XML

eXtensible Markup Language

A Basic Overview

XML

What is it?

What is it not?

What is it good for?

A short blurb

XML is a simple data format
that balances the needs of
people to read/write data
with the needs of machines
to read/write data.

-- Dan Connolly, W3C

The XML specification

- Describes a class of data objects called XML documents
- Partially describes the behavior of computer programs which process them

Character data and markup

- XML data is made up of characters: some form *character data*, others form *markup*
- Markup represents the document's storage layout and logical structure
- XML keeps markup logically separate from data but physically associated with it

XML documents

- XML documents are made up of storage units called entities (files, pages, etc.)
- XML provides a mechanism to constrain the storage layout and logical structure of entities

XML processors

- XML processors are used to read XML documents and provide access to their content and structure
- XML processors do their work on behalf of applications
- Many simple XML processors are already freely available

The official goals of XML

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.

The official goals of XML

5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.

The official goals of XML

8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

Browsing XML

- XML has lots of applications besides browsable text
- XML browsers (unlike HTML browsers) don't have hard-coded knowledge about how to display tags
- Stylesheets (**not yet standardized**) will solve this problem

Time for some sample XML!

```
<CHECK>
    <ADDRESS>
        John W. Cowan
        Gale L. Cowan
        123 E. 345th St, #2A
        New York, NY  10111
    </ADDRESS>
    <CHECKNO>1013</CHECKNO>
    <DATE>19980615</DATE>
    <PAYEE>Consolidated Edison</PAYEE>
    <AMOUNT>USD 65.75</AMOUNT>
```

Time for some sample XML!

```
<BANK>
        The Chase Manhattan Bank
        270 Park Avenue
        New York, NY  10017
</BANK>
<MICR>
        <ROUTE>021000128</ROUTE>
        <ACCOUNT>077442422</ACCOUNT>
</MICR>
<SIG TYPE="MD5">
        0921937AB903C93D8932F39D9F9EC123
</SIG>
</CHECK>
```

Notes on the check

- The account number, address, and digital signature are fakes!
- Every start-tag has an end-tag (unlike HTML)
- Everything between a start-tag and the matching end-tag, inclusive, is called an *element.*

The content of elements

- The CHECK and MICR elements contain other elements
- The other elements contain just text
- It is possible for elements to contain both other elements and text (called *mixed content*).

Attributes

- Elements can have *attributes*
- Attributes have *values*
- Attributes and values are written within the start-tag
- The SIG element has attribute TYPE, and value “MD5”.

XML is self-documenting

Compare this version of the check data:

```
021000128000077442422000101398061565 75CONSOLIDATED EDISON
```

It contains the same information (except for the digital signature), but who can decipher it? Only some program somewhere.

XML is self-documenting

When the source to that program is lost, or the data dictionary is replaced by a new system, the rules for interpreting this record become a black art.

The XML version requires more characters, but it is plain text that will remain readable over the years.

Plain text

- Proprietary formats for text and data come and go.
- The handwritten 1860 U.S. census data is more accessible than the 1960 data on IBM punched cards.
- Plain text doesn't depend on specific programs to create, edit, or interpret it.

XML is SGML--

- XML is a restricted form of SGML, the Standard Generalized Markup Language (ISO 8879)
- XML documents are conforming SGML documents
- XML offers most of SGML's power while avoiding most of its complexity

XML is not HTML++

- HTML has a fixed set of tags for use in display of rich text with hyperlinks and images
- XML has no built-in tags, but allows designers to specify their own tag set for use in structuring data
- XML is case-sensitive everywhere

XML is not HTML++

- XML does not make HTML obsolete
- The next HTML definition will probably be compatible with XML
- Using freely available software, conforming HTML files can be conformed to the XML specification
- Lots of HTML out there is unfortunately non-conformant

Another XML example

<HTML><HEAD><TITLE>Sample
Document</TITLE></HEAD>

<BODY><H1>This is a sample document</H1>

<P>Here's a paragraph of plain text. It is very boring, both visually and in terms of its content. But what do you expect from a sample?</P>

<HR />

<P>Here's a paragraph which contains <EM>emphasized text</EM>,

<STRONG>strongly emphasized text</STRONG>, and <SAMP>sample text</SAMP>. The content is still boring.</P>

</BODY></HTML>

Netscape Navigator 4.0

This is a sample document

Here's a paragraph of plain text. It is very boring, both visually and in terms of its content. But what do you expect from a sample?

Here's a paragraph which contains *emphasized text*, **strongly emphasized text**, and `sample text`. The content is still boring.

Internet Explorer 4.0

This is a sample document

Here's a paragraph of plain text. It is very boring, both visually and in terms of its content. But what do you expect from a sample?

Here's a paragraph which contains *emphasized text*, **strongly emphasized text**, and `sample text.` The content is still boring.

A few little differences

- All the <P> tags are matched with </P> tags, which XML requires and HTML permits
- The <HR> tag looks funny: <HR />, which will upset current HTML validators but not most browsers

Empty elements

- An *empty element* is one which has no content
- The HTML tags BR, HR, META, etc. are empty
- Because XML tags are invented by users, XML has to be told explicitly which ones are empty

Empty elements

- All empty elements end with “/>“ instead of just “>“ in XML
- Therefore, “<HR />“ is a valid empty XML tag
- The space before “/” is a requirement of current browsers, not of XML

Document Type Definitions (DTDs)

- Describe the valid elements and what kind of content they have
- Describe the valid attributes and provide default values
- Increase self-documentation
- Not required for well-formed XML documents

Part Of The Check DTD

```
<!-- The DTD for an ordinary bank check -->
<!-- Uses '<!DOCTYPE CHECK SYSTEM "check.dtd">' -->
<!ELEMENT ACCOUNT #PCDATA>
<!ELEMENT ADDRESS #PCDATA>
<!ELEMENT AMOUNT #PCDATA>
<!ELEMENT CHECKNO #PCDATA>
...
<!ELEMENT MICR (ROUTE, ACCOUNT)>
<!ELEMENT CHECK (ADDRESS, CHECKNO, DATE, PAYEE,
                 AMOUNT, BANK, MICR, SIG, MEMO?)>
<!ATTLIST SIG
          TYPE CDATA #REQUIRED>
```

Attaching the DTD to Checks

```
<!DOCTYPE CHECK SYSTEM “check.dtd”>
<CHECK>
   <ADDRESS>
...
...
</CHECK>
```

Internal DTDs

- Simple DTDs can be placed inside the DOCTYPE declaration
- Internal declarations override external ones
- Allows customization of specific documents

XML and Unicode

- Any Unicode character can be incorporated in XML documents
- 38,887 characters from 25 writing systems (1100 languages) plus many symbols
- Random characters can be included even in ASCII text with &#xnnnn;

Is XML Stable?

- Yes!
- Version 1.0 became a Recommendation in February 1988
- No further changes are expected
- Higher-level protocols layered on XML are not completely stable yet
- XML is ready for use

XML Links

- **Not yet (May 1998) part of the standard**
- Simple links are like HTML links, but any element can be a link if it has the proper attributes
- Extended links are multi-valued, multi-directional, and out-of-band

XPointers

- **Not yet (May 1998) part of the standard**
- XPointers can be IDs like “ITEM39”, or references like:
  root().child(5, DIV).child(4, P)
  meaning “the 4th paragraph of the 5th division of the text”

XML Applications

- Microsoft Channel Definition Format (for push technology)
- W3C MATHML (Mathematics Markup Language, for displaying equations)
- Digitome Notes Flat File format (can be exported/imported from Domino databases)

XML Applications

- W3C Synchronized Multimedia Integration Language (laying out complex multimedia objects)
- XML/EDI (electronic business transactions between companies)
- Many, many more applications are in the works

Available XML Parsers

- Expat by James Clark, C++, non-validating
- Microsoft, C++ (non-validating) and Java (validating)
- Many others, mostly free for general use
- Perl, Tcl, and Python scripting languages have parsers too!

Common Interfaces

- SAX/Simple API for XML: very simple, event driven (start of element, text, end of element, etc.)
- DOM/Document Object Model: tree-based, not fully defined yet, but will provide complete access to XML
- Most parsers comply with one or both

Good introductory articles

http://www.sil.org/sgml/xmlIntro.html

And much, much more...

The XML home page is:

http://www.w3.org/XML/

Essentially everything else is reachable from there