LIS565 Lecture 8

Thomas Krichel

http://openlib.org/home/krichel

2001–11–08

Reading

Young, Michael J, (2000) "Step-by-step XML", Microsoft Press, 2000

Bray, Tim, (1998) "The Annotated XML Specification", http://www.xml.com/axml/testaxml.htm (advanced)

XML

Is a W3C recommendation. It is a new (1998) markup language that will transport a lot of contents over the Internet in the future.

As its level of complexity goes it sits in between HTML and SGML.

HTML, XML, SGML

XML is more flexible than HTML. It allows to define tags.

XML is less flexible than SGML. It does not allow to implement seldomly used features of SGML.

XML has not yet replaced HTML, but it may do so one day.

Design Goals

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.
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 document structure

prolog

root element

other stuff optional

XML document example

```
<?xml version="1.0"?>

<-- a file to contain sofix data-->

<sofix>

</sofix>
```

well-formed XML documents

- the document must have exactly one top level (root or document) element
- elements must be nested
- each element must have a start and an end tag
- element names are case-sensitive

another XML document example

```
<?xml version="1.0"?>

<-- break every rule in the book -->

<sofix>

<work>

<contributor> <name> <role>

</name> </role> </contrIbutor> </work>

</work>

</sofix><sofix></sofix>
```

Element names

Element names must begin with a letter or underscore. They may be followed by zero or more letters, digits, periods, hyphens or underscores.

The use of the name string "xml" at the beginning of an element name, in any capitalization is reserved for future standards.

Do not use the colon in element names.

Element contents

Character data in elements can contain any data except < (use `&#60;`), & (use `&#38;`), and]]> (don't use it).

An element may also contain

- Entity references (discussed in another lecture)
- Character references (discussed in another lecture)
- CDATA sections

- Processing instructions
- Comments

Comments

Start with <-- and end with -->

Example:

```
<-- I can put < and & in a comment -->
```

Comments can be placed any where except in the markup, i.e. the stuff that is between < and >.

Processing instructions

Provide instructions to XML parsers. General form

<? *target instruction* ?>

Example

```
<?xml-stylesheet type="text/css" href="sofix.css" ?>
```

Can be placed anywhere where comments can be placed.

CDATA section

`[CDATA[` opens a CDATA section.

`]]>` closes a CDATA section.

In the CDATA section you can have any type of character data. For example, you can include a HTML page. Can be placed anywhere where comments can be placed.

CDATA sections do not nest.

Element attributes

General form

<*element_name attribute_name*="*attribute_value*">

or

<*element_name attribute_name*='*attribute_value*'>

Attribute names are subject to the same restrictions as element names.

Value may not contain its delimiter or <. It may contain & only if this is part of an entity reference (to be discussed in another lecture).

Well-formed and valid documents

All restrictions that we have discussed apply to every XML document in order for it to be well-formed.

In addition to being well-formed, an XML document may also be valid. That means that they obey to further constraint set out in a Document Type Declaration or in an XML schema.

Example applications of XML

write a one-page description of the following XML application and present in class.

Bioinformatic Sequence Markup Language
Channel Definition Formant CDF
Chemical Markup Language CML
Extensible Forms Description Language XFDL
Mathematical Markup Language MathML
Open Financial Exchange OFX
Open Software Description OSD
Synchronized Multimedia Integration Language SMIL
Vector Markup Language VML

The sofix CD description format

This a format for CD cataloging using XML.

Format called "sofix".

Thomas and students have made it all up.

Thomas named it after his former lover Sophie C. Rigny.

Conceptual framework

Three key concepts

- Item: an individual CD or a collection of CDs kept physically together (i.e. sold together)
- Work: a piece of music as recorded on a CD. For simplicity, we do not distinguish between composition and recording of that composition.
- Track: semantics associated with physical separation on disk

Generalities

- All titles in English.
- If no English title provided, use a translation if it is obvious.
- If the translation is not obvious, use original language.
- All personal names as *Lastname*, *Initials*.

XML implies a nested structure

```
<item>
 <work>
   <track>
   </track>
   ...
 </work>
 ...
</item>
```

attibutes of item

`<labelname>`*name of label* `</labelname>`

`<number>`*number of the CD* `</number>`

attibutes of work

<work>

<compositionyear>*take last year if there are several, including several possible years* </compositionyear>

<recordingyear>*take last year if there are several* </recordingyear>

<title>*take full title including opus number and key* </title>

<contributor role="*role*"> *name of person or group with a role in work/performance* </contributor>

controlled vocabulary of roles

composer, conductor

chamber_orchestra, orchestra, piano_trio, string orchestra, string quartett

alto, bariton, bass, soprano, speaker choir

alto sax, bassoon, chello, clarinett, flute, french_horn, horn, oboe, organ piano, prepared_piano, recorder, viola, violin, xylophone

The complete listing of all possible values is held at

http://wotan.liu.edu/home/krichel/itr8/sofix_roles

attributes of a track

<track>

<tracktitle>*full title as given on CD* </tracktitle>

<time>*minutes*:*seconds* </time>

</track>

</work>

</item>