

As the business community braces for the explosive growth in B2B eCommerce, cXML is leading the way in creating new buyer and seller relationships.

# cXML

by Kevin Clements,  
Chief Technology Officer, SHI

This is an interesting time in the evolution of business-to-business electronic commerce. Industry pundits agree that we are poised for explosive growth in B2B commerce in the next few years – growth that will make the boom in the web-based consumer market pale by comparison.

In order for this market to grow to its potential, we will need to develop ways to integrate buyer and seller systems into a single network, capable of efficiently exchanging information between them.

To understand why, a brief trip through history is in order. We'll begin with a quick look through the short, but furious, evolution of B2B eCommerce supplier sites.

## First generation – brochureware

In the beginning, many businesses deployed simple web sites best described as brochureware. They took existing content and layouts from their existing marketing collateral, and converted them to HTML pages for delivery over the web. Companies were overjoyed to be “on-the-net,” and had not yet had an opportunity to think about how to exploit the possibilities of this new way of communicating.

## Second generation – on-line shopping

The next generation of B2B sites essentially mimicked the growth of consumer-oriented shopping sites. They provided customers with a way to shop a supplier's wares, place orders, and obtain information on the

status of those orders. They were often built with the same tools as consumer sites, and worked in much the same way.

Some suppliers began to offer services above and beyond those meant for an individual consumer by providing such things as customized catalogs and pricing, detailed product specifications, and realtime inventory information.

Still, each B2B site was essentially an island, consisting of a single, self-contained application. The links from the site were bridges that could convey a user from one of these islands to the other.

## Third generation – one-off interfaces

Many B2B suppliers began to grow specialized interfaces to small sets of their own suppliers, and their customers. Some B2B sites became capable of querying their own suppliers for inventory information or to generate estimated lead times or delivery schedules. Interfaces to clients evolved to handle some authentication issues, or to allow completed orders to be posted to customer financial systems.

This generation of sites exhibited one major problem – they did not scale particularly well. In most cases, these interfaces were implemented as one-off, special projects. A company wanting to integrate its systems to its suppliers found that it needed to negotiate with each supplier to get the desired interfaces built, requiring a mixture of blackmail, cajolery, and





# - Toward a new Web Order

bribery. Suppliers, faced with the prospect of building and supporting similar, but unique interfaces to each customer who demanded it, found their web teams stretched thin.

## The rise of procurement systems

Our customers have come to view managing the cost of procurement as a key to improving the profitability of their companies. They are investing in systems that empower their employees to do much of their own purchasing, and leverage the information they already have in their human resources systems and financial systems to help them control what happens.

In order for these systems to function, they need a

method to interface to suppliers. They need to exchange a host of information, including:

- ◆ Product catalogs
- ◆ Pricing and availability
- ◆ Quotes
- ◆ Orders

As some of these systems grew, customers and suppliers found that there was a heightened need to develop standardized, flexible interfaces between their systems. The adoption of these procurement systems offered tremendous opportunities to buyers and suppliers, but only if the infrastructure to support them could be evolved.

## Enter XML

Fortunately, as the need for these flexible interfaces became apparent, the Extensible Markup Language (XML) was there to help provide them. XML is a deceptively

simple text format that allows information to “describe” itself by carrying information about the structure of the data within the file containing the data. For example, a set of XML data describing an item in a catalog may look something like this:

```
<catalog_item>
  <item_number>1234-ABC</item_number>
  <description>Super Widgets</description>
  <manufacturer>Widget Ware</manufacturer>
  <price>99.95 </price>
</catalog_item>
```

One of the nice things about XML is that this data can be read by humans and computers with equal aplomb. The “tags,” enclosed in angle brackets, allow

both the human and computer to understand what elements of information are present, where they begin and end, and how they are hierarchically related.

An agreed upon structure for an XML file can be documented using a structure such as a Document Type Definition (DTD). A DTD describes a contract about the set of elements (or tags) that can be found in an XML document, including such things as:

- ◆ The set of tags that will be found in the file;
- ◆ The relationships between them (e.g., a catalog item is described by an `item_number`);
- ◆ Which elements are optional or mandatory;
- ◆ What type of information is stored in a tag (e.g., description is a text string, price is a number, etc.).

These DTDs allow the sender and receiver of an XML document to validate that a piece of XML matches the rules agreed to and written in the DTD. Most XML parsers available today can quickly and easily compare the structure of an XML document with the type defini-



tion meant to describe it, and can tell you if the file meets the agreed upon standard or not.

XML is an ideal solution to the problems we have exchanging information between buyer, supplier and market maker systems. Why?

- ◆ The format is machine and platform independent;
- ◆ It's simple and manageable;
- ◆ It can be mechanically validated efficiently, allowing you to write systems that can easily detect and defend themselves against invalid messages;
- ◆ Support for it is rapidly becoming pervasive—most web development or deployment platforms provide some support for it today.

### Ariba's cXML

As wonderful as XML is for B2B commerce, it doesn't quite solve all of the problems linking buyer and supplier systems. It does provide the structures and the basic plumbing that we need; it does not, in and of itself, tell us how we will describe a part in a catalog, or a line in an order. It simply gives us the tools to do so.

In theory, a buyer and supplier could sit down at a table and come to an agreement as to the pieces of information they'll exchange for a catalog item. They could then document that agreement using a Document Type Definition so that they each could be sure they were living up to the agreement, and begin exchanging information. Unfortunately, that would take us back to where we started: Buyers and sellers needing to negotiate the set of information they'll pass back and forth. XML might make that process simpler, and provide them with new tools to do it well, but it would not alter the fundamental problem.

What's actually needed to solve our problem is a standard, agreed upon set of XML elements and structures that can be adopted by buyers, suppliers and market makers. Instead of reinventing the wheel each time we need to establish a new buyer and seller relationship, we could simply agree to adopt this existing standard.

One evolving standard is cXML. Originally proposed and adopted by Ariba, cXML is a set of standardized XML structures, documented in a set of Document Type Definition files. In some ways, the choice of names is a little unfortunate; quite a number of people new to XML have asked me if cXML is a proprietary extension to XML. It is not. It is simply an agreed upon set of standard DTDs.

The cXML DTDs cover much of the information required for B2B eCommerce systems to exchange information. They span the description of goods offered for sale, the exchange of shopping carts from supplier sales systems to buyer procurement systems, and the routing and distribution of complete orders.

cXML has a number of advantages for the B2B supplier:

- ◆ The basic DTDs cover most of the elements you will routinely need to describe products and orders. Some competing formats only define a minimal set of elements, meaning you cannot effectively convey a rich set of information to describe a product.
- ◆ The format has support for adding information unique to a buyer/supplier relationship through an XML

element called an *extrinsic*. This allows you to add information to an XML document without diverging from the standard.


- ◆ The elements in the standard already make provision for multi-language and multi-currency issues others are only beginning to plan for;
- ◆ It's been battle-tested by use in the Ariba procurement system.

### Toward the future

If you read the industry trades, you see that there are many consortia, suppliers, and standards bodies that are beginning to address the need to adopt standards, such as cXML, to allow electronic commerce to really live up to its potential. Adopting a standard such as cXML today allows you to keep current with this trend.

Ariba and Microsoft are currently working to integrate cXML with Microsoft's BizTalk XML framework for B2B communications. BizTalk is an emerging framework for integrating applications of all kinds: supply-chain management, ERP systems, logistics systems, and

more. It's designed to help software companies, service providers, and customers integrate their systems using an XML-based platform. BizTalk defines rules for including XML schemas and tags in documents to provide BizTalk-enabled applications with document handling and routing information. They act as a common "envelope" for the business information to be transmitted, making it possible for applications from many different suppliers to cooperate.

Support for cXML is growing today, and Ariba's work to integrate cXML with the BizTalk framework should mean that adopters of cXML will be well positioned to use it to participate in the larger networks to come. The cXML standard provides a solid foundation for carrying procurement related documents today, and BizTalk support should allow it to inter-operate with other systems tomorrow. 

---

*Software House International, Somerset, N.J. ([www.shi.com](http://www.shi.com)) is a global corporate value-added reseller of software, hardware, and related services.*