1 XML: Semantics for the WWW

1.1 Current trends of the WWW world

(90% or more of distributed computing usage today)

1. Who processes web documents?
   • Now: HTML is written for humans to read
   • Soon: XML is written for software to read

2. What kinds of things are kept separate:
   • Now: HTML keeps content and presentation separate
   • Soon: XML keeps content, presentation, and meaning (semantics) separate, and does not deal with presentation (XSL does…)

3. How is web searching done now?
   • Now: Crudely, by brute-force searches on text strings
   • Soon: Smartly and efficiently, by using the meaning of the data in a document.

4. What are elements in a web page?
   • Items from a predefined set of graphical items (images, text, …) in HTML
   • Items from an extensible set of data items, with CORBA IDL interfaces

5. What kind of distributed system service does the web resemble?
   • Now: A big file server.
   • Soon: A big object database.

6. How do we identify components on the web?
   • Now: Uniform Resource Locator (URL)
   • Soon (??): Uniform Resource Indicator (URI) (naming alert…; also URNs .. unsolved problems!!)
1.2 Pieces in the XML World

- XML Language
- DOM: Distributed Object Model
- XSL: XML-based style sheets
- XLinks and XPointers

1.3 What is XML?

A meta-language: a language for defining languages, in order to

- Standardize data exchanges involving complex documents
- Establish a framework for weaving in the meaning of the data

1.4 What does XML look like?

Example of information on a song being stored

HTML:

```html
<dt> Dock of the Bay
<dd> by Joe Smith and Pete Jones
<ul>
<li> Producer: Joe Smith
<li> Publisher: Motown Records
<li> Length: 6:20
<li> Written: 1968
<li> Artist: Otis Redding
</ul>
```

XML:

```xml
<SONG>
  <TITLE> Dock of the Bay </TITLE>
  <COMPOSER> Joe Smith </COMPOSER>
  <COMPOSER> Pete Jones </COMPOSER>
  <PRODUCER> Jacques Cousteau </PRODUCER>
  <PUBLISHER> Motown Records </PUBLISHER>
  <LENGTH> 6:20 </LENGTH>
  <YEAR> 1968 </YEAR>
  <ARTIST> Otis Redding </ARTIST>
</SONG>
```

So just what is wrong with the HTML?

- HTML robots cannot tell if `<dt>` or `<li>` refer to a song, a definition, or some random verbiage
- Humans can’t always read any meaning into tags, for anything other than presentation
1.5 Why are people getting excited about XML?

- Design of Domain-Specific Markup languages to trade notes, data, other information
- Self-describing data (and often also human-readable)
- Inter-application data exchange
- Top-10 list from the “Client/Server Survival Guide (3ed):
  1. Exchange data between clients and servers on the web
  2. Provide a common data exchange medium among the Web’s various data stores
  3. Provide common tags (ontologies, or data vocabularies) for different industries and domains
  4. Server as the electronic data interchange (EDI) language for web commerce
  5. Server as a packaging technology for Web components (e.g., CORBA Beans using XML)
  6. Enable web bots, crawlers, and agents to act more intelligently in their search for information
  7. Serve as the lingua franca for web-based workflow
  8. Provide a common mechanism for objects to exchange state information
  9. Provide common data vocabularies for component suites
  10. Provide a channel definition format for push technology (Microsoft’s CDF uses XML…)

1.6 What is the Document Object Model (DOM)?

Problems:
- How do we access the contents and rich structure of an XML document?
- What is the data manipulation language for XML documents?
- What is for XML documents as SQL is to relational databases?

Answer: DOM

DOM is an object model which lets you manipulate and navigate a web document which contains objects.

E.g., one can
- change, delete, or add an element or its attributed on a document.
- Query for the list of all elements (the tags, not the contents) such as <li>
- Find the contents of a given element

1.7 What is XSL?

A rich way to view an XML document (remember, XML does not deal with presentation issues)
1.8 What are XPointers and XLinks?

HTML Limitations:

- URLs point to a single document
- Tags point to a fixed point, not to “8th word in the 6th paragraph”
- Links are one way: not sense of history or relationship between documents

XLinks will help by

- Letting any element become a link, not just a “<a “ element
- Be bidirectional or multidirectional or pointing to multiple replicas (so you can choose the closest)

XPointers will help by

- Letting links point to a location calculated like “3rd paragraph (or other element) after the second <SONG> element”
- Can point not just to a single point in a document, but to a range/span

1.9 Bottom Line

Quote: “XML will do for data portability what Java has done for application portability”. (can’t find source.)

Q: So, will XML save the planet?
A: Possibly, but only if its main threat is data unportability.

Q: Is XML the answer, then?
A: Possibly, but only if your question only involves data portability (not transport or architecture or …)

For more info, see the OMG’s XLM page, http://www.omg.org/xml/ and also the world wide web consortium, http://www.w3.org/.