Beyond XML: Limitations of XML 1.0

Simon Mahony
From an original document by Susan Hockey

This document is part of a collection of presentations and exercises on XML. For full details of this and the rest of the collection see the cover sheet at:
http://ucloer.eprints-hosting.org/id/eprint/19

XSLT 2.0

- Became w3c recommendation in January 2007
- Works with XPath 2.0
- Some improvements and variations on the existing spec (barely noticeable to the non-geek)
- New elements and functions

Grouping

- Allows elements to be grouped based on common values
  - element or attribute
- Can be based on place in a sequence
  - (eg paragraphs 15-19)
- Fixed size groups
  - Process a group with 3 adjacent nodes
- An item can be in more than one group

Allows Creation of Multiple Output

- Possible to output part of document in a different form using <xsl:result-document>

- Might want to output some as CSS, XHTML or XML

- Set up output formats

Use <xsl:output> at beginning of document

- Specify output file names

- Example from w3c allows XHTML to be validated in strict and transitional

Allows multiple inputs

- Use range of separate XML documents
- Treats as one large single document
- Create a 'master' XML document ref the others

Problems

- Tutorials are still sadly lacking online
- No w3schools tutorial yet
- Specialist textbooks and reading lists required
- Plenty for version 1.0

- For more see:
- W3C XSL Transformations (XSLT) Version 2.0
  http://www.w3.org/TR/xslt20/

Problems with XML

Overlapping hierarchies

- Lines of poetry vs syntax
- Paragraphs vs page breaks
- Overlapping speeches
- Etc etc etc
- Google this for much online discussion

Solutions or ways around?

- In-house specific tools
- Neo-Platonic view of the text
- (see Jerome McGann: *Radiant Textuality*)
- Ontologies: a possible answer
- (a series of relationships)
- XML organises items hierarchically
- Ontologies describe the nature of the relationship

- RDF: ideas about semantic web
- linked data allows overlapping

Ontologies

•A network of relationships

A   -   B   -   C

If A has a relationship with B

And B has a relationship with C

Then A and C have an implicit relationship mediated by B

RDF OWL: Web Ontology Language

RDF- Resource Description Framework

- [W3C RDF](#)
- Metadata standard written in XML
- Used to describe resources but not usually shown on pages
- Important part of Semantic web efforts
- Can be used with Dublin Core elements
- [OWL](#) Web Ontology Language

RDF parts

- the thing being described
- the characteristic of interest
- the value of that characteristic

- subject / predicate / object
- correspond to a single (triple) statement

Contents

- A resource: identified using a URI (Uniform Resource Identifier)

   − "http://www.w3schools.com/RDF"

- Resources have properties

   − a name, eg "author" or "homepage"

- Properties have values

   − "Jan Egil Refsnes"

   − "http://www.w3schools.com"

   − (a property value can be another resource)

RDF Syntax

• <RDF> is the root element, which contains

• <description>

contents can be arranged using further elements like <bag> and <sequence>

```
<RDF>
  <Description about="http://www.w3schools.com/RDF">
    <author> Jan Egil Refsnes</author>
    <homepage> http://www.w3schools.com</homepage>
  </Description>
</RDF>
```

Google TouchGraph
http://www.touchgraph.com/TGGoogleBrowser.html

Linked Data

Connect Distributed Data Across the Web

http://linkeddata.org/

- Fragments of meaning
- Mixed models

Topic Maps

Standard for the representation and interchange of knowledge

ISO standard

XML syntax (XTM 2.0)

More ideas for the semanitc web

XML: Other standards and activities

XLink and XPointer

- Methods of linking and pointing
- XLink defines how one document links to another

- XPointer defines how individual parts of a document are addressed and is built on XPath

Limitations of HTML Linking

- Can only point to a single document
- Cannot link to places within a document unless you have write access to make anchors

- Linking is one-way within the HTML document

XLink Links

- Multi-directional
- Any element can be a link, not just <a>
- Links can be stored externally
- Specify role of a link
- Annotate read-only documents

Link Tags

- Almost any element
- Attribute *xlink:type* describes the type of link

<ltag xlink:type="simple" Link text </ltag>

Linking to a Web document as the target

- Attribute *xlink:href* identifies the target of the link

<ltag xlink:type="simple"
    xlink:href="http://www.ucl.ac.uk/infostudies">
    Text of link</ltag>

Descriptions on Links

- *xlink:role* describes the purpose of the link and is intended to be picked up by machine processors

<ltag xlink:role="footnote">text of link</ltag>

Link Behaviour

- *xlink:show* – how the content should be displayed when the link is activated, for example in a new window, or replacing the existing window

<ltag xlink:show="new" xlink:href=" …url …."> text of link</ltag>

Embedding Information

- xlink:show="embedded" embeds the target of the link into the current document
- Can traverse various links, and assemble targets of the links into one document

Specifying attributes

- Only link to elements with certain attributes

<horse breed="Thoroughbred" id="Irish">

#xpointer(id("Irish"))

- Link is

xlink:href="http://www.equine.com/horses.xml#xpointer(id('Irish'))"

- Or if you have an id attribute you can use shorthand version

xlink:href="http://www.equine.com/horses.xml#Irish"

xlink:actuate Attribute

- Determines when the link is traversed
- Values are
- *onRequest*: only when the user requests it
- *onLoad*: traverse the link when it is loaded
- *Undefined*: leaves it to the application

Extended Links

- Can point to more than one target, for example mirror sites
- Rather than listing them all as in HTML, the browser could choose the most appropriate one

Out-of-line Links

- Links can be stored externally to the document
- Collection of links is a *linkbase*
- Changes to *linkbase* do not affect documents
- A *linkbase* is an XML document

Uses of External Links

- Re-ordering documents
- Multiple annotations – can be identified by name of annotator etc
- Creating links to and from resources in formats that do not have native support for links – multimedia

Extended Pointers

- Link to portions of a document – identified by XPath expressions

- E.g. the last child of <x>
- The instance of <x> which contains the string "*yyyyy*"
- The instance of <x> within <y>

Document Object Model (DOM)

- Model of a document using object-oriented definitions
- Document consists of nodes with properties and methods associated with each node

W3C Description

"The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page."

http://www.w3.org/DOM/

W3C and DOM

- Application Programmer Interface (API) for working with XML documents

- Standard method of accessing and manipulating an XML document

- Based on the tree structure

XML and JavaScript

- Manipulate XML documents client-side
- Extract XML information by identifying tags
- Also now use of Java to manipulate XML

AJAX

- Asynchronous JavaScript and XML
- Updates data on a webpage without reloading the full page
- Allows interactive and dynamic interfaces on web
- eg Googlemail
- Greatly reduces bandwidth use and load times

Querying XML Documents

- XQuery

functional programming language designed to query

XML (DB type) collections

- Built on XPath expressions same data model
- Rather like SQL
- Better with database-like documents

RSS (web feeds)

- Now very common way to provide news feeds for web pages (eg L&S blog)

- But it's not actually XHTML, it's XML

- Different versions from 0.91 to 2.0

- Very simple to use

- Consists of <channel> and <item>

- <item> contains

  # <title> - title of the item

  # <link> - the hyperlink to the item

  # <description> short description of item

Developing and Managing an XML Project

- Tool requirements
  - Entering documents or converting existing documents
  - Management of documents
  - Delivery system – querying etc
  - Stylesheets for delivery

Developing and Managing an XML Project

- Tool requirements
- Used to be large monolithic programs such as Dynatext/Dynabase
- Now more likely built up from open source tools

Web Services

- New technologies to manage and deliver XML documents
- Define services to run on the web
- Definitions are themselves pieces of XML
- For example to combine bibliography items from catalogue, Amazon and Google