Navigating the XML Tree

Syd Bauman
Navigating the XML Tree

XML representations

```xml
<?xml version="1.0" encoding="UTF-8"?>
<book>
  <introduction>Blah blah blah ... </introduction>
  <chapter>
    <heading>Wines</heading>
    <section>White wines ... </section>
    <section>Red wines ... </section>
  </chapter>
  <chapter>
    <heading>Beers</heading>
    <section>Ales ... </section>
    <section>Lagers ... </section>
  </chapter>
  <index> stuff ... </index>
</book>
```

Sample Portion of TEI Tree

Most of you may remember this grossly simplified sub-tree representation of a TEI document.
Navigating the XML Tree

Walking the Tree
tree (with self:::)

Navigating the XML Tree
descendants (child:: and descendant::)
ancestors (parent:: and ancestor::)
preceding (preceding:: and preceding-sibling::)
following (following::: and following-sibling:::)

Navigating the XML Tree
Navigating the XML Tree

axes :-)

XPath ...

... lets us (among other things) select nodes in the tree.

... is used by XSLT, XQuery, Schematron; XLink and XPointer, too.
Navigating the XML Tree

basic filepath-like path expressions

A bare-bones path expression is similar to filesystem addressing: if the path starts with a solidus (/ aka "forward slash"), then it represents a path from the root; if it does not start with a solidus then it represents a path from "here".

/TEI/teiHeader/fileDesc/titleStmt/title

list/item/label

Try it!

In oXygen open the file http://www.wwp.brown.edu/outreach/seminars/uvic_xslt_2013/demos/xslt_intro/ham.xml. (To open a URL use File > Open URL... or #-u (Mac) or ctrl-u (Windows).)

In the upper L corner there is a text-entry field (looks like a search box). Ensure the box is labelled XPath 2.0. Then type in /TEI/teiHeader/fileDesc/titleStmt/author. Notice that you have content completion, so it is very easy to type. (If you don’t, set it in Options / Preferences ; Editor / Content Completion / XPath.)

Here is what the oXygen XPath console will look like.
XPath axes: me, my descendants, and my ancestors

<table>
<thead>
<tr>
<th>Short</th>
<th>Long</th>
<th>Means</th>
<th>Try it!</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>self::node()</td>
<td>me, whatever I happen to be (element, attribute, comment, processing instruction, document root, or even text) (note: long and short are not exactly equivalent)</td>
<td>oXygen shows node cursor is in</td>
</tr>
<tr>
<td>said</td>
<td>child::said</td>
<td>my &lt;said&gt; children</td>
<td>list all &lt;head&gt; elements for &lt;div&gt; children of the body</td>
</tr>
<tr>
<td>*</td>
<td>child::*</td>
<td>all my element children</td>
<td>list all children of &lt;titleStmt&gt;</td>
</tr>
<tr>
<td>//div</td>
<td>descendant::*</td>
<td>my &lt;div&gt; descendants (note: long and short are not exactly equivalent)</td>
<td>list all stage directions (&lt;stage&gt;)</td>
</tr>
<tr>
<td>..</td>
<td>parent::node()</td>
<td>my parent, whatever it happens to be (element directions or root)</td>
<td>list the parents of stage directions</td>
</tr>
<tr>
<td>[none]</td>
<td>parent::author</td>
<td>my parent, if it is an &lt;author&gt;</td>
<td>list the speech (&lt;sp&gt;) parents of stage directions</td>
</tr>
<tr>
<td>[none]</td>
<td>ancestor::*div</td>
<td>my &lt;div&gt; ancestors</td>
<td>list all the headings of the &lt;div&gt; ancestors of the famous &quot;To be, or not to be&quot; (which you can find by searching for it)</td>
</tr>
<tr>
<td>[none]</td>
<td>ancestor::*</td>
<td>all my ancestors</td>
<td>list the ancestors of the &lt;author&gt; element</td>
</tr>
</tbody>
</table>
Navigating the XML Tree

(XKay, chapter 9)

XPath axes: following, preceding, and attributes

<table>
<thead>
<tr>
<th>Short</th>
<th>Long</th>
<th>Means</th>
<th>Try it!</th>
</tr>
</thead>
<tbody>
<tr>
<td>[none]</td>
<td>following::item</td>
<td>all the &lt;item&gt;s that come after me</td>
<td>find all the &lt;title&gt; elements that occur after the &lt;titleStmt&gt;</td>
</tr>
<tr>
<td>[none]</td>
<td>following-sibling::l</td>
<td>the &lt;l&gt; children of my parent that come after me (i.e., the rest of the stanza)</td>
<td>find all the children of &lt;titleStmt&gt; that follow &lt;author&gt;</td>
</tr>
<tr>
<td>[none]</td>
<td>preceding::pb</td>
<td>the &lt;pb&gt;s that come before me</td>
<td>find all the &lt;ab&gt; elements that occur before &quot;To be, or not to be:&quot;</td>
</tr>
<tr>
<td>[none]</td>
<td>preceding-sibling::head</td>
<td>the &lt;head&gt; children of my parent that come before me</td>
<td>do any elements ever occur before the &lt;speaker&gt; within a speech?</td>
</tr>
<tr>
<td>@rend</td>
<td>attribute::rend</td>
<td>my @rend attribute</td>
<td>list all the @who attributes</td>
</tr>
<tr>
<td>@*</td>
<td>attribute::*</td>
<td>all of my attributes</td>
<td>do any of the &lt;stage&gt; elements have attributes at all?</td>
</tr>
</tbody>
</table>

(Kay, chapter 9)

Don’t want them all?

- Many of the above XPaths return multiple nodes — what if you only want a particular one?
- If you only want Act 3, Scene 1:
Navigating the XML Tree

/TEI/text/body/div[3]/div[1]

- Works well presuming you know what you want by element count.
- But in many cases, that is at least inconvenient, if not outright unknown.
- No matter how many `<div>`s there are, we know this scene has the identifier "sha-ham301". Thus:

```
//div[@xml:id = 'sha-ham301']
```

selects the same node.

predicates

An XPath *predicate* filters the nodes retrieved by a given step

Predicates are expressed after the node test in square brackets

The following are based on http://www.wwp.brown.edu/outreach/seminars/uvic_xslt_2013/demos/xslt_intro/places.xml

XPath

```
//listPlace/place[1]
```

selects the first `place` of each `listPlace` (of which there only happens to be one)

```
//*[@cRef]
```

all elements that have a `@cRef` attribute

```
//title[@level='m']
```

all monographic titles

```
/TEI/text//name[ not( @key ) ]
```

`<name>` elements that are missing their `@key` attributes

```
//lg[@type='song']/l[1]
```

list first line of each song (16 nodes)

```
//lg[@type='song']/l[1][1]
```

returns first line of all songs (1 node)
Sample Document Instance

<?xml version="1.0" encoding="UTF-8"?>
<lg type="limerick" rhyme="aabba" n="3">
  <head>Warp Speed, Ms Bright!</head>
  <l>There was a young lady named <rhyme label="a">Bright</rhyme>,</l>
  <l>Who travelled much faster than <rhyme label="a">light</rhyme>,</l>
  <l>She departed one <rhyme label="b">day</rhyme>,</l>
  <l>In a <term xml:id="t17">relative</term> way <rhyme label="b">way</rhyme>,</l>
  <l>And returned on the previous <rhyme label="a">night</rhyme>.</l>
</lg>

XSLT View

Same document, but adding text and attribute nodes
Element nodes have just the element’s local name; attribute nodes have just the name of the attribute; text nodes have just the word text.