

XML Explained

There has been a lot of talk recently about XML (Extensible Mark-up Language), a data format for standard document exchanges on the Web. Therefore, what does it do, what does it look like, what are its advantages, and why is it of importance for B2B E-commerce?

XML development commenced in 1996, so you would be forgiven for thinking that this is an immature technology. In fact, XML is a subset of SGML (Standard Generalized Mark-up Language), that was developed in the early 1980's and is widely used for large documentation projects. Perhaps more familiar is HTML (Hypertext Mark-up Language), whose development began in 1990 and is also a subset of SGML. But unlike HTML, XML is not in a fixed format; it is a metalanguage that lets users design their own mark-up languages to meet system or application requirements. XML is used for marking up data in ways that reflect its meaning rather than its presentation as opposed to HTML whose mark-up is specifically related to the presentation of information in a browser. Simply stated, XML provides a communication standard for B2B trading that includes both the data being shared and the data definitions. In other words, it's like sending an encoded message along with the secret decoder ring in one bundled message.

One of the advantages that XML offers is that its data format can be read by both computers and humans. The fact that XML documents can be structured hierarchically into sets of related data elements can make it easier for humans to understand as this is the traditional model we find in documentation, with its hierarchies of chapters, sections, subsections, paragraphs, etc. The same hierarchical structure allows computer programs to identify subsets of messages that need to be processed as a unit, for example to prompt users to provide a particular piece of information within a form, or to load a set of fields into a particular database record.

XML Structure

An XML *document instance* must be created and stored as a set of properly nested data storage *entities*, each of which is made up of a number of logical *elements* that contain data or define processes to be performed. The outermost storage entity is referred to as the *document entity*: it contains both the start and the end of the *root* or *document element* of the document instance. Elements can be nested to create hierarchies (information trees). Elements can be assigned *attributes* (properties) which indicate how the contents of the element should be interpreted.

Each XML element starts with a named *start-tag* and ends with an *end-tag* with a matching name. Because each element of an XML document has clearly marked limits, it is easy to determine when its contents have been received over a network. Attributes of XML elements are defined as part of its start-tag (e.g. <Order type="production">). Each XML attribute must be fully defined,

with the *attribute name* followed by a *value indicator* (=) and a quote delimited string containing the *attribute value*.

XML and B2B Transactions

XML provides unlimited flexibility when exchanging data between businesses and machines. Data can be exchanged with or without human intervention. For example, using XML a business can

- Define a document such as a Purchase Order (PO) with a valid data structure.
- Display this XML PO through a browser or a 'XML Parser'
- Integrate XML documents with internal applications like E-catalogues
- Pass on this XML PO directly to partners or intermediaries who can interpret the document without any prior knowledge of data structure

The B2B global data exchange standards that were established by ANSI and UN have been evolving as a result of changing requirements of B2B E-commerce. The emergence of the Internet has fuelled the need for technologies that fosters real-time application to application, web to application and web to web integration. This is where XML fits in. It accomplishes the need for real-time, Internet-centred solutions, providing businesses a flexible and extensible environment for integrated and interoperable data exchange. XML standardization makes it possible to use message brokers or shopping agents to search the Internet for specific information, no matter how disparate the systems are.

Advantages

XML offers the following benefits to businesses conducting transactions over the web:

Application integration. With XML, standard formats for data interchanges can be defined and XML templates that integrate EDI transactions with other business processes and workflow automation tools can be created.

Cost savings. By using generic software tools and technical skills to eliminate the need for specialized EDI resources, XML/EDI makes exchanging business documents more cost-effective.

Faster processing. Many Web-based applications are slow because a server typically must perform a new query and create a new HTML document each time a client requests data from the server. With XML/EDI, an HTML document still defines how data is displayed, but the data itself is provided to the client through XML. By letting clients browse, search, and view data locally, this division of labour eliminates the need for multiple queries to the server, which updates displayed information only as needed. This in turn, means faster access to information for the user.

Dynamic mapping and routing. XML/EDI scripting can specify how XML-defined elements or objects must be processed. For example, scripting may define how a document is to be displayed or, if the document is an EDI transaction, how it is to be mapped. Scripting can also further define rules to integrate transactions into existing workflow processes or back-end applications.

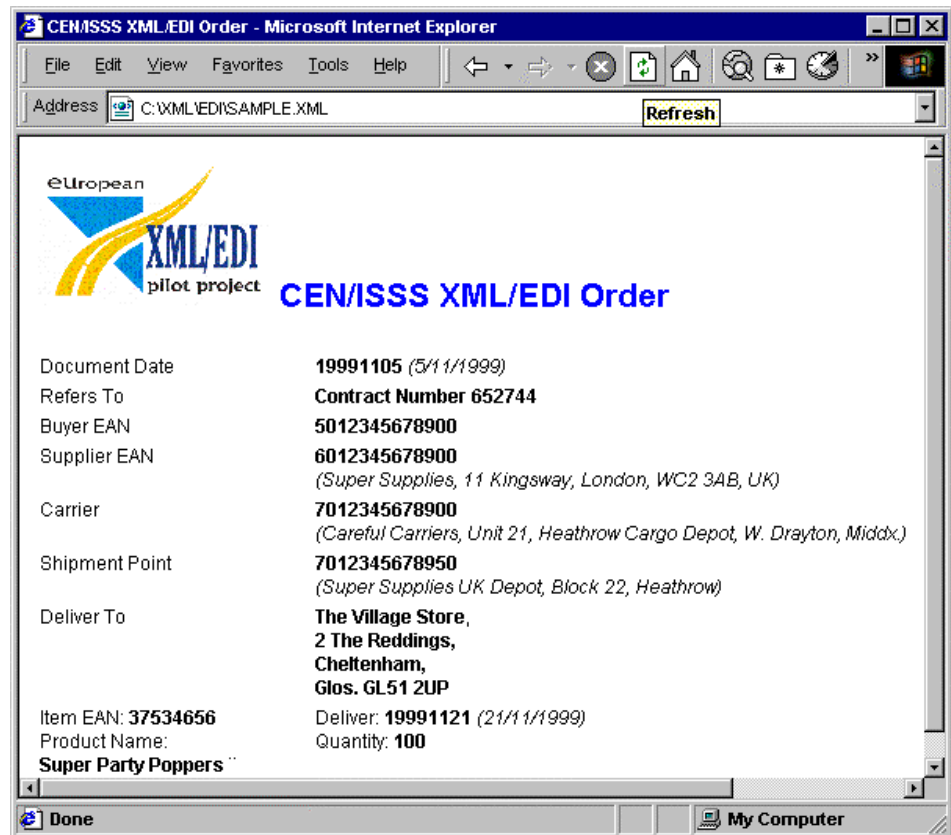
Forms for seamless integration. To help integrate XML with your current EDI processes, you can have your users complete forms that generate XML/EDI messages to be transmitted via the Internet or traditional VANs. Once a standard XML/EDI-capable browser receives a message, the EDI data can be easily interpreted using predefined rules for displaying or integrating the data with application systems. This method eliminates the need for custom program development, which is often required in a traditional EDI environment.

An example of XML code to generate a Purchase Order

```
<?xml version="1.0" ?>
<!DOCTYPE Order SYSTEM "C:\xml\dtds\order.dtd">
<Order>
  <MessageID type="Autogenerate"/>
  <Date>19991105</Date>
  <RefersToDocType="ContractNo" DocID="652744"/>
  <Buyer>
    <EAN>5012345678900</EAN>
  </Buyer>
  <Supplier>
    <EAN>6012345678900</EAN>
  </Supplier>
  <OtherParty Role="Carrier">
    <EAN>7012345678900</EAN>
  </OtherParty>
  <OtherParty Role="ShipFrom">
    <EAN>7012345678950</EAN>
  </OtherParty>
  <OtherParty Role="DeliverTo">
    <Name>The Village Store</Name>
    <AddressLine>2 The Reddings</AddressLine>
    <AddressLine>Cheltenham</AddressLine>
    <AddressLine>Glos. GL51 2UP</AddressLine>
  </OtherParty>
  <Item>
    <ItemID Agency="EAN">37534656</ItemID>
    <ItemDescription>Super Party Poppers &trademark;</ItemDescription>
    <Quantity>100</Quantity>
    <Deliver>19991121</Deliver>
  </Item>
</Order>
```

The first line of the message indicates it is an XML message, and which version of XML it conforms to. The second line of the message identifies the document type definition that has been used to validate the structure of the message. The outward pointing angle brackets are the delimiters that separate XML mark-up from contents. The first word within each set of angle brackets indicates the name of the XML element. The word before each = sign represents an attribute name, and text between quotes following the = sign represents the attribute value. Text not between angle brackets represents element content. The name between "&" and ";" in the fifth from last line identifies a reference to an entity whose replacement text will be the trademark symbol.

XML documents can be transformed into displayable formats such as HTML or PDF using a standard known as the XSLT (Extensible Style Language Translation) standard. An example would be the generation of a purchase order from an XML document on a WWW browser using this standard.



Conclusions

Though XML was only intended to be an evolution of EDI to a Web-based environment, its flexibility and other advantages listed above is helping many companies use it as an alternative to traditional EDI. Because of their well-structured nature, XML applications are easier to understand, and therefore easier to implement, than existing EDI protocols. Coupled with the cost-effectiveness and general availability of XML tools, and their general ease of use, the adoption of XML/EDI techniques should significantly reduce the cost of starting to use Web based data exchange and transactions within business enterprises.

Michael Broberg/Lalith Reddy, Jun 2000