Document Relationships

Hypertext

- **XML is also tackling the need for hypertext**
  - two proposals being worked on
    - *XML Linking Language (XLL, XLink)*
      - a vocabulary allowing the creation of suites of documents
        - 'we can expect "content" to be created that consists solely of huge databases of links, with no content actually created by the link authors.'
    - *XML Pointer Language (XPointer)*
      - a flexible and powerful tool for addressing documents and document fragments
        - *without the need to have predefined structures and targets*
  - both attempt to solve different parts of the problem
  - development has been slow
    - *now slowly gathering pace after a long delay*
  - neither proposal is present in any 'real' piece of software
    - *although Mozilla has some support*
      - so may appear in some future Netscape
  - a lot of earlier work was done for SGML as 'HyTime'

Solving Problems

- **Both work towards solving some problems inherent in HTML's linking mechanism:**
  - HTML links are not self-descriptive
    - *you have no idea what a link will do until it is actuated*
  - HTML links waste bandwidth when only a portion of a target document is needed
    - *need to retrieve whole document and then filter it*
  - HTML links can only return a single target resource
    - *compare this with many 'help' applications that can present multiple targets*
  - HTML links are too closely coupled to the structure of the target document
    - *makes for a maintenance nightmare; lots of broken links*
    - *makes it difficult for groups of documents to be worked on in isolation*

XLink/XPointer

- **"A revolution in the way documents can be linked."**
  - *very* much working draft technologies
    - nowhere near *ready to meet the world yet...*
  - XLink and XPointer are separate (but related)
    - *provide for simple linking mechanisms, similar to HTML*
    - *also allows complex, multiple-destination links and document fragment addressing*
- **Sun claims a patent on XPointer technology**
  - but
    - *"as long as XPointer either is on the Recommendation track or has been adopted as a Recommendation…agrees it will grant royalty-free licenses, under reasonable terms and conditions, and on a non-discriminatory basis, under Sun's essential patent claims to make, use, sell, offer for sale, or import implementations of XPointer."*

XPointer

- **A tool for referencing portions of documents, built on the clean tree structures of XML documents**
  - extends XPath syntax
    - *allowing interdocument addressing*
  - can specify single or multiple locations in a target document
    - *points/ranges*

```
doc.xml#xpointer(/descendant::elem[@attr='value'])
doc.xml#xmlns(x=urn:thename:space)xmlns(y=urn:anothername:space)
  xpointer(//x:elem[@y:attr='value'])
doc.xml#xpointer(/*[1]/*[3]/*[1])
doc.xml#xpointer(range-inside(//element/child)
doc.xml#xpointer(id('elem1')/range-to(id(elem2)))
doc.xml#xpointer(string-range(//elem, 'searchterm')[3])
doc.xml#xpointer(here()/ancestor::chapter[1])
```

A Slideshow

```
<?xml version="1.0"?>
<SLIDESHOW>
  <SLIDE>
    <TITLE>Welcome to the slide show!</TITLE>
    <BUTTON xml:link="simple" href="origin().following(1,SLIDE)">
      Next
    </BUTTON>
  </SLIDE>
  <SLIDE>
    <TITLE>This is the second slide</TITLE>
    <BUTTON xml:link="simple" href="origin().preceding(1,SLIDE)">
      Previous
    </BUTTON>
    <BUTTON xml:link="simple" href="origin().following(1,SLIDE)">
      Next
    </BUTTON>
  </SLIDE>
  <SLIDE>
    <TITLE>This is the second slide</TITLE>
    <BUTTON xml:link="simple" href="origin().preceding(1,SLIDE)">
      Previous
    </BUTTON>
    <BUTTON xml:link="simple" href="origin().following(1,SLIDE)">
      Next
    </BUTTON>
  </SLIDE>
  ...
  <SLIDE>
    <TITLE>This is the last slide</TITLE>
    <BUTTON xml:link="simple" href="origin().preceding(1,SLIDE)">
      Previous
    </BUTTON>
  </SLIDE>
</SLIDESHOW>
```

May *not work with latest spec.*

From: "XML Bible" by Elliotte Rusty Harold

July 3, 2009

ZVON.org

Range function

For elements the container node is the parent of the element. For nth element, the index of the start-point of the resulting location is n-1, the index of the end-point of the resulting location is n.

XPointer: xpointer(range(//AAA/BBB[2]))

```
<AAA>
  <BBB bbb="111">
    Text in the first element BBB.</BBB>
<BBB bbb="222">
  Text in another element BBB.
  <DDD ddd="999">
    Text in more nested element.</DDD>
</BBB>
  <CCC ccc="123" xxx="321">
    Again some text in some element.</CCC>
</AAA>
```

- X marks point (zero width)
- This style marks the container node.
- This style marks resulting locations.
- X marks collapsed range (zero width)

Range-inside function

If the argument is not a range, than it is used as a container location of the start- and end-points of the resulting range. In this example the container is `//AAA/BBB[2]`.

XPointer: xpointer(range-inside(//AAA/BBB[2]))

```
<AAA>
  <BBB bbb="111">
    Text in the first element BBB.</BBB>
  <BBB bbb="222">
Text in another element BBB.
<DDD ddd="999">
  Text in more nested element.</DDD>
  </BBB>
  <CCC ccc="123" xxx="321">
    Again some text in some element.</CCC>
</AAA>
```

- X marks point (zero width)
- This style marks the container node.
- This style marks resulting locations.
- X marks collapsed range (zero width)

Full XPointer expression

Full XPointer expression may consist of one or more XPointer parts. (Or Schemes, but currently only XPointer scheme is allowed). Separation with whitespaces is not mandatory. The result of the first XPointer part whose evaluation (in left-to-right order) succeeds is taken to be the fragment located by XPointer as a whole.

XPointer: xpointer(//AAA/BBB[1]) xpointer(//AAA/BBB[2])

XPointer (alternative): xpointer(//AAA/BBB[1])xpointer(//AAA/BBB[2])

```
<AAA>
  <BBB myid="b1" bbb="111">
    Text in the first element BBB.</BBB>
  <BBB myid="b2" bbb="222">
    Text in another element BBB.
    <DDD ddd="999">
      Text in more nested element.</DDD>
    <DDD ddd="888">
      Text in more nested element.</DDD>
    <DDD ddd="777">
      Text in more nested element.</DDD>
  </BBB>
  <CCC ccc="123" xxx="321">
    Again some text in some element.</CCC>
</AAA>
```

- X marks point (zero width)
- This style marks the container node.
- This style marks resulting locations.
- X marks collapsed range (zero width)

XML escaping

XPointer escaping

Elements

XLink

- **A replacement for HTML's <A>**
  - plus extra capabilities
    - *although the actual implementation of these is left to the processing application*
  - a flexible vocabulary for connecting documents and document fragments
    - 'XLink is an "enabling" vocabulary; you don't use it by itself, but rather incorporate it into your own vocabulary'
    - *an arbitrary element can be specified as a link,*
      - the special *xml:link* attribute is recognised
  - ***Note***: the traversal mechanism/visual representation of an XLink is defined to be left to an associated style sheet to specify
    - neither CSS or XSL (currently…) provide anything that can be used to define such a mechanism!

Link Species

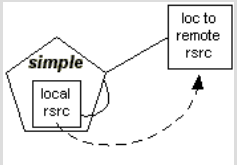
- **Two types of link are defined, with two options**
  - simple
    - *nearest match to <A>*
    - *target is single document or resource and link can only be traversed in one direction*
  - extended
    - *link target can be several resources and the links can define a complex relationship. Links can be bi-directional*

Simple Links

- **Equivalent of <A HREF="…">**
  - interesting stuff
    - *show=embed*
      - "transclusion"
    - *actuate*
      - onLoad, onRequest
      - other
        - *perhaps after a given time or in conjunction with other documents*
    - *role*
      - describe the function of a link's remote resource in a machine-readable fashion
        - *perhaps a name or an identifier useful in 'intelligent' traversal*

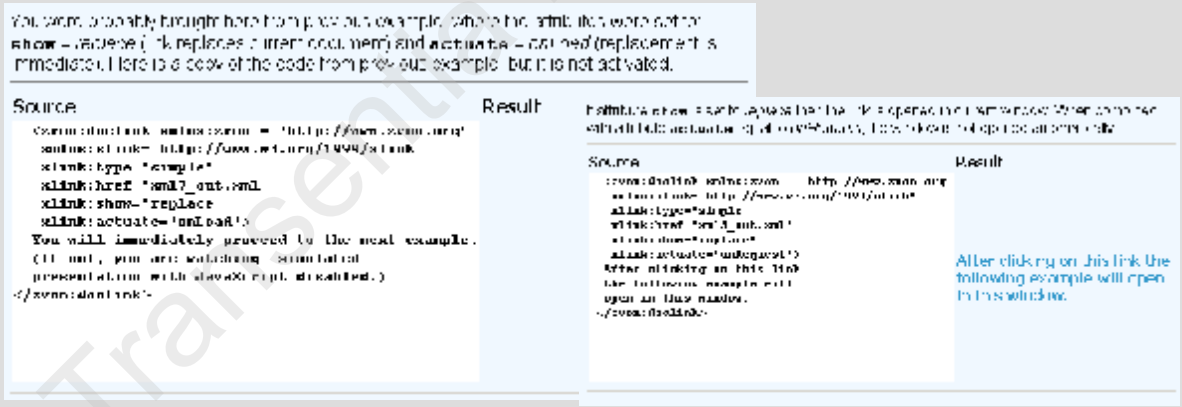
```
<simplelink xlink:type="simple"
            xlink:href="#xpointer(//prod[@num='22'])"
            xlink:show="replace"
            xlink:actuate="onRequest">production 22</simplelink>.
```

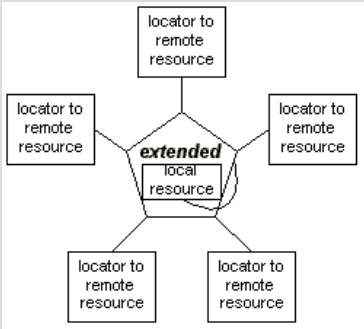
```
<!ELEMENT studentlink ANY>
<!ATTLIST studentlink
  xlink:type      (simple)        #FIXED "simple"
  xlink:href      CDATA           #IMPLIED
  xlink:role      NMTOKEN         #FIXED "http://www.example.com/linkprops/student"
  xlink:arcrole   CDATA           #IMPLIED
  xlink:title     CDATA           #IMPLIED
  xlink:show      (new
                  |replace
                  |embed
                  |other
                  |none)          #IMPLIED
  xlink:actuate   (onLoad
                  |onRequest
                  |other
                  |none)          #IMPLIED>
```



Extended Links

- **Not supported by any mainstream application as yet**
  - difficult to implement; may also need infrastructural support…
  - *"As link management software develops, the option of storing links externally to documents will become both easier to implement and more efficient to manage…"*
  - link element may contain a mixture of the following:
    - *locator-type elements that address the remote resources participating in the link*
    - *arc-type elements that provide traversal rules among the link's participating resources*
    - *title-type elements that provide human-readable labels for the link*
    - *resource-type elements that supply local resources that participate in the link*





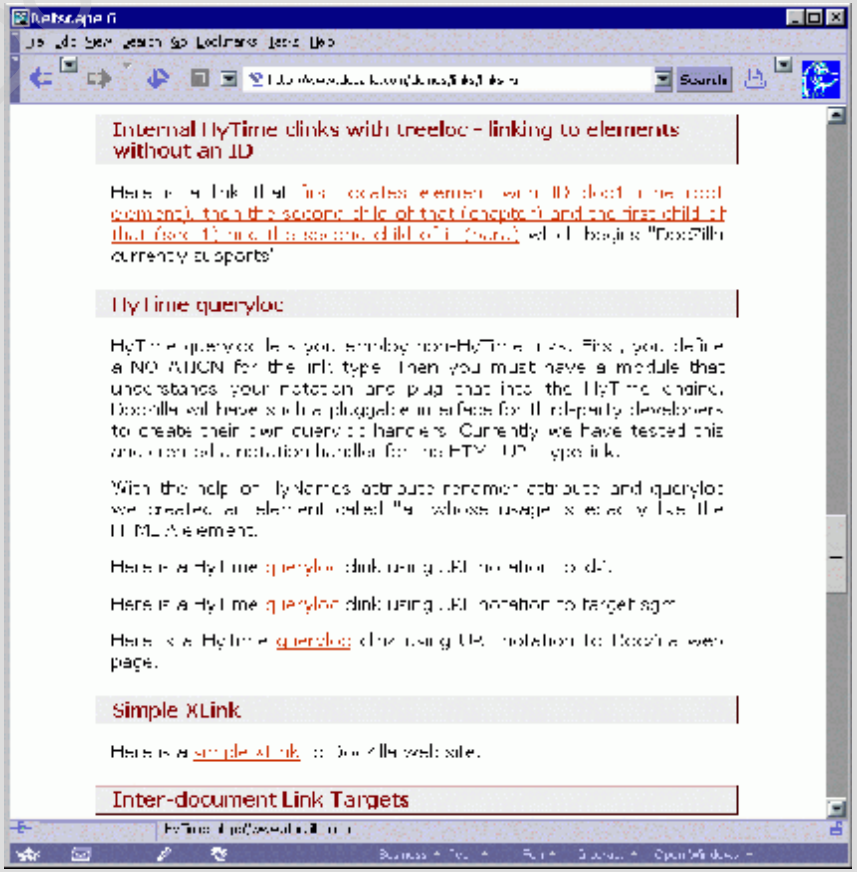
Extended Link Attributes

- **When xml:link="extended"**
  - intersting stuff
    - *arc*
      - pair up participants and indicate which is the starting point and which is the ending point for link traversal
        - *inbound, outbound, third-party*
      - define traversal rules for the link
    - *locators to resources*

```
<extendedlink xlink:type="extended">
  <loc xlink:type="locator"
       xlink:label="prolog-ref"
       xlink:href="#xpointer(string-range(//text(),'prolog'))" />
  <loc xlink:type="locator"
       xlink:label="prolog"
       xlink:href="#/1/2/5" />
  <arc xlink:type="arc"
       xlink:from="prolog-ref"
       xlink:to="prolog"
       xlink:show="new"
       xlink:actuate="onRequest" />
</extendedlink>
```

Software

- **Various, small-scale implementations**
  - mostly concerned with client side
  - mostly simple linking
  - generally immature
  - From http://www.w3.org/XML/Linking:
    - *X2X*
      - *seems* the most mature
    - *Fujitsu XLink Processor (XLip)*
    - *xlinkit.com*
    - *Mozilla*
      - DocZilla
    - *Amaya*
    - *XLink2HTML*
    - *XTooX*



XML Query

- **"The goal of XML Query is to be a data model for XML documents, a set of query operators on that data model, and a query language based on these query operators."**
  - *a functional language*
    - a query is represented as an expression
  - *based on OQL, SQL, XML-QL, XPath*
  - *readable syntax*

List each publisher with the average price of its books

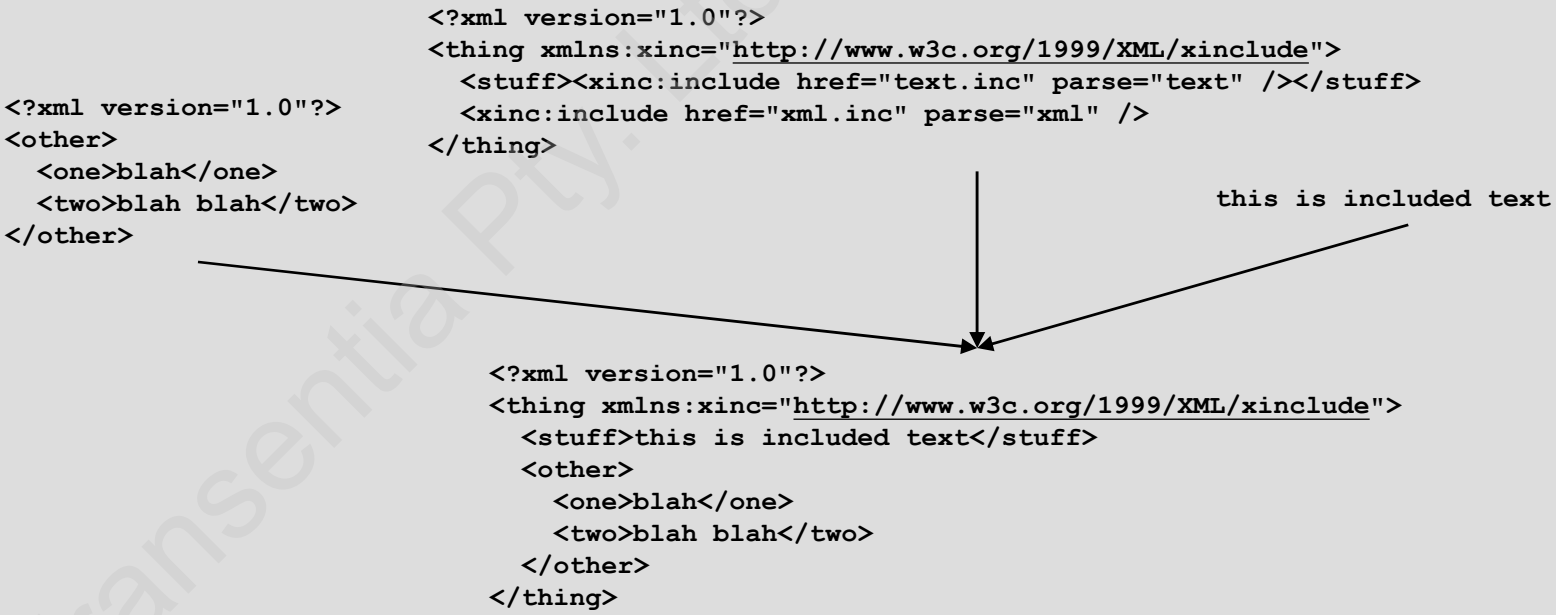
```
FOR $p IN distinct(document("bib.xml")//publisher)
LET $a := avg(document("bib.xml")
    /book[publisher = $p]/price)
RETURN
    <publisher>
        <name> $p/text() </name>,
        <avgprice> $a </avgprice>
    </publisher>
```

List the titles of books published by Wiley in 2000

```
FOR $b IN document("bib.xml")//book
WHERE $b/publisher = "Wiley"
    AND $b/year = "2000"
RETURN $b/title
```

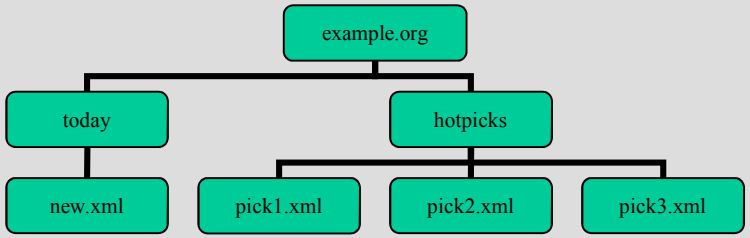
XInclude

- **Defines a syntax for general-purpose, XML-based inclusions**
  - still work in progress…
  - similar to C's #include
  - merges infosets
    - *trees, that is*



XML Base

- Allows authors to explicitly specify a document's base URI
  - makes it easier to resolve relative links to external resources
    - *images, applets, style sheets, etc.*
- A modular specification so that other parts of XML can also make use of it
- Introduces a single XML attribute named xml:base




```
<?xml version="1.0"?>
<doc xml:base="http://example.org/today/"
     xmlns:xlink="http://www.w3.org/1999/xlink">
  <head>
    <title>Virtual Library</title>
  </head>
  <body>
    <paragraph>See <link xlink:type="simple" xlink:href="new.xml">what's new</link>!</paragraph>
    <paragraph>Check out the hot picks of the day!</paragraph>
    <olist xml:base="/hotpicks/">
      <item><link xlink:type="simple" xlink:href="pick1.xml">Hot Pick #1</link></item>
      <item><link xlink:type="simple" xlink:href="pick2.xml">Hot Pick #2</link></item>
      <item><link xlink:type="simple" xlink:href="pick3.xml">Hot Pick #3</link></item>
    </olist>
  </body>
</doc>
```