IBM WebSphere software platform for e-business

XML Review

Cao Xiao Qiang
Solution Enablement Center, IBM
May 19, 2001

Agenda

- **What is XML?**
  - **Why XML?**
- **XML Technology**
  - **Types of XML Documents**
  - **DTD**
  - **XSL/XSLT**
- **Available XML Tools from IBM**
- **Competitive Perspectives**
- **Summary**
- **For More Information...**

What is XML?

What is XML?

- EXtensible Markup Language
- New technology for web applications
- Based on tags
  - User-defined tags identify the *kind* of data represented
- Designed to easily represent data in a very portable manner
  - What is the key message of Java?
    - Code portability (i.e. Write Once, Run Anywhere)
  - What is the key message of XML?
    - Data portability

The difference is WebSphere.

What is XML?

XML and Java technology are the "yin and yang of cross-platform computing"

-- Jon Bosak
"Father of XML"

What is XML?

- **XML tags are user defined**
  - **Can be enforced by rules (covered later)**

```
<OBCustomerList>
 <customer>
   <cust-fname>Rob</cust-fname>
   <cust-lname>Roy</cust-lname>
   <account>
     <acct-number>1000</acct-number>
     <acct-balance>1000.25</acct-balance>
   </account>
 </customer>
</OBCustomerList>
```

```
<MyTuesdayDinner>
 <appetizer>Buffalo Wings</appetizer>
 <entree>Sirloin Steak></entree>
 <vegetable>Peas</vegetable>
 <desert>Ice Cream</desert>
</MyTuesdayDinner>
```

The difference is WebSphere.

Why XML?

- XML provides a standard way of exchanging data:
  - Business-to-Business (B2B)
    - B2B revenue $109 billion in 1999
    - B2B revenue $2.7 trillion by 2004 (Forrester)
  - How would B2B data exchange without XML?
    - EDI, TextFiles, PDF, proprietary messages?
- XML certainly has applicability in non-B2B environments as well

Bottom Line: XML simplifies B2B transactions the web

What's wrong with just HTML?

- HTML simply defines how something should be rendered
  - Does not define what the data is

CustList.html

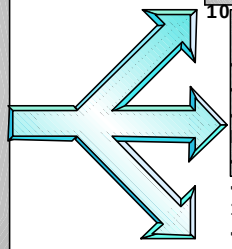
```
<p><b>Customer List</b>
<hr>
<br>Customer: Rob Roy
<br>Account # 1000
<br>Balance: 1000.25</p>
```

Why XML

- XML defines the data
- Rendering can easily be applied to XML using Style Sheets (XSL) (covered later)

CustList.xml

```
<customer>
  <cust-fname>Rob</cust-fname>
  <cust-lname>Roy</cust-lname>
  <account>
    <acct-number>1000</acct-number>
    <acct-balance>1000.25</acct-balance>
  </account>
</customer>
```



CustList.html

```
<p><b>Customer List</b>
<hr>
<br>Customer: Rob Ro
<br>Account # 1000
<br>Balance:
```

CustList.WM

```
<card>
<hr>
<br>Customer: Rob Roy
<br>Account # 1000
<br>Balance:
```

CustList.doc

```
Customer Rob Roy
 Account List
  Account # 1000
  Acct Balance: 1000.25
```

Key Benefits of XML

- Vendor and Platform Independence
- Easily model data at any level of complexity
- Extremely Extensible
  - Define new tags as needed
- Easily validate data to check for structural correctness
- Media independence to publish content in multiple formats:
  - e.g. HTML, WML, CompactHTML, etc.

Practical Uses for XML

- **B2B Communication**
  - Company A wants to exchange information with Company B
  - Company A doesn't need to understand how Company B organizes their data
  - Company A simply follows some simple rules when sending the data
    - These rules are defined in a DTD (covered later)
- **Smart Searches**
  - Search Engines can't process HTML intelligently
    - e.g.: Search for Rock
    - Results: Rock Bands, Rock Candy, Rock the movie, etc.
  - XML addresses data content

XML Technology

Anatomy of an XML Document

```
<?xml version="1.0" encoding="ISO-8859-1?
 standalone="yes">
<!-- OnlineBank.com customer list -->
<OBCustomerList>
 <customer id="12000">
    <cust-fname>Rob</cust-fname>
    <cust-lname>Roy</cust-lname>
    <account>
      <acct-number>1000</acct-number>
      <acct-balance>1000.25</acct-balance>
    </account>
 </customer>
</OBCustomerList>
```

Legend

XML Prolog		XML Element	
XML Comment		XML Attribute	

What are DTDs?

- **Document Type Definition**
- **"Schema" definition for XML documents**
  - **Defines the structure of an XML document**
  - **Defines what tags can go in an XML document**
  - **Defines what tags can contain other flags**
- **XML References DTD in <!DOCTYPE> XML prolog**

CustList.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OBCustomerList
 SYSTEM "CustList.dtd" >
<OBCustomerList>
...
```

Anatomy of a DTD

- **DTD Symbols:**

Symbol	Meaning	Example
ELEMENT	Declaration of Element	<!ELEMENT customer(name, email)>
ATTLIST	Attribute List for an element	<!ATTLIST customer gender(male|female)>
|	OR operator	<!ATTLIST person gender(male|female)>
?	Element is optional	<!ELEMENT customer(name,email?)>
+	Must be 1 or more elements	<!ELEMENT customer(name,account?>
*	Must be 0 or more elements	<!ELEMENT customer(name,account*>
#PCDATA	String Data	<!ELEMENT name (#PCDATA)

Anatomy of a DTD

- **The OBCustomerList element contains 0 or more customer elements**

CustList.DTD

```
<!ELEMENT OBCustomerList (customer)*>
```

The difference is WebSphere.

Anatomy of a DTD

- **The customer element contains:**
  - **1 cust-id element**
  - **1 cust-fname element**
  - **1 cust-lname element**
  - **1 cust-addr element**
  - **0 or 1 cust-email elements**
  - **0 or more account elements**

CustList.DTD

```
<!ELEMENT customer (cust-id, cust-fname,
cust-lname,
cust-addr ,cust-email?,account*)>
```

Three Types of XML Documents

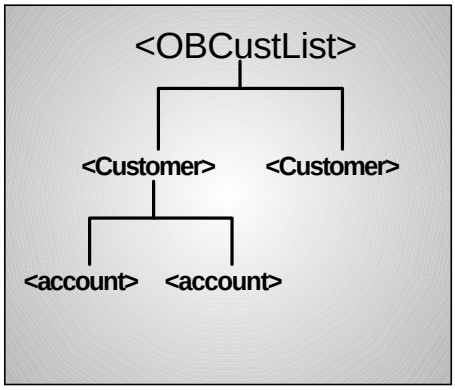
- **Well-formed Documents**
  - **Documents which follow the XML tag rules, but don't have a DTD**
- **Invalid Documents**
  - **Documents which doesn't follow the rules defined in its DTD**
  - **Documents which are incorrect syntactically**
    - **Invalid: <a>**
    - **Valid: <a>**
- **Valid Documents**
  - **Documents which follow the rules defined in their DTD**

Interpreting (Parsing) XML Documents

- XML Parsing is the mechanism used to interpret the content of an XML document
  - Two Approaches to Parsing XML documents
    - Document Object Model (DOM)
    - Simple API for XML (SAX)
- DOM is a standard of the World Wide Web Consortium (W3C)
- SAX is the "de-facto" standard for parsing XML documents

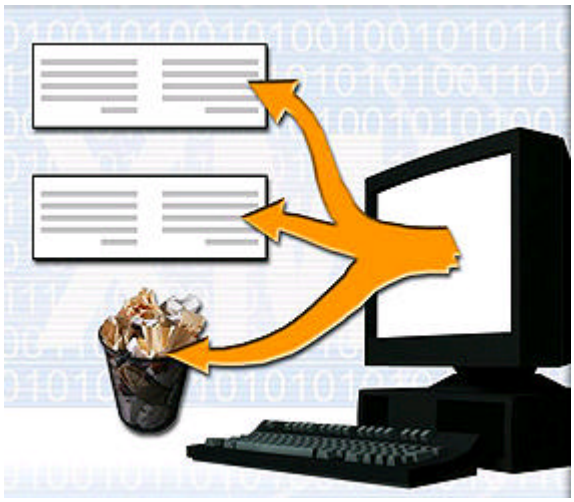
DOM Parsing

- DOM creates a tree view of your XML document
  - Recursion is used to navigate the tree



SAX Parsing

- **SAX is event driven**
  - Events are sent as the XML document is parsed
    - e.g. StartDocument, StartElement, EndElement, characters, etc.
  - Application program determines what elements are important



SAX vs. DOM

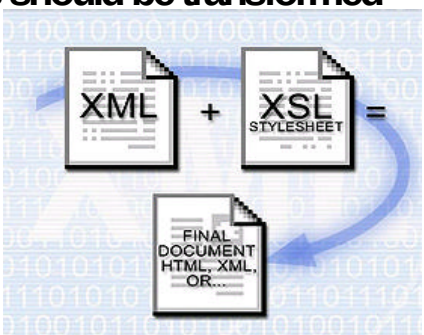
- DOM has richer set of functions for interpreting content in XML document than SAX
- DOM should be used when you need to know a lot about the structure of a document
  - In-memory tree constructed of XML document
- SAX has a simpler programming model
  - Event driven (SAX) vs. recursion (DOM)
- SAX uses less memory than DOM
  - SAX doesn't constuct the XML document in-memory

SAX vs. DOM

- **Use DOM when:**
  - **You need to use the information in an XML document more than once**
    - **DOM creates an in-memory tree which you can navigate many times while only parsing the XML document once**
  - **You need to know a lot about the structure of a document**
    - **e.g. Relationships between elements**
- **Use SAX when:**
  - **You need to parse/navigate XML document once**
  - **You need only a few elements in XML document**
  - **Small memory requirements**

XSL

- **XSL is a transformation and formatting language**
- **An XSL stylesheet is used to transform XML into another format**
  - **e.g. HTML, XML, etc**
- **XSL stylesheets contain templates that define how the XML elements should be transformed**



XSL Example

CustList.XSL

```
<?XML version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/Transform">

<xsl:template match="OBCustomerList">
 <html>
   <xsl:apply-templates/>
 <html>
</xsl:template>

<xsl:template match="customer">
  <P><B>Customer Found:
 <xsl:value-of select="cust-id"/>
  </B></P>
</xsl:template>
```

XML Element

Template Rule

XSL Transformation

- **Where can the transformation happen?**
  - **Client Side (with XML/XSL capable browser):**
    - **XML document and style sheet are served to client**
    - **Microsoft Internet Explorer 5.0 support this**
  - **Server Side:**
    - **Server applies XSL style sheet to XML document and sends the transformed document to the client**
    - **WebSphere with LotusXSL**
    - **AlphaWorks XML Enabler**
  - **A "third" program performs transformation and places transformed document on server**
    - **XT or Microsoft msxsl tool**
- **Server Side transformation are usually best**

The difference is WebSphere.

IBM

XML Tools from IBM

XML Tools from IBM

- **IBM XML Parser for Java**
  - Industry leading DOM/SAX XML parser
- **LotusXSL**
  - Server-side XSL processor
  - Available in WebSphere or AlphaWorks
- **XML Enabler**
  - Uses LotusXSL to implement server-side XSL processing
- **Visual XML Tools**
  - Visual DTD
  - Visual XML Creation
  - Visual XML Transformation
  - Visual XML Query

The difference is WebSphere.

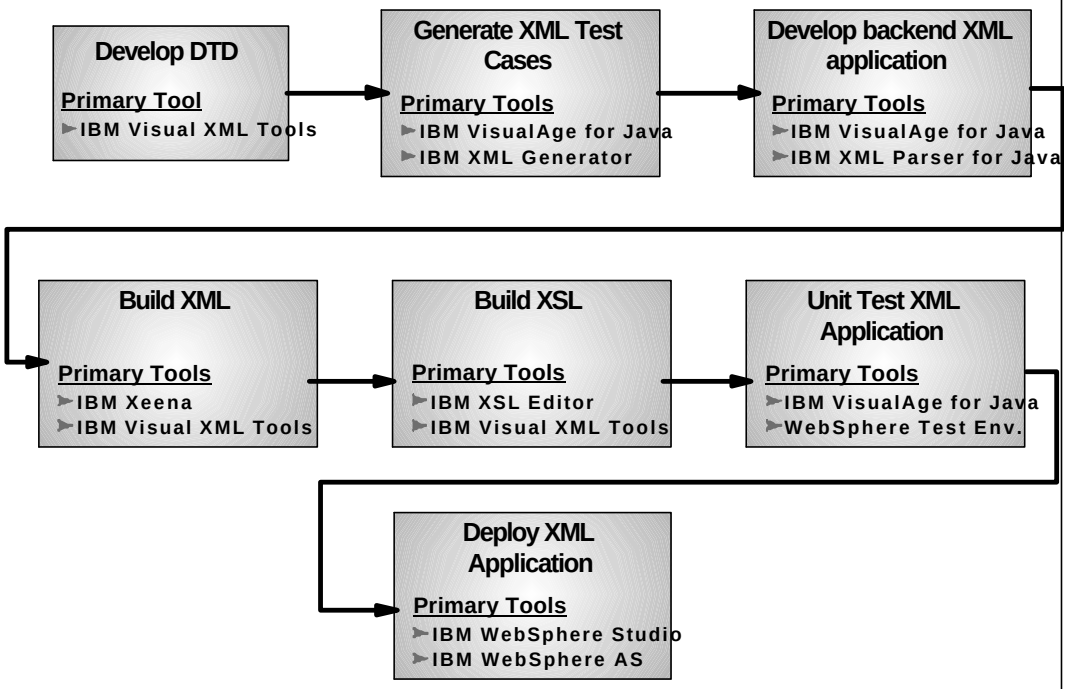
XML Tools from IBM *(continued)*

- Xeena
  - Visual XML authoring tool
- XML Viewer
  - View XML/DTD from a single window
- XML Generator
  - Generate XML test cases from a DTD
- XML Security Suite
  - Digital Signature, element-wise encryption
- XSL Editor
  - Visually edit/test XSL stylesheets
- Alphaworks contains 40+ XML tools/solutions

What does VisualAge for Java Support?

- Everything we've covered!
- VisualAge for Java 3.5 ships with:
  - IBM XML Parser for Java
  - IBM XML Generator
- Import LotusXSL to test/build XSL based solutions
- Many of the AlphaWorks Java based tools can be imported into VisualAge for Java as well
- Launch External XML tools from Resources Page

IBM XML Tooling: Usage Scenario



Competitive Perspectives

Competitive Perspectives

- Microsoft making major investments in XML
  - Have XML, XSL, DTD solutions today
    - XML Software Development Kit (SDK)
  - XML support in (or coming):
    - Windows 2000
    - Windows DNA 2000
    - Microsoft BizTalk Server 2000 (may slip to 2001)
      - aka Microsoft XML Server
    - Microsoft SQL Server 7.0
    - Microsoft Office 2000
    - SOAP
    - .NET framework

IBM and Microsoft are the two XML leaders in industry

Competitive Perspectives

- IBM XML technology based primarily on Java
  - XML and Java offer great portability and openness
- Microsoft technology not Java based
  - Linking XML technology with their product solutions

The difference is WebSphere.

Summary

Summary - IBM and XML

- **IBM provides the most comprehensive XML solution in industry:**
  - **Some key products enabled for XML:**
    - **WebSphere** http://www-4.ibm.com/software/webservers/appserv
    - **MQSeries**

      http://www2.itworld.com/cma/ett_article_frame/0,2848,1_245,00.html
    - **DB2 UDB 7.1**

      http://www-4.ibm.com/software/data/db2/udb/pdfs/whatsnew.pdf
  - **Industry leader in quality XML development tools**

    JARS TOP 25%
    - **Why are they free from AlphaWorks? Leadership.**
      - **Tools are consistently ranked very high by users**
    - **Microsoft XML tools in SDK include an xml parser, xsl parser, and code examples**
      - **non-productive tools**

The difference is WebSphere.

Summary

- XML is changing How companies manage/exchange data
  - In an open and portable manner
- B2B data exchange will be a major factor in driving XML
- The ultimate success of XML outside of high-end applications may depend on easy-to-use tooling:
  - "XML is today where PostScript was before Adobe Illustrator"
  - http://www.zdnet.com/zdnn/stories/comment/0,5859,2594206,00.html
- IBM is positioned to be the leader in XML technology

For More Information...

For More Information...

- **http://www.ibm.com/xml**
  - **Links to all things XML**
- **http://pdbeam.uwaterloo.ca**
  - **Several XML Tutorials**
- **http://www.javaworld.com**
  - **Several articles on programming XML in Java**
- **http://www.alphaworks.ibm.com**
  - **Many XML Tools**
- **XML and Java - Developing Web Applications**
  - **Hiroshi Maruyama, Kent Tamura, Naohiko Uramoto**