CS155b: E-Commerce

Lecture 15: March 20, 2001

Introduction to XML

Acknowledgement: R. Glushko and A. Gregory
Some Acronyms Used In This Lecture

HTML = Hyper Text Markup Language  
XML = Extensible Markup Language  
EDI = Electronic Data Interchange  
ERP = Enterprise Resource Planning  
MRP = Materials Requirement Planning
The XML Revolution

• The Web was created to publish information for people
  – “eyes-only” was dominant design perspective
  – Hard to search
  – Hard to automate processing

• The Web is using XML to become a platform for information exchange between computers (and people)
  – Overcomes HTML’s inherent limitations
  – Enables the new business models of the network economy
Extensible Markup Language

- Instead of a fixed set of format-oriented tags like HTML, XML allows you to create whatever set of tags are needed for your type of information.
- This makes any XML instance “self-describing” and easily understood by computers and people.
- XML-encoded information is smart enough to support new classes of Web and e-commerce applications.
Why XML?

Sample Catalog Entry in HTML
<TITLE> Laptop Computer </TITLE>
<BODY>
<UL>
<LI> IBM Thinkpad 600E
<LI> 400 MHz
<LI> 64 Mb
<LI> 8 Gb
<LI> 4.1 pounds
<LI> $3200
</UL></BODY>
XML’s Big Idea: Document Types

- Customer Profiles
- Vendor Profiles
- Catalogs
- Datasheets
- Price Lists
- Purchase Orders
- Invoices
- Inventory Reports
- Bill of Materials
- Payments
- Deposits
- Credit Reports
- Schedules
- Directories
- …whatever you need

In XML the formal definition of permitted elements, attributes, and the rules by which they combine is called a Document Type Definition or DTD or schema
Catalog Entry in XML

<COMPUTER TYPE="Laptop">
  <MANUFACTURER>IBM</MANUFACTURER>
  <LINE> ThinkPad </LINE>
  <MODEL>600E</MODEL>
  <SPECIFICATIONS>
    <SPEED UNIT = “MHz”>400</SPEED>
    <MEMORY UNIT=“MB”>64</MEMORY>
    <DISK UNIT=“GB”>8</DISK>
    <WEIGHT UNIT=“POUND”>4.1</WEIGHT>
    <PRICE CURRENCY=“USD”>3200</PRICE>
  </SPECIFICATIONS>
</COMPUTER>
Smart Processing with XML

- `<COMPUTER>` and `<SPECIFICATIONS>` provide logical containers for extracting and manipulating product information as a unit
  - Sort by `<MANUFACTURER>`, `<SPEED>`, `<WEIGHT>`, `<PRICE>`, etc.
- Explicit identification of each part enables its automated processing
  - Convert `<PRICE>` from “USD” to Euro, Yen, etc.
Traditional Business Models and Integration Requirements

Traditional models for electronic business are based on long-term, point-to-point, and tightly coupled relationships

- EDI is used here because high integration costs can be recovered over time
- Partners are more willing to invest in compatible IT infrastructure at each end or in middleware that creates a distributed application
Making Money in B2B

- Licenses and support
  - Traditional model, works for technology providers to B2B marketplaces
- Equity
  - But only if the B2B company can IPO
- XML has little to say about this
Making Money in B2B

• Transaction fees
  – What counts as a transaction?
  – Who pays the fees – buyers or suppliers?

• Market efficiency
  – Driving costs out of supply chain for all participants
  – Exploit & refine existing business relationships & experience

• XML is crucial to these concerns
XML and Information “IQ”

- Content/structure-based text objects: XML, SGML, databases
- Formatted electronic test: HTML, EDI, word processing files
- Unstructured electronic text: ASCII
- Printed text

More “processability”/reusability  Easier to translate to
DTDs, Parsers, and Validation

• From any DTD an XML parser can be generated that:
  – Reads a document instance (the XML data stream)
  – Identifies the markup in it
  – Create a processable form of some kind that is used by an application

• The parser can also test the XML document for conformance with the rules of the DTD
  – A document instance that follows the rules of the DTD is “valid”
DTDs And Validation

- XML Purchase Order Instance
- Purchase Order DTD
- XML Parser
- Some Processable Form
XML Schemas in Electronic Commerce

- Essential to treat Dates, Monetary amounts, *etc.* as datatypes to enable validation
- Schema inheritance and extension mechanisms allow custom versions of same document to co-exist
  - Software can distinguish extensions from standard document and decide whether or not extensions can be safely ignored
  - Trading partners can customize messages for specialized needs while standard message maintains backward compatibility
Connecting with HTML ("by eye")

**Problem**: Company 1 has no integration with order management → manual and error prone data entry
XML as Internet-Friendly Integration Technology

… exchange data in an application and vendor neutral format
Connecting using XML

Benefit: XML can be processed automatically with huge cost savings

Problem: Company 1 and Company 2 have to agree on document format
Business Processes are XML Document Exchanges

If you send me a **request** for a **catalog**, I will send you a catalog.

If you send me a **purchase order** and I can fill it, I will send you a **purchase order response**.
Significance of XML Document Exchange Architecture

- Document exchange is a natural way to think about doing business.
- Easy to provide “open” marketplace with 3rd party buying and selling apps
- Easy to add and maintain services
- Document exchange between marketplaces is fundamentally the same as within a marketplace.
- Services can be reused across marketplaces.
Functions of “Market Makers” in a Document Exchange Architecture

• Specifying document standards
• Routing documents between participants
• Providing standard interfaces for sharing services (registration, logistics, taxation, payment, etc.)
XML is *Part* of the Solution

- XML has the potential to enable a standards-conforming, open and extensible architecture for electronic commerce.
- XML standards could enable ubiquitous connectivity and interoperability and create the network effects of “describe once, {sell, buy} anywhere” and reusable marketplace services.