Outline

- Definition (brief)
- Genesis
- More detail; examples
- Processing XML
- Related specs
- XML languages
- References

XML -- What is it?

- E**X**tensible **M**ark-up **L**anguage, but really a meta-language to be used to define mark-up languages
- The mark-up is used to make structural elelments of the text explicit. A typical and early example is *editorial mark-up*, which indicates things like start of paragraph, italics, etc.

History

- In the beginning (c. 1970) there was GML, **G**eneralized **M**ark-up **L**anguage, which soon (c. 1980) morphed to SGML (**S**tandard…)
- The earliest driving requirements were
  - typesetting
  - semantic structure, useful for queries

History (cont'd)

- Like XML, SGML is a meta-language used to define mark-up languages with a DTD (**D**ocument **T**ype **D**efinition). HTML is an SGML language. See http://www.w3.org/TR/REC-html40/sgml/dtd.html
- XML is a recommendation of W3C (**W**orld **W**ide **W**eb **C**onsortium)

Mark-up Language Example

```
<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title> Nearly empty example </title>
  </head>
  <body> <H1 align = "center">The one and only heading </H1>
    <p>Some <strong>content.</strong></p>
    <hr>
    <p> The rest of the content, including
          a <a href=http://www.slac.stanford.edu/~jrb> link. </a>
    </p>
  </body>  </html>
```

Why XML?

- HTML is primarily links , low-level structural information and some presentation; at best limited semantic information may be inferred.
- HTML is not easily extensible.
- SGML is complex, not suited to distributed applications.
- XML is SGML with restrictions (any XML language is an SGML language). Pre-existing SGML tools may be used.

What's What

$$HTML \in \{SGML\ languages\}$$

$$\cup$$

$$XHTML \in \{XML\ languages\}$$

..where HTML, XHTML define sets themselves:

$$myDoc.html \in \{HTML\ docs\}$$

XHTML is functionally equivalent to HTML but follows the stricter syntax of an XML language

XML Goals (from spec)

The design goals for XML are:

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.
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.
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.

XML Goals (my summary)

- Aimed at internet usage
- Must be easily extensible
- Compatible with SGML
- Must be feasible to implement tools in finite time
- Human-readability, ease of computer processing more important than brevity

XML Features

- Syntax of XML is simpler to parse than SGML. Can check for *well-formedness** without a DTD.
- Unlike HTML, readily extensible since there is no mandated set of tags.

* A form of low-level syntax checking: all the tags nest properly, attributes are quoted, etc.

XML Features (cont'd)

- Separate presentation from content.
  - Good because these things really are logically separate; in particular, considerations of presentation should not interfere with laying out semantic structure.
  - Downside is that XML documents can't be rendered except in a very simple generic way without further specification.

XML as Database

Although it is ***not*** among the explicit goals of XML to support database-like applications, in fact, with its extensibility and emphasis on separating presentation from content, XML can be a suitable tool for this, if the data maps well to a predominantly hierarchical representation. (Not surprising since SGML has traditionally been used this way.)

Such applications may “not care” about the webbable nature of XML (but it may be useful after the fact anyway) .

```
<lcdparm>
   <global file="largeParms2.xml" />
   <physical_detector topology="large" id="EM_L2" >
   <volume id="EM_BARREL" >
     <tube>
        <barrel_dimensions inner_r = "196.0" outer_z = "322.0" />
        <layering n="40">
          <slice material = "Pb" width = "0.4" />
          <slice material = "Tyvek" width = "0.05" />
          <slice material = "Polystyrene" width = "0.1" sensitive = "yes" />
         </layering>
         <segmentation theta = "300" phi = "300" />
     </tube>
     <calorimeter type = "em" />
  </volume>
  ...
```

Start subdetector description

function

End subdectector description

What is XML (details)

- Well-formed XML can be parsed without a DTD.
- *With* a DTD, an XML document may be checked for **validity:** does the document conform to application-specific constraints?
- A DTD is a separate file (**entity** is a better term), but DTD-like information can also be kept in the **internal subset**.

What's in a DTD?

- Defines which **elements** (the things in < >) may appear in the document
- Allowable **attributes** for each element type; may specify defaults
- **Element content;** that is, what goes between the start tag and the end tag: more elements, text, both, or neither

```
<!-- Mark-up language for lcd detector description, possibly other
    parms.
  -->

  <!ELEMENT lcdparm (global, physical_detector, proc_parm?)>
  <!ELEMENT global EMPTY>
  <!ATTLIST  global file NMTOKEN #REQUIRED>
  <!ELEMENT physical_detector (volume | complex_volume)+ >
  <!ATTLIST physical_detector topology (large | small)  #REQUIRED
                    id      ID              #IMPLIED>

   <!ELEMENT volume ((tube | disk |  cone),
                (calorimeter | tracker | coil | cryo | support)?) >
  <!ATTLIST volume  id ID #REQUIRED
                inter_len_cm NMTOKEN #IMPLIED
                rad_len_cm NMTOKEN #IMPLIED>
```

comment

element declaration

content model

attributes declarations

More XML Features

- **Parsed entities** - can be used like #define or even #include
- **Character entities -** express a character not available in literal form, e.g. mostly have to use < to get <
- Processing instructions
- Internationalization support (*not* in SGML)

Processing XML

- You need a parser. Two interface styles:
  - SAX. Stream oriented, event-driven. Application signs up handlers to be called back when, e.g., a new element start tag is encountered.
  - DOM. (**D**ocument **O**bject **M**odel) stores XML document as hierarchy of objects, all available at once in memory.

Processing XML (cont'd)

Another way to process XML is to transform it into some form more suitable for display or for consumption by some other program.

Most typical use of this kind is to transform XML into HTML or XHTML so that browsers can display it.

XSL (E**X**tensible **S**tylesheet **L**anguage) can be used to specify how the transformation and formatting should be done. One writes an XSL stylesheet and submits it along with an XML document to an *XSL stylesheet processor*.

XSL processing may involve sorting, filtering, indexing,...

Related W3C Specs

- **DOM** (rec, cand. rec), **XSLT** (rec), **XSL** (wd last call)
- **XML Namespaces**: avoid collisions (rec)
- **XLink**: much-generalized (relative to HTML anchors) document linking capability (wd last call)
- **XPath**: address parts ({nodes}) of XML doc (rec)
- **XPointer**: layered on XPath; extensions for (wd)
- **XML schema**: DTD replacement (wd last call)

More XML Related Specs

- **XInclude**: sort of like #include (wd)
- **XBase**: specify URI (e.g. file or directory) used to resolve relative URI (wd last call)

Of course XML itself also has a spec (rec) but SAX doesn't!

NB: W3C doesn't write standards. The strongest term they use is ***recommendation***.

Sample XML Languages (W3C)

- MathML: Defines two categories of mark-up: ***presentation*** and ***content***. Products already exist to render it.
- SVG (Scalable Vector Graphics): “a language for describing 2-dimensional graphics in XML”
- SMIL (Synchronized Multimedia Integration Language)
- XHTML
- XML Schemas

References

- See Robin Cover's website to find anything else
  - http://www.oasis-open.org/cover/xml.html
- W3C reports and recommendations
  - http://www.w3.org/TR/
- XML Annotated Spec
  - http://www.xml.com/axml/axml.html
- More XML languages
  - http://xml.org/xmlorg_registry/index.shtml
- Open source tools
  - http://xml.apache.org/
- Mailing lists, newsgroups (but can take a lot of time)