XPointer and XLink

Michael B. Spring
Department of Information Science and Telecommunications
University of Pittsburgh
spring@imap.pitt.edu
http://www.sis.pitt.edu/~spring

Context

- XML adds new capabilities to documents on the World Wide Web
- These capabilities will not be immediately apparent to end users as it will be some time before servers and browsers implement the capability
- The XML Pointer Standard defines a mechanism for locating a point or region within an XML document
- The XML Linking Language (XLL) provides a new set of capabilities for links between documents

Overview

- XPointer
  - Points
  - Regions
  - Sets of regions
- XLink
  - Simple Links
  - Extended Links

XPointers

- Path information allows a link to be made to a specific location within a document using XPointer
- Xpointer extends the capabilities of URI, URL, URN, and fragment identifier
- In some ways, XPointer is a shell for Xpath
  - consider the following url:
    - http://www/c/g/xyz.xml#xptr(/mydoc/chap[3])

The Uses of XPath

- Having XPath, the question is why all this functionality.
- In the remaining slides we address two standards that define application functionality built upon XPath.
- XPointer is a way of identifying a location within an XML document. We will cover most of what is defined for XPointer
- XSLT is used to match, select, choose, and filter sets of nodes from an input document passed to an output document. XSLT is a very complex standard, and we will only overview it here.

XPointers

- Under html, a reference to a part of a document – a fragment could be made using the following:
  - <A NAME = “SOMEPLACE”>some info</A>
  - <A HREF = “#SOMEPLACE”>anchor to “info”</A>
- XPointer allows a link to be made to a specific location within a document using XPath
- Any of the XPath specifications could be used
- For example, the following URL would point to the third chapter within doc
  - http://abc.com/xyz.xml#xptr(/doc/chap[3])

XPointer Options

- XPointer supports the use of “|” as well as “#” to separate the fragment identifier. The “|” implies that a conforming server would only return the fragment.
- XPointer allows absolute location specification using:
  - id(xyz) would locate the element in the tree with the specified id.
  - root() would locate the root element
  - html(xyz) would locate the element that contains the old style html named anchor

XPointer Options (continued)

- More interesting uses of XPointers involve the specification of relative locations
  - #root().child(6,ITEM) would select the 6th ITEM element that is a child of the root – if one exists.
  - #root().descendant(3,BOOK) would select the third BOOK element that is a descendant of root – i.e., anywhere in the tree
- Similarly, ancestor looks up the tree, preceding and following look at preceding and following nodes in a list
- Finally psibling and fsibling look at the preceding and following sibling nodes

Still More XPointer Options

- Each of the operations (child, preceding, etc.) allows for two arguments in the form child(number, NAME).
  - If only a number is presented, that node, regardless of type is selected
  - If only a name is presented, all those elements, regardless of position are selected
  - Where the name is placed, node type may also be used where the types allowed include: #text,#comment, #cdata, #element, #pi, #all
- In reality, a third and fourth argument are allowed specifying attribute name and value

Strings

- A specific string or set of strings can be identified using the form:
  - #string(1, “Hello”) which targets the first occurrence of the string Hello. This would target the position right before the H in Hello
  - #string(3, “Hello”, 5) targets the position 5 characters after the position before the third Hello, i.e. right after the o.
  - #string(1, “Hello”, 1, 5) would target the first “Hello”
  - #string(200,””,1,100) would target the characters from 200-300
  - #string(all, “Pitt”) would target all instances of “Pitt”

Spans

- A span of text can be specified using the keyword span
  - #root().span(child(1), child(3)) would span the text from the first to the third child of the root node.

XLINK

The XML Linking Language (XLL)

- XLL provides more linking capability
- simple linking, like that in html would look as follows
  - <citation xlink:type="simple“
  - xlink:href=URL>text</citation>
- use of the xlink attributes requires the xlink namespace
  - <rootname xmlns:xlink="http://www.w3.org/XML/Xlink/0.9">

XLINK Attributes

- the definition of Xlink allows a variety of different link types to be developed.
- many of these are defined by the show attribute of xlink; xlink:show may be set to the following values
  - "replace" does what we see on the WWW
  - "new" causes a new window to be opened
  - "parsed" causes the href to be parsed and included
- another attribute of xlink is actuate which can take the following values
  - "user" indicates that traversal is based on user action
  - "auto" specifies that traversal should be automatic

Extended Links

- extended links include links that make use of the locator element
  - <mylink xlink:type="extended">
    - <locator xlink:type="locator“ xlink:href = "url“
    - xlink:role="type of link">
    - <locator xlink:type="locator“ xlink:href = "url“
    - xlink:role="type of link">
  - </mylink>
- link groups allow sets of documents to be linked together
- behavior and processing of these is undefined
  - <xlink:group>
    - <xlink:document href="url"/>
    - <xlink:document href="url"/>
    - <xlink:document href="url"/>
  - </xlink:group>

XLINK

The XML Linking Language (XLL)

- XLL provides more linking capability
- simple linking, like that in html would look as follows
  - <citation xlink:type="simple“
  - xlink:href=URL>text</citation>
- use of the xlink attributes requires the xlink namespace
  - <rootname xmlns:xlink="http://www.w3.org/XML/Xlink/0.9">

XLINK Attributes

- the definition of Xlink allows a variety of different link types to be developed.
- many of these are defined by the show attribute of xlink; xlink:show may be set to the following values
  - "replace" does what we see on the WWW
  - "new" causes a new window to be opened
  - "parsed" causes the href to be parsed and included
- another attribute of xlink is actuate which can take the following values
  - "user" indicates that traversal is based on user action
  - "auto" specifies that traversal should be automatic

Extended Links

- extended links include links that make use of the locator element
  - <mylink xlink:type="extended">
    - <locator xlink:type="locator“ xlink:href = "url“
    - xlink:role="type of link">
    - <locator xlink:type="locator“ xlink:href = "url“
    - xlink:role="type of link">
  - </mylink>
- link groups allow sets of documents to be linked together
- behavior and processing of these is undefined
  - <xlink:group>
    - <xlink:document href="url"/>
    - <xlink:document href="url"/>
    - <xlink:document href="url"/>
  - </xlink:group>

XPointers

- Path information allows a link to be made to a specific location within a document using XPointer
- Xpointer extends the capabilities of URI, URL, URN, and fragment identifier
- In some ways, XPointer is a shell for Xpath
  - consider the following url:
    - http://www/c/g/xyz.xml#xptr(/mydoc/chap[3])