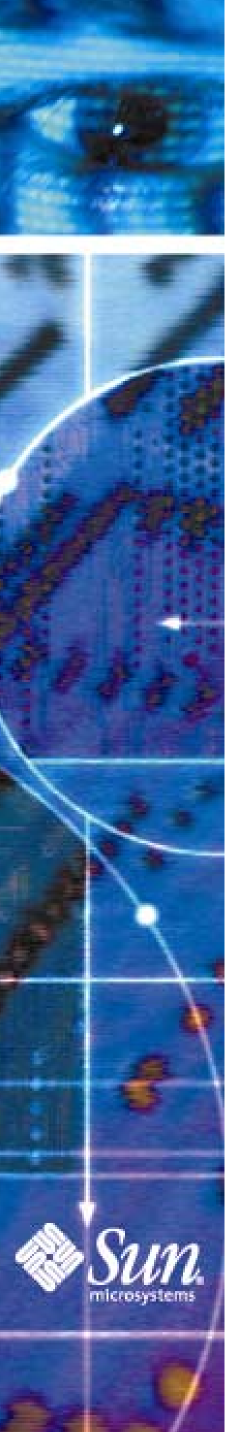
Lecture 0

XPointer

Sang Shin
Java™ Technology Evangelist
sang.shin@sun.com



Topics

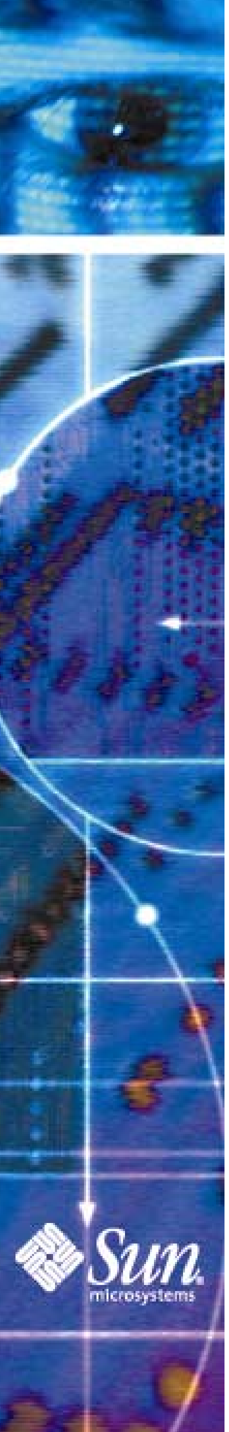
- Xpointer Syntax
- Xpointer Extension to Xpath
  - Bare names
  - Child Sequence
  - Points
  - Ranges

Xpointer Overview

- Non-XML syntax
- Used as fragment identifier (not the whole XML document)
- Attached to the end of URI
- Builds on syntax of Xpath
  - Adds points and ranges

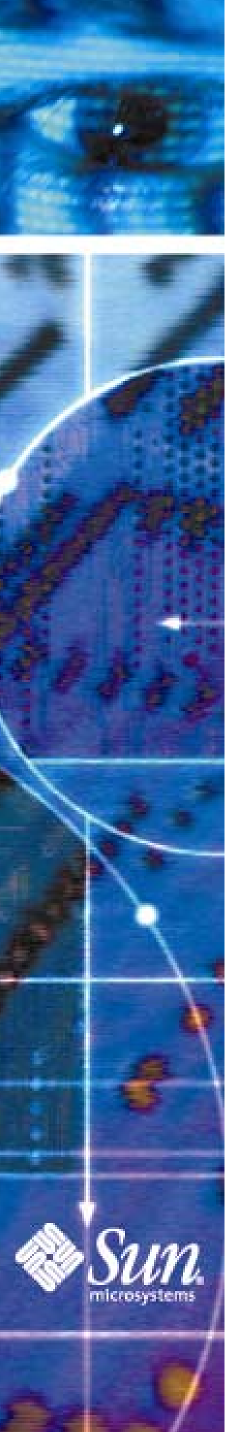
Motivation for XPointer

- HTML anchoring is inconvenient
  - ◆ You need to change HTML document in order to insert an anchor
- Need for more fine-grained referencing
  - ◆ Range of text currently selected by the mouse in an editor



XPointer Syntax

- Xpath expression enclosed in xpointer()
- May identify zero, one, or more than one node
- Mostly element and attribute nodes
- Two new node types
  - point
  - range



XPointer Examples

- xpointer(/)
- xpointer(//first_name)
- xpointer(id(‘sec_intro’))
- xpointer(/people/person/name/first_name/text())
- xpointer(//middle_initial[position()=1]/../first_name)
- xpointer(//professional[.=“physicist”])
- xpointer(/child::people/child::person[@id<4000])
- xpointer(/child::people/child::person/attribut::id)

Identification of Multiple Elments

- By stringing them together
- xpointer(//first_name)xpointer(//last_name)
  - All first_name and last_name elements
- xpointer(//first_name)xpointer(//last_name)xpointer(//middle_initial)
  - All first_name, last_name, and middle_name elements

Xpointer Usage

- First name element in the document at http://www.ibiblio.org/xml/people.xml
  - http://www.ibiblio.org/xml/people.xml#xpointer(//name[position()=1])
- Browser or application behavior is not determined

Xpointer with Simple Xlink

- First book child of the bookcoll child of the testament root element in a relatively located document ot.xml

```
<link xlink:type=“simple”
        xlink:href=“ot.xml#xpointer(
                        /testament/bookcoll/book[position()=1])”>
      Genesis
</link>
```

Xpointer with Extended Xlink

- Identify starting and ending resources of an arc

```
<link xlink:type=“extended” >
  <testament xlink:type=“locator” xlink:label=“ot”
    xlink:href=“ot.xml#xpointer(//v[position()=last()])”/>
  <testament xlink:type=“locator” xlink:label=“nt”
    xlink:href=“ot.xml#xpointer(//v[position()=1])”/>

  <next  xlink:from=“ot”  xlink:to=“nt”/>
  <previous xlink:from=“nt” xlink:to=“ot”/>
</link>
```

Xpointer for Internal linking

```
<slide>
  <previous xlink:type=“simple” xlink:href=
  “xpointer(ancester::slide/preceding-sibling::slide[position()=1])”>
      Back
  <next xlink:type=“simple” xlink:href=
 “xpointer(ancester::slide/following-sibling::slide[position()=1])”>
      Forward
  </next>
</slide>
```

Xpointer Extensions to Xpath

- Bare names
- Child Sequence
- Points
- Ranges

Bare Names

- Identifies the element by the value if its ID attribute
- Convenient shorthand of Xpath expression id()
- They are same
  - http://www.brandeis.edu/parentElement#idValueofElement
  - http://www.brandeis.edu/parentElement#xpointer(id('idValueofElement'))

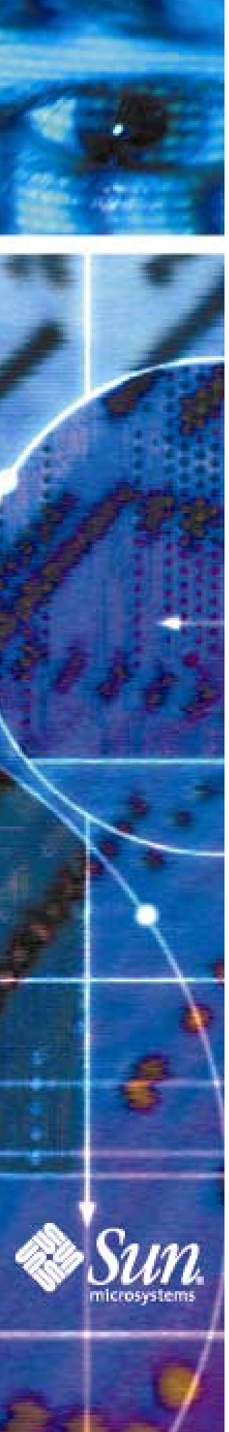
Child Sequences

Points

- Xpath, bare names, and child sequences point only to entire node or set of nodes
- Points are zero dimensional
- A point is identified by its container node and an index into that node

Points

- If the container node has child nodes
  - Document, or element nodes
  - Points exist before and after each of its children
- Otherwise
  - comment, processing instruction, attribute, text nodes
  - Points exist before and after each character in the node’s string value



Example

- Points inside *novel* element

```
<novel copyright="public domain">0
    1<title>The wonderful wizard of Oz</title>2
    3<author>L. Frank Baum</author>4
    5<year>1900</year>6
7</novel>
```

Example

- Points inside *year* element

```
<novel copyright=“public domain”>
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>112930405</year>
</novel>
```

Point Syntax

- Use *point* in an Xpath expression

Example

- xpointer(//title[position()=1]/text()/point[position()=3])

```
<novel copyright="public domain">
    <title>The* wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```

Example

- xpointer(/novel/point[position()=2])

```
<novel copyright="public domain">
    *<title>The wonderful wizard of Oz</title>
     <author>L. Frank Baum</author>
     <year>1900</year>
</novel>
```

Example

- xpointer(/novel/text()[position()=1])/point[position()=3])
- Count whitespace characters

```
<novel copyright="public domain">
   *<title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```

start-point()

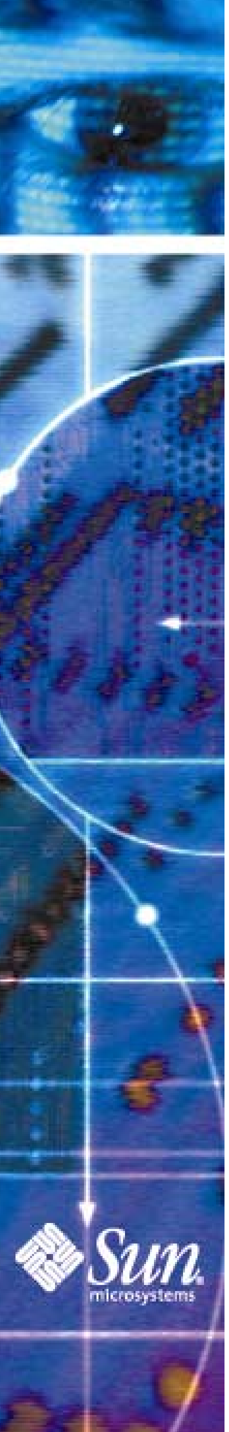
- Immediately before
- xpointer(start-point(//title))

```
<novel copyright=“public domain”>
    *<title>The wonderful wizard of Oz</title>
     <author>L. Frank Baum</author>
     <year>1900</year>
</novel>
```

end-point()

- Immediately after
- xpointer(end-point(//author))

```
<novel copyright=“public domain”>
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>*
    <year>1900</year>
</novel>
```



Ranges

- Span of parsed character data between two points
  - May or may not represent a well-formed chunk of XML
- Represented by
  - range()
  - range-inside()
  - range-to()
  - string-range()

range()

- Take Xpath expression as an argument
- Returns a node set, which is then used as xpointer argument
- For each node in a node set, xpointer returns a range
    - start point is the point immediately before the node
    - end point is the point immediately after the node

Example

- xpointer(range(//title))

```
<novel copyright=“public domain”>
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```

Example

- xpointer(range(/nove/*)) returns 3 ranges

```
<novel copyright=“public domain”>
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```

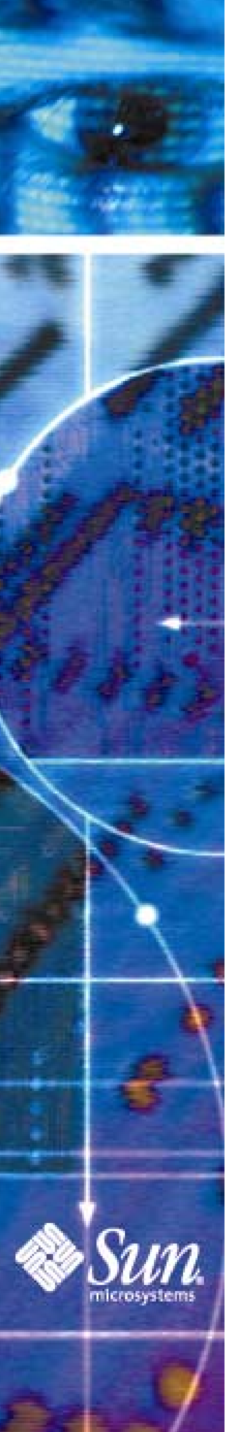
range-inside()

- Same as range() except element nodes
- For element nodes, everything inside the starting and end tags

Example

- xpointer(range-inside(//title))

```
<novel copyright="public domain">
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```



range-to()

- Take Xpath expression as an argument
- Returns a node set, which is then used as xpointer argument
- start points - start points of context nodes
- end points - end points in the argument

Example

- xpointer(/title/range-to(year))

```
<novel copyright=“public domain”>
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```

Example

- xpointer(/title/range-to(/title/text()))

```
<novel copyright="public domain">
    <title>The wonderful wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```

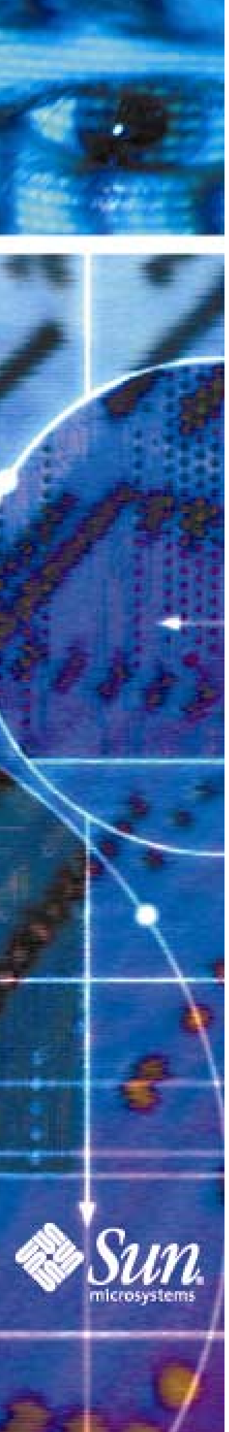
string-range()

- Operates on text of a document after all the markup has been stripped from it
- For each node in a node set, match string argument against the text of the node
  - Returns a range that for all occurrences of the matched string
- Can specify the offset and length

Example

- xpointer(string-range(//title, “Wizard”))

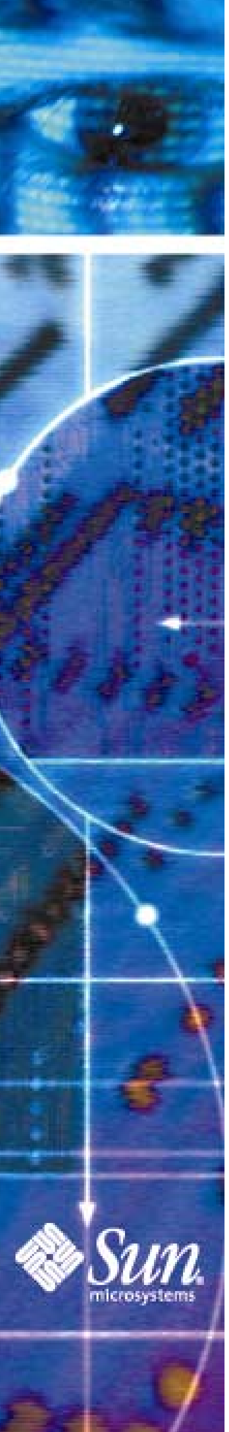
```
<novel copyright=“public domain”>
    <title>The Wonderful Wizard of Oz.
            Another Wizard </title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```



Example

- xpointer(string-range(//title, “Wizard”, 5, 4))

```
<novel copyright=“public domain”>
    <title>The Wonderful Wizard of Oz</title>
    <author>L. Frank Baum</author>
    <year>1900</year>
</novel>
```



Summary

- Xpointer Syntax
- Xpointer Extension to Xpath
    - Bare names
    - Child Sequence
    - Points
    - Ranges