<XML>
<Introduction>Yes</Introduction>
</XML>

Dongwon Lee, Ph.D.
The Pennsylvania State University
dongwon@psu.edu

Overview

- Motivation of XML
- HTML vs. XML
- What is XML?
- XML Usages
XML

What does XML stand for (TOEFL question)?

1. X-rated Modular 3 Language
2. eXtensible Standard ML Programming Language
3. eXtensible UML (Unified Modeling Language) for the Web
4. eXtensible Meta Language for the Web
5. eXtensible Markup Language originating from SGML
6. None of the above

What Human Sees

About KU
Admissions
Academic Programs
Outreach Programs
Research
Campuses and Colleges
Visitors Guide
Alumni, Friends, and Giving …
What Machine Sees

Solution #1: Give Data Meaningful Tags
What Machine Sees from Solution #1

Better than before. Still NO clear and precise meanings of tags known to machines

XML

- XML is a framework for defining markup languages
- XML was designed to describe data, not format
  - Format is separately described by stylesheets
- XML separates syntax from semantics to provide a common framework for structuring information
- NO fixed collection of markup tags: One must define own tags, tailored for specific apps
- XML uses a schema language (eg, DTD, XML-Schema) to formally describe the data.
XML

- XML is NOT a replacement for HTML:
  - HTML should ideally be just another XML language
  - in fact, XHTML is just that
  - XHTML is a (very popular) XML language for hypertext markup

- HTML is about **displaying** information, but
  XML is about **describing** information.

HTML vs. XML

```
<center>
<h1>SIGMOD</h1>
<p><b><u>ACM</u> <a href="sigmod14.org">SIGMOD Conference</a>, Madison, WI, 2014</p>
</center>
```

```
<event eID="sigmod14">
<acronym>SIGMOD</acronym>
<society>ACM</society>
<url>www.sigmod14.org</url>
<loc>
<city>Madison</city>
<state>WI</state>
</loc>
<year>2014</year>
</event>
```
HTML vs. XML

- Need a stylesheet to define browser presentation semantics

![Diagram showing HTML vs. XML comparison]
XML 1.0 and 1.1

- Two current versions of XML

  - XML 1.0
    - Initially released in 1998
    - Latest release in 2008 (5th edition)
  - XML 1.1
    - Initially released in 2004
    - Latest release in 2006 (2nd edition)

SGML → XML

- Widely implemented & supported as of 2011

- SGML
  - DocBook
  - HTML
  - XHTML

- XML
  - SVG
  - MathML

A Conceptual View of XML

- An XML document is an **ordered, labeled tree**
- **Character data** leaf nodes contain the actual data (text strings)
- **Elements** nodes are each labeled with
  - a name (often called the element **type**), and
  - a set of **attributes**, each consisting of a name and a value,
  - and can have child nodes

Start, End, Empty Element Tags

- Element Tags can be 3 types:
  - Start: `<foo>`
    - Eg, “<html> …”
  - End: `</foo>`
    - Eg, “… </html>”
  - Empty: `<foo />`
    - Eg, `<img src="foo.jpg" />`
A Concrete View of XML

- An XML document is a text with markup tags and other meta-information.
- Markup tags denote elements:

```xml
..<foo attr="val" ...>bar</foo>...
    |    |    |    |    |    a matching element end tag
    |    |    |    |    |    the contents “bar” of the element
    |    |    |    |    |    an attribute with name attr and value val, enclosed by ’ or "
    |    | an element start tag with name foo
```

Example of XML Document

```xml
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don’t forget me this weekend!</body>
</note>
```

- XML declaration: the FIRST line of XML documents
- XML declaration consists of:
  - The version number
    ```xml
    <?xml version="1.0"?>
    ```
  - The encoding declaration
    ```xml
    <?xml version="1.0" encoding="UTF-8"?>
    ```
Example of XML Document

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don’t forget me this weekend!</body>
</note>
```

- Root element is `<note>`
- In XML all elements must have a closing tag:
  - `<p>foo</p>bar` → `<p>foo</p><p>bar</p>`
- XML tags are case sensitive:
  - `<Message>This is incorrect</message>`
- Must be properly nested within each other:
  - `<b><i>This is incorrect</i></b>`

Well-Formed XML

- Well-Formed: a document conforms to XML syntax rules such as:
  - Begin with XML declaration
  - One unique root
  - Case-sensitive
  - Matching Start / End tags
  - Properly nested
- Well-formed documents can still contain semantic errors or inconsistencies
  - Need VALID documents according to schema
Element vs. Attribute

- The same information can be captured by either Element or Attribute in XML

```xml
<event eID="sigmod14">
    <acronym>SIGMOD</acronym>
    <society>ACM</society>
    <url>www.sigmod14.org</url>
    <loc>
        <city>Madison</city>
        <state>WI</state>
    </loc>
    <year>2014</year>
</event>
```

```xml
<event eID="sigmod14" acronym="SIGMOD" society="ACM" url="www.sigmod14.org" city="Madison" state="WI" year="2014"/>
```
When to use Element vs. Attribute?

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Need order</td>
<td>Need more concise notations</td>
</tr>
<tr>
<td>Nested sub-structure</td>
<td>Want to provide default value</td>
</tr>
<tr>
<td>&lt;a&gt;&lt;b&gt;&lt;c/&gt;&lt;/b&gt;&lt;/a&gt;</td>
<td>Need to refer to other data piece</td>
</tr>
<tr>
<td></td>
<td>Eg, ID/IDREF in DTD</td>
</tr>
</tbody>
</table>

XML Schema: Motivation

- The company “Nittany Vacations, LLC” wants to standardize all internal documents using the XML-based format → \(nvML\)
- Gather requirements from all employees
  - Informal description (ie, narrative) of how \(nvML\) should look like
- Q
  - How to describe formally and unambiguously?
  - How to validate an \(nvML\) document?
Motivation: Schema = Rules

- XML Schemas is all about expressing rules:
  - Rules about what data is (not) allowed
  - Rules about how the data must be organized
  - Rules about the relationships between data

Motivation: sep-9.xml

```xml
<Vacation date="3000-09-09" guide-by="Lee">
  <Trip segment="1" mode="air">
    <Transportation>airplane</Transportation>
  </Trip>
  <Trip segment="2" mode="water">
    <Transportation>boat</Transportation>
  </Trip>
  <Trip segment="3" mode="ground">
    <Transportation>car</Transportation>
  </Trip>
</Vacation>
```

Example modified from Roger L. Costello’s slides @ xfront.com
Motivation: Validate

```
<Vacation date="3000-09-09" guide-by="Lee"> 
  <Segment id="1" mode="air"> 
    <Transportation>airplane</Transportation> 
  </Segment> 
  <Segment id="2" mode="water"> 
    <Transportation>boat</Transportation> 
  </Segment> 
  <Segment id="3" mode="ground"> 
    <Transportation>car</Transportation> 
  </Segment> 
</Vacation>
```

Validate the XML document against the XML Schema

```
xml.dtd or xml.xsd
```

XML Schema = RULES

Rule 1: A vacation has segments.
Rule 2: Each segment is uniquely identified.
Rule 3: There are three modes of transportation: air, water, ground.
Rule 4: Each segment has a mode of transportation.
Rule 5: Each segment must identify the specific mode used.

Schema Languages

- Schema: a formal description of structures / constraints
  - Eg, relational schema describes tables, attributes, keys, ...
- Schema Language: a formal language to describe schemas
  - Eg, SQL DDL for relational model

```
CREATE TABLE employees 
( 
id INTEGER PRIMARY KEY, 
first_name CHAR(50) NULL, 
last_name CHAR(75) NOT NULL, 
dateofbirth DATE NULL 
); 
```

- Why bother formalizing the syntax with a schema?
  - A formal definition provides a **precise but human-readable** reference
  - Schema processing can be done with **existing implementations**
  - **One’s own tools for own language can benefit:** by piping input documents through a schema processor, one can assume that the input is valid and defaults have been inserted

Schema Processing

![Diagram](http://www.brics.dk/~amoeller/XML/schemas/schemas.html)
Which Schema Language?

- Many proposals competing for acceptance
  - W3C Proposals: DTD, XML Data, DCD, DDML, SOX, XML-Schema, ...
  - Non-W3C Proposals: Assertion Grammars, Schematron, DSD, TREX, RELAX, XDuce, RELAX-NG, ...

- Different applications have different needs from a schema language

Family Tree of Schema Languages for Markup Languages

By Rick Jelliffe

1986
1992
1996
1998
2000
2002
2006
DTD: Attributes

- Attribute defaults:
  - #REQUIRED: the attribute must be explicitly provided
  - #IMPLIED: attribute is optional, no default provided
  - "value": if not explicitly provided, this value inserted by default
  - #FIXED "value": as above, but only this value is allowed

DTD: Attributes

Eg: “Name” element consists of an
- optional FirstName, followed by
- mandatory LastName attributes, where
  - Both are text string

<!ELEMENT Name (EMPTY)>
<!ATTLIST Name FirstName CDATA #IMPLIED
LastName CDATA #REQUIRED>
DTD: Attributes

- ID vs. IDREF/IDREFS
  - ID: document-wide unique ID (like key in DB)
  - IDREF: referring attribute (like foreign key in DB)

```xml
<!ELEMENT employee (…)>  
<!ATTLIST employee eID ID #REQUIRED  
  boss IDREF #IMPLIED>

...  
<employee eID="a">…</employee>…
<employee eID="b" boss="a">…</employee>
```

note.dtd

```xml
<!ELEMENT note (to, from, heading, body)>  
<!ELEMENT to (#PCDATA)>  
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```
**note.xml: Template**

```xml
<?xml version="1.0"?>
// Reference to schema goes here
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

**Exercise: event.dtd**

```xml
// event.dtd
<!ELEMENT event (acronym, society*, url?, loc, year)>
<!ATTLIST event eID ID #REQUIRED>
<!ELEMENT acronym (#PCDATA)>
<!ELEMENT society (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT loc (city, state)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT year (#PCDATA)>
```
Exercise: event.xml

XML document that conforms to “event.dtd”

```xml
<?xml version="1.0"?>
<!DOCTYPE event SYSTEM "../../dir/event.dtd"
<event eID="sigmod14">
  <acronym>SIGMOD</acronym>
  <society>ACM</society>
  <url>www.sigmod14.org</url>
  <loc>
    <city>Madison</city> <state>WI</state>
  </loc>
  <year>2014</year>
</event>
```

XML Editor Examples

- SyncroSoft <oXygen/>
  - 30 day free license, both Mac and Windows
  - http://www.oxygenxml.com/xml_editor/

- Altova XML Spy
  - 30 day free license, both Mac and Windows
  - http://www.altova.com/xml-editor/

- XMLPad
  - Free license, Only Windows version
  - http://www.wmhelp.com/xmlpad3.htm or
  - http://download.cnet.com/XmlPad/3000-7241_4-10252051.html

- XMLGrid
  - Online XML editor: http://xmlgrid.net/
Applications of XML

- XML is a meta-language to create another languages; the main application of XML is making new languages
- XHTML: W3C's XMLization of HTML 4.0.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head><title>Hello world!</title></head>
<body><p>foobar</p></body>
</html>
```

MathML: Mathematical Markup Language

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<msup>
<mi>x</mi>
<mn>2</mn>
</msup>
<mo>&#x2062;</mo>
<mi>b</mi>
<mo>+</mo>
<mi>c</mi>
</math>
```

From: http://en.wikipedia.org/wiki/MathML
CML: Chemical Markup Language

```xml
<molecule id="METHANOL">
  <atomArray>
    <stringArray builtin="elementType">C O H H H H</stringArray>
    <floatArray builtin="x3" units="pm">-0.748 0.558 -1.293 -1.263 -0.699 0.716</floatArray>
  </atomArray>
</molecule>
```

RSS: Really Simple Syndication

The XML Source view of IST’s RSS
RSS: Really Simple Syndication

XBEL: XML Bookmark Exchange Language

- XML language to express bookmarks (ie, favorites) in common format
- Several web browsers store their bookmarks in XBEL format
  - Eg, Galeon, Konqueror, Arora
- Apps use XBEL format
  - Eg, XBELicious
Weather Service in XML

XML Feeds of Current Weather Conditions

```xml
<Observation version="1.0" encoding="ISO-8859-1" re>=
  <current_observation>
    <credit>NWS's National Weather Service</credit>
    <credit_URL>http://weather.gov</credit_URL>
    <image><url>http://weather.gov/images/xml_logo.gif</url>
    <title>NWS's National Weather Service</title>
    <link>http://weather.gov</link>
  </image>
  <Location>New York/JFK</Location>
  <Station_ID>NY_JFK</Station_ID>
  <Latitude>40.66</Latitude>
  <Longitude>-73.78</Longitude>
  <Observation_time_rfc822>Mon, 11 Feb 2008 06:51:00 -0500</Observation_time_rfc822>
</current_observation>
```

XML Usages in S/W

- Open Office XML
- OpenDocument
- Apple's iWork
Why is XML Important?

- Technically, ... little initially; Just old simple tree model...
- Non-technically, ...
  - Hot ($$$)
  - The standard for representation of Web information
  - The real force of XML is **generic languages** and **tools**!
  - By building on XML, you get a massive (standard) infrastructure for free

XML Offers

- **Common extensions to the core XML specification**: a namespace mechanism, document inclusion, etc.
- **Schemas**: grammars to define classes of documents
- **Linking between documents**: a generalization of HTML anchors and links
- **Transformation**: conversion from one document class to another
- **Querying**: extraction of information, generalizing relational databases
  - More…!
Exercise: Make Your First XML

TO-DO:
1. Add an Empty Element
2. Add an Attribute to <body>
3. Add a child element to <from>
4. Change <to> as an attribute of <note> element
References

- World-Wide Web Consortium:
  - www.w3c.org/xml/
- XML Cover Page: www.oasis-open.org/cover/
- XML Articles: www.xml.com
- Latest XML News: www.xmlhack.com
- XML Tutorial:
  - www.w3schools.com/xml/default.asp
  - www.brics.dk/~amoeller/XML
  - www.xml101.com/xml/default.asp
- XML WIKI:
  - en.wikibooks.org/wiki/XML