those very delivery services. EbXML lists several sources of product classifications from the UN and World Trade Organization.[42]

[40] “Regional Context,” Catalogue of Context Drivers version 1.04, 10 May 2001, page 7.

[41] “Official Constraints Context,” Catalogue of Context Drivers version 1.04, 10 May 2001, page 8.

[42] “Product Context,” Catalogue of Context Drivers version 1.04, 10 May 2001, page 8.

Industry

Differences in industry use of data will make an impact on the meaning of that data, since industries often develop their own terminology to meet specific business needs. We discussed earlier how the paper industry uses the term 'merchant' to refer to wholesaler. Measures (core component 000152) will also vary from one industry to another, as in when different industry conventions have their own definitions of gross or net weight. EbXML lists two sources of recommended industry classifications from the UN.[43]

Role

As discussed earlier, business processes often define the role of a party in a transaction or collaboration, and that role can markedly change the meaning of a component. Companies often play different roles, as buyers and sellers of goods or services, investors, or employers, depending on the type of transaction or collaboration. The ebXML business process methodology, derived from the UN's unified modeling methodology, lists several types of roles a party can play:

- *Organizational*. In most e-business relationships (or general business relationships for that matter), companies act on their own behalf, with the authorization to represent their own interests.

- *Employee*. Employees act as representatives of organizations, but in limited and defined roles. For example, specific staff members of an organization may have authorization to hire contractors for periods of time, but not extend full-time job offers. As a result, the context of that role can affect the meaning of data in components identifying parties in these transactions.

- *Functional*. The functional role covers the contingency when an employee has full authorization to act on behalf of the organizational, for example, when the CEO agrees to terms of a contract.

- *Initiator*. This role, and the responder (in the following bullet), refer to actions taken as part of business processes rather than in organizations or on their behalf. As the name suggests, the initiator begins a process and represents its initial state, for

[43] "Industry Context," Catalogue of Context Drivers version 1.04, 10 May 2001, page 9.

example, a vendor company publishing its parts catalog, and sending the catalog to its regular customers.

- *Responder*. The responder role interacts with the initiator in the process, taking an action that follows in the process. Continuing with the example in the previous bullet, a company receiving the vendor's new catalog could issue a new order using the part numbers and prices listed in the catalog.[44]

Chapter 7 gives a real-life example of common business messages and their potential use of core components. The example shows each data item in the messages, the applicable core component from the current library, and the context driver from this list of six categories.

Assembling business documents with core components and context

The ebXML core components technical team recommended rules for systematic assembly of business documents by combining core components and context. The team discovered that despite the relatively simple idea behind the idea of automated assembly of electronic business documents, the execution turned out to be a much more complex task.

One of the complexities arose in resolving conflicts in rules from accommodating multiple contexts. For example, an industry practice may exclude extended product descriptions from business messages, especially where companies have a vendor-managed inventory arrangement. In a vendor-managed inventory, customers often use the vendor product numbers since the vendor is responsible for replenishing the customer's supplies while keeping customer inventory levels to a minimum.

However, if the shipments are crossing borders, customs authorities will need to visually read the documentation and sometimes inspect the shipment itself. Thus the documents accompanying the shipment will need to include the extended product descriptions as well as the product codes. Component context and business document assembly rules need to accommodate these kinds of contingencies. [45]

The core components team proposed electronic rules, written as XML document type definitions (DTDs), for determining context and building business documents. The group's technical report cautions that the DTDs it prepared give examples of the solutions, and should not be interpreted as normative specifications.[46]

[44] "Role Context, Catalogue of Context Drivers version 1.04, 10 May 2001, pp. 9-10.

[45]"Context rules," Document Assembly and Context Rules version 1.04, 10 May 2001, pp. 8-9.

[46] "Introduction," Document Assembly and Context Rules version 1.04, 10 May 2001, page 5.

The `ContextRules DTD` lists the rules that can be applied in the contexts used in this document. Each rule has an attribute called `apply` with values of `exact` and `hierarchical`. The value `exact` is used if the conditions for applying the rule require an exact match between the value in the provided context and the specifications of the rule. For example, if a rule applies only when prices are quoted in U.S. currency (Geographic context driver, nation = USD), the rule would use the value `exact` with the attribute `apply` for this contingency.

The value `hierarchical` applies when the rules require either an exact match with the value of the context or a child element in its hierarchy. Extending the currency example above, if a certain context rule applies when using currencies from any of the NAFTA member countries (Geographic context driver, economic region = NAFTA), then the rule would accept the value NAFTA, as well as the values USD, CAD, and MXP for the subordinate individual country currencies.

The DTD contains the conditions under which a rule should be run and the desired actions when the rule is run. It includes as well an attribute for the prescribed order for applying the rules, with numerical values assigned to the rules, and the values with the higher numbers run first. For rules with hierarchical conditions, the rules list taxonomies that define the applicable hierarchies.

The `Assembly` DTD creates a separate DTD with the XML syntax for a business document, identifying the individual elements or groups of elements created or used in the document. If the elements are retrieved from a registry, the DTD includes a `location` attribute with the registry address – either a unique identifier in the registry or a uniform resource identifier – of that element. The `Assembly` DTD can create multiple business documents, if the implementation requires it.[47]

While the DTDs are considered examples and not prescriptive, the last ebXML proof of concept tests in May 2001 did include a live trial of these DTDs. The demo included a sample invoice that generated its business information objects from core components and contexts of industry, role, and geography. The demo successfully generated the elements and assembled them in an XML document. [48]

[47] “XML Based Rules Model,” Document Assembly and Context Rules version 1.04, 10 May 2001, pp, 10-14.

[48] Alan Kotok, “ebXML: It Ain't Over 'til it's Over,” XML.Com, 16 May 2001, http://www.xml.com/pub/a/2001/05/16/ebxml.html

6. Current work on core components

This part of the briefing describes the current compilation of core components and discusses work underway to build on this start. As mentioned in the Introduction, as of Summer 2001, work continues on defining core components and will likely for a while. Because it involves the expression of business ideas and actions, also known as business semantics, it needs the involvement of business people to succeed. When core components and ebXML overall begin to get widespread use, they will likely have a significant and beneficial impact on business. Therefore, the more that business people contribute their needs and ideas to this work, the more they willbenefit. This chapter describes the ongoing efforts defining core components in which business people are invited to participate.

Current core components

The ebXML core components team identified a number of components for use in business messages and that serve as examples for future work defining these business objects. The team applied its discovery and analysis methods, as outlined in Chapter 4, and generated a first group of core components. The team also categorized and listed properties of the components to provide additional details for prospective users.

The first part of the current list has the core component types, which you can think of as the inner-core of the core components. These core component types, also described in Chapter 4, give the functions of the components and offer little meaning on their own. The eight core component types include:

1. *Code* - A character string – letters, figures or symbols – that for brevity and/or language independence may be used to represent or replace a definitive value or text of an attribute. Codes usually are maintained in code lists for each attribute type (e.g. color).

2. *Identifier* - A character string used to identify and distinguish uniquely, one instance of an object within an identification scheme from all other objects within the same scheme.

3. *Date time* - A particular point in the progression of time.

4. *Amount* - Number of monetary units specified in a currency where the unit of currency is explicit or implied.

5. *Quantity* - A number of non- monetary units, associated with the indication of objects. Quantities need to be specified with a unit of quantity.

6. *Text* - A character string generally in the form of words of a language.

7. *Measure* - A numeric value determined by measuring an object. Measures are specified with a unit of measure.

The catalogues give the aggregate components (known as aggregate information entities), and list the basic components it contains for each of these aggregate components. The main catalogue, the core component dictionary, includes the following information for each entry:

- Name
- Definition
- Unique identifier (UID)
- Synonyms
- Component reused, if applicable
- Data type (such as string, integer, or boolean), used with most basic components
- Remarks
- Core component type, one of the eight types listed above
- Naming convention, defined as object class, property term, and representation type (Chapter 4)[49]

The dictionary of core components is available on the Web at http://www.ebxml.org/specs/ccDICT.pdf .

The core components team issued a separate listing of the components, in the form of a spreadsheet, called core component structures. This spreadsheet gives for each component:

- Unique identifier or UID
- Aggregate name
- Embedded entity name, the name of basic components included under aggregate components

[49] "Dictionary Overview," Guide to the Core Components Dictionary version 1.04, 10 May 2001, pp. 8-10.

- Core component type, one of the eight types described above
- Data type, applicable to most basic components
- Component reused, if applicable
- Category type, core component type, aggregate, or basic
- Requirement indicator, R = required
- Definition
- Remarks, examples or references[50]

The core component structures document is found on the Web at http://www.ebxml.org/specs/ccSTRUCT.xls .

Appendix 1 gives a list of the names and unique identifiers for this first group of core components. Chapter 7 gives examples of core components in common business documents and shows how even this limited list can be used now in day-to-day transactions.

Continuing development work on core components

After completion of ebXML's initial phase in May 2001, the two sponsors of ebXML – OASIS and UN/CEFACT – divided up the work to continue its development. OASIS took the work on XML infrastructure and UN/CEFACT inherited the work on business content that includes business processes and core components. In the first phase of ebXML, most teams completed at least one specifications document, but the ebXML core components teams had to issue a series of technical reports.[51] As a result, the business content development work, especially the work on core components, still continues.

One of the divisions of UN/CEFACT is the UN/EDIFACT Work Group (EWG), the organization responsible for the UN/EDIFACT standard used as the basis for most EDI transactions outside North America. Even before the end of the first phase of ebXML, representatives of EWG and the X12 Committee, accredited by ANSI for EDI standards in North America, agreed to jointly carry on developing core components. That joint project kicked off in June 2001.[52] Four more meetings are scheduled by March 2002.

[50] "Format of Core Component Structures," Guide to the Core Components Dictionary version 1.04, 10 May 2001, pp. 10-11.

[51] "EbXML approved: UN/CEFACT and OASIS Deliver on 18-Month Initiative for Electronic Business Framework," ebXML, 14 May 2001, http://www.ebxml.org/news/pr_20010514.htm .

[52] "ASC X12 & EWG Develop ebXML-compliant Core Components Enabling Global Trade Using EDI in X12 & XML Formats," Data Interchange Standards Association, 21 June 2001, http://www.disa.org/pr_doc.cfm?Name=588 .

The EDI standards organizations bring important credentials to the table for this project. Many X12 and EWG members took part in ebXML, and most represent end-user companies or industry organizations, rather than equipment or software vendors. Because they bring as much (if not more) business as technical expertise, they can represent business needs in this important exercise of defining the core semantics in e-business messages.

Like the ebXML initiative itself, this joint X12/EWG effort is conducted entirely by volunteers and contributions of time by interested parties is always welcome. For more information about the project, visit the ASC X12 Web site at: http://www.x12.org/x12org/meetings/jcc-stlouis1.pdf, or at the UN/EWG Web site, http://www.edifact-wg.org/ .

Full disclosure: The author is a staff member of the Data Interchange Standards Association that serves as the secretariat for ASC X12.

7. Core Components Example

In this part of the briefing, we offer an example of core components in business documents, and how the interoperability of core components can open up new business opportunities. This example is based on a reallife experience, one of my recent business trips. At the end of June 2001, I gave a talk at a conference in Toronto, Canada, and used the occasion to visit family in Buffalo, New York, about 90 miles (55 km) from Toronto.

As I started making the travel plans, I discovered no airline offered direct service between Buffalo and Toronto. And for various reasons, the prices for flights between the Washington, DC area airports and Toronto were much more expensive than Buffalo. In fact, one discount air carrier had amazingly low fares for jet service between Baltimore and Buffalo.

While I could fly as far as Buffalo, I still had to get to and from Toronto. I did not want to drive, and thus pay for parking a car in Toronto that I would not use for three days. The remaining choices were scheduled busses or rail. When I discovered that the rail coaches had electrical outlets in which I could plug my laptop, that made my choice easy,[53] and the fares were quite reasonable. The itinerary resulted in:

27 June	Air	Baltimore to Buffalo
	Rail	Buffalo to Toronto
		Toronto hotel check-in
30 June		Toronto hotel check-out
	Rail	Toronto to Buffalo
	Air	Buffalo to Baltimore

Between the air and rail segments, I hung out with my brother and his family in Buffalo.

The business documents and data presented below come from the air, rail, and hotel reservation responses, all made on-line directly with the vendors. The sample documents list only the data objects, not the values. They also simplify some of the processes. For example, the rail schedules divide at the U.S./Canada border, where Amtrak serves only the U.S. side and VIA, the Canadian passenger rail service, serves the Canadian side.

[53] Much of this document was written during the rail segments of the trip.

The columns show the major data items and sub-items if applicable, along with ebXML core component name, number, and context category that provides specificity and meaning to the component.

Table 1.

Sample reservation responses with ebXML core components

Major item	**Sub item**	**Core component name**	**Number**	**Context**
Air ticket				
Carrier name		organization. name	000115	Industry
Confirmation number		product service. identifier	000156	Product
Passenger name	Family name	person. surname. name	000020	Industry
	Given name	person. given. name	000018	Industry
	Middle name	person. middle. name	000019	Industry
Frequent flyer number		party. identifier	000016	Industry
Passenger address	Address type	address. type. code	000024	Industry
	Number	street building. identifier	000026	Industry
	Street	street. name. text	000027	Industry
	City	town. name. text	000029	Industry
	State	state. identifier	000030	Industry
	Postal code	post code. identifier	000031	Industry
	Country	country. code	000032	Industry
Passenger telephone	Type of contact	communication. mode. code	000053	Industry
	Telephone number	communication. identifier	000055	Industry

Major item	Sub item	Core component name	Number	Context
Passenger e-mail	Type of contact	communication. mode. code	000053	Industry
	E-mail address	communication. identifier	000055	Industry
Outbound flight number		product service. identifier	000156	Industry
Out departure date		product service start. dateandtime	000159	Industry
Out departure time		product service start. dateandtime	000159	Industry
Out departure airport code		location. identification. code	000061	Industry
Out arrival date		product service end. dateandtime	000160	Industry
Out arrival time		product service end. dateandtime	000160	Industry
Out arrival airport code		location. identification. code	000061	Industry
Inbound flight number		product service. identifier	000156	Industry
In departure date		product service start. dateandtime	000159	Industry
In departure time		product service start. dateandtime	000159	Industry
In departure airport code		location. identification. code	000061	Industry
In arrival date		product service end. dateandtime	000160	Industry
In arrival time		product service end. dateandtime	000160	Industry
In arrival airport code		location. identification. code	000061	Industry

Major item	Sub item	Core component name	Number	Context
Ticket price		charge price. amount	000127	Product
Taxes		tax. amount	000149	Official
Credit card	Type of payment	financial account product. type. identifier	000084	Official
	Account number	financial account. identifier	000077	Official
	Expire date	payment card. expiration. date	000083	Official
	Name on card	financial account. name	000078	Official
Rail ticket				
Carrier name		organization. name	000115	Industry
Confirmation number		product service. identifier	000156	Product
Passenger name	Family name	person. surname. name	000020	Industry
	Given name	person. given. name	000018	Industry
	Middle name	person. middle. name	000019	Industry
Frequent rider number		party. identifier	000016	Industry
Passenger address	Address type	address. type. code	000024	Industry
	Number	street building. identifier	000026	Industry
	Street	street. name. text	000027	Industry
	City	town. name. text	000029	Industry
	State	state. identifier	000030	Industry
	Postal code	post code. identifier	000031	Industry
	Country	country. code	000032	Industry
Passenger telephone	Type of contact	communication. mode. code	000053	Street
	Telephone number	communication. identifier	000055	City

Major item	Sub item	Core component name	Number	Context
Passenger e-mail	Type of contact	communication. mode. code	000053	State
	E-mail address	communication. identifier	000055	Postal code
Outbound train number		product service. identifier	000156	Country
Out departure date		product service start. dateandtime	000159	Industry
Out departure time		product service start. dateandtime	000159	Industry
Out departure station code		location. identification. code	000061	Industry
Out arrival date		product service end. dateandtime	000160	Industry
Out arrival time		product service end. dateandtime	000160	Industry
Out arrival station code		location. identification. code	000061	Industry
Inbound train number		product service. identifier	000156	Industry
In departure date		product service start. dateandtime	000159	Industry
In departure time		product service start. dateandtime	000159	Industry
In departure station code		location. identification. code	000061	Industry
In arrival date		product service end. dateandtime	000160	Industry
In arrival time		product service end. dateandtime	000160	Industry
In arrival station code		location. identification. code	000061	Industry

Major item	Sub item	Core component name	Number	Context
Currency for pricing		currency. identification. code	000133	Geographic
Ticket price		charge price. amount	000127	Product
Taxes		tax. amount	000149	Official
Credit card	Type of payment	financial account product. type. identifier	000084	Official
	Account number	financial account. identifier	000077	Official
	Expire date	payment card. expiration. date	000083	Official
	Name on card	financial account. name	000078	Official
Hotel room reservation				
Property name		organization. name	000115	Industry
Confirmation number		product service. identifier	000156	Product
Guest name	Family name	person. surname. name	000020	Industry
	Given name	person. given. name	000018	Industry
	Middle name	person. middle. name	000019	Industry
Guest frequent stay acct		party. identifier	000016	Industry
Guest address	Address type	address. type. code	000024	Industry
	Number	street building. identifier	000026	Industry
	Street	street. name. text	000027	Industry
	City	town. name. text	000029	Industry
	State	state. identifier	000030	Industry
	Postal code	post code. identifier	000031	Industry
	Country	country. code	000032	Industry
Guest telephone	Type of contact	communication. mode. code	000053	Industry
	Telephone number	communication. identifier	000055	Industry

Major item	Sub item	Core component name	Number	Context
Guest e-mail	Type of contact	communication. mode. code	000053	Industry
	E-mail address	communication. identifier	000055	Industry
Check-in date		product service start. dateandtime	000159	Industry
Check-out date		product service end. dateandtime	000160	Industry
Room type (Smoking/No)		product service. type. code	000157	Product
Bed size		product service. type. code	000157	Product
Currency for pricing		currency. identification. code	000133	Geographic
Daily rate		charge price. amount	000127	Product
Taxes		tax. amount	000149	Official
Credit card	Type of payment	financial account product. type. identifier	000084	Official
	Account number	financial account. identifier	000077	Official
	Expire date	payment card. expiration. date	000083	Official
	Name on card	financial account. name	000078	Official

A reading of these lists shows that each business document uses most of the same data elements, and reusing the data across the three documents would save the customer time and probably improve data integrity for the three companies. But what does ebXML or core components bring to the table here? One could call this example a routine travel industry case, already handled by groups like the Open Travel Alliance (www.opentravel.org).

To answer the question, remember that I went to Toronto to attend a conference, which had an entirely separate registration process. The conference registration form asked for the following data:

- Registrant’s name
- Company name
- Address
- Telephone number
- E-mail address

- Registering for conference only
- Or conference and tutorials?

- Form of payment

If we put these items in a table with core components, they may look like Table 2.

Table 2.

Conference registration form data items with core components

Major item	Sub item	Core component name	Number	Context
Registrant name	Family name	person. surname. name	000020	Industry
	Given name	person. given. name	000018	Industry
	Middle name	person. middle. name	000019	Industry
Company name		organization. name	000115	Industry
Registrant address	Address type	address. type. code	000024	Industry
	Number	street building. identifier	000026	Industry
	Street	street. name. text	000027	Industry
	City	town. name. text	000029	Industry
	State	state. identifier	000030	Industry
	Postal code	post code. identifier	000031	Industry
	Country	country. code	000032	Industry
Registrant telephone	Type of contact	communication. mode. code	000053	Industry
	Telephone number	communication. identifier	000055	Industry

Major item	Sub item	Core component name	Number	Context
Registrant e-mail	Type of contact	communication. mode. code	000053	Industry
	E-mail address	communication. identifier	000055	Industry
Participating in:	Conference only, or	product service. type. code	000157	Product
	Conference and tutorials	product service. type. code	000157	Product
Amount charged		charge price. amount	000127	Product
Form of payment	Type of payment	financial account product. type. identifier	000084	Official
	Account number	financial account. identifier	000077	Official
	Expire date	payment card. expiration. date	000083	Official
	Name on card	financial account. name	000078	Official

This little example shows some of the potential of ebXML and core components. All of the items in the registration form are identical or similar to the data collected for travel reservations. Yet, in the vast majority of conferences or meetings, registrants need to make their own travel arrangements, separate from the meeting registration. The fees hotels charge conference organizers are based in large part on the number of hotel rooms blocked off for the conference, and failure to meet that number of rooms results in penalties. It is not unusual for hotels and conference organizers to carry on a long and detailed reconciliation to determine the accurate numbers of guests credited to an event.

Why don’t hotels add meeting registrations at their properties as a value-added and income-producing service? Or why can’t conference organizers make hotel reservations for registrants, and in return get charged a lower fee from the hotel properties? In both cases, it would result in a common list of event attendees registered at the hotel, making the post-conference reconciliation much less work.

One reason up to now for not adding these services is the lack of compatibility in systems between conference organizers and hotels. A standard like ebXML, with interoperable business semantics provided by core components, removes much of that obstacle.

The important message for business people from this little example is that ebXML helps remove many of these barriers, encouraging more and different kinds of collaborations than before. By helping remove these barriers, the potential contributions of ebXML and core components are immense.

Appendix 1.

Names and identifiers of current core components

Source: ebXML Core Components Structure v1.04, 10 May 2001

http://www.ebxml.org/specs/ccSTRUCT.xls

UID	Aggregate Core Component	Embedded Basic Core Component
000066	date time. type	n/a
000067		date time. content
000068		date time. format. text
000089	code. type	n/a
000091		code. content
000092		code list. identifier
000093		code list. agency. identifier
000099		code list. version. identifier
000100		code. name
000075		language. code
000090	text. type	n/a
000094		text. content
000075		language. code
000101	identifier. type	n/a
000102		identifier. content

000103		identification scheme. name
000104		identification scheme agency. name
000075		language. code
000105	amount. type	n/a
000106		amount
000107		amount currency. identification. code
000108	quantity. type	n/a
000109		quantity
000110		quantity. unit. code
000111		quantity unit code list. identifier
000112		quantity unit code list agency. identifier
000152	measure. type	n/a
000153		measure. content
000154		measure unit. code
000001	party. details	n/a
000016		party. identifier
000009		party. description. text
000002		party. type. code
000095		person. details
000096		organization. details
000095	person. details	n/a

000098		person. name
000014		person. tax. identifier
000012		birth. date
000018		person. given. name
000019		person. middle. name
000020		person. surname. name
000021		person. gender. code
000113		person. nationality. code
000022		person. name prefix. code
000114		person. name suffix. code
000096	organisation. details	n/a
000115		organization. name
000097		organization registration. country. code
000015		organization registration. date
000013		organization. tax. identifier
000005	postal address. details	n/a
000024		address. type. code
000026		street building. identifier
000027		street. name. text
000028		district. name. text
000029		town. name. text
000030		state. identifier
000031		post code. identifier
000032		country. code

000033		post office box. identifier
000034		building. identifier
000035		building. name
000036		suite. identifier
000037		mail delivery. sub-location. identifier
000038		floor. identifier
000039		lot. identifier
000040		block. identifier
000041		district sub-division. identifier
000042		region. identifier
000043		county. identifier
000045		postal address. line 1. text
000046		postal address. line 2. text
000047		postal address. line 3. text
000048		postal address. line 4. text
000049		postal address. line 5. text
000050	communication number.	n/a
000052		communication. type. code
000053		communication. mode. code
000055		communication. identifier
000057		telephone country. identifier
000058		telephone area. identifier
000059		telephone subscriber. identifier
000060		telephone extension. identifier

000006	location. details	n/a
000061		location. identification. code
000062		location. type. code
000063		location. description. text
000010	financial account. details	n/a
000077		financial account. identifier
000078		financial account. name
000080		financial account. country. code
000081		financial account currency. identification.
000084		financial account product. type. identifier
000085		financial account nickname. name
000116	transport. means. details	n/a
000117		transport means. identification. code
000118		transport means. name
000119		transport means. nationality. code
000165	packaging. details	n/a
000166		packaging. type. code
000167		packaging. quantity
000168		packaging. level. code
000169		packaging. material. code
000155	product service. details	n/a
000156		product service. identifier

000157		product service. type. code
000158		product service. description. text
000159		product service start. dateandtime
000160		product service end. dateandtime
000163		product service classification. identifier
000136	currency exchange. details	n/a
000137		source currency. identification. code
000138		target currency. identification. code
000139		base currency. identification. code
000120		currency. exchange. rate
000140		source currency. scale. value
000141		target currency. scale. value
000142		exchange. rate. dateandtime
000122	charge price. details	n/a
000127		charge price. amount
000143		charge price. description. text
000130		charge price. tax inclusion. indicator
000126		percentage charge price. details
000125		unit charge price. details
000126	percentage charge price.	n/a
000144		charge price. percentage
000145		chargeable. amount

000125	unit charge price. details	n/a
000146		unit charge price. amount
000147		base charge price. quantity
000121		chargeable. quantity
000148	tax. details	n/a
000149		tax. amount
000150		tax. description. text
000151		tax. category. code
000171		tax. percentage
000172		taxable. amount

Appendix 2.

EbXML Specifications and Technical Reports

Specifications

Document Name	Download Address
ebXML Technical Architecture Specification v1.04	http://www.ebxml.org/specs/ebTA.pdf
Business Process Specification Schema v1.01	http://www.ebxml.org/specs/ebBPSS.pdf
Registry Information Model v1.0	http://www.ebxml.org/specs/ebRIM.pdf
Registry Services Specification v1.0	http://www.ebxml.org/specs/ebRS.pdf
EbXML Requirements Specification v1.06	http://www.ebxml.org/specs/ebREQ.pdf
Collaboration-Protocol Profile and Agreement Specification v1.0	http://www.ebxml.org/specs/ebCCP.pdf
Message Service Specification v1.0	http://www.ebxml.org/specs/ebMS.pdf

Technical reports

Document Name	Download Address
Business Process and Business Information Analysis Overview v1.0	http://www.ebxml.org/specs/bpOVER.pdf
Business Process Analysis Worksheets & Guidelines v1.0	http://www.ebxml.org/specs/bpWS.pdf
E-Commerce Patterns v1.0	http://www.ebxml.org/specs/bpPATT.pdf

Catalog of Common Business Processes v1.0	http://www.ebxml.org/specs/bpPROC.pdf
Core Component Overview v1.05	http://www.ebxml.org/specs/ccOVER.pdf
Core Component Discovery and Analysis v1.04	http://www.ebxml.org/specs/ebCCDA.PDF
Context and Re-Usability of Core Components v1.04	http://www.ebxml.org/specs/ebCNTXT.pdf
Guide to the Core Component Structure and Dictionary v1.04	http://www.ebxml.org/specs/ccCTLG.pdf
Naming Convention for Core Components v1.04	http://www.ebxml.org/specs/ebCCNAM.pdf
Document Assembly and Context Rules v1.04	http://www.ebxml.org/specs/ebCCDOC.pdf
Catalogue of Context Drivers v1.04	http://www.ebxml.org/specs/ccDRIV.pdf
Core Component Dictionary v1.04	http://www.ebxml.org/specs/ccDICT.pdf
Core Component Structure v1.04	http://www.ebxml.org/specs/ccSTRUCT.pdf
Technical Architecture Risk Assessment v1.0	http://www.ebxml.org/specs/secRISK.pdf
ebXML Glossary	http://www.ebxml.org/specs/ebGLOSS.pdf

About the Author

Alan Kotok is a Washington, DC, reporter and writer, and editor of E-Business Standards Today (published by Data Interchange Standards Association (www.disa.org/dailywire/), a daily news wire and weekly newsletter on standards affecting e-business. He is a frequent contributor to XML.Com and editor of the U.S. Techno-Politics page on Suite101.Com (www.suite101.com/welcome.cfm/us_techno_politics). He has written articles and columns for technology, business, and public policy publications such as Electronic Commerce World, Datamation, Publishing and Production Executive, and Foreign Service Journal.

Before joining DISA, Kotok served 10 years with Graphic Communications Association as Director of Management Technologies, then as Vice-President for Electronic Business, responsible for educational and technical programs in EDI, bar codes, and later XML. Before joining GCA, Kotok founded Overseas Technology, a high-tech export company, and served 15 years with U.S. Information Agency on assignment in the USA, Middle East, Africa, and Japan.