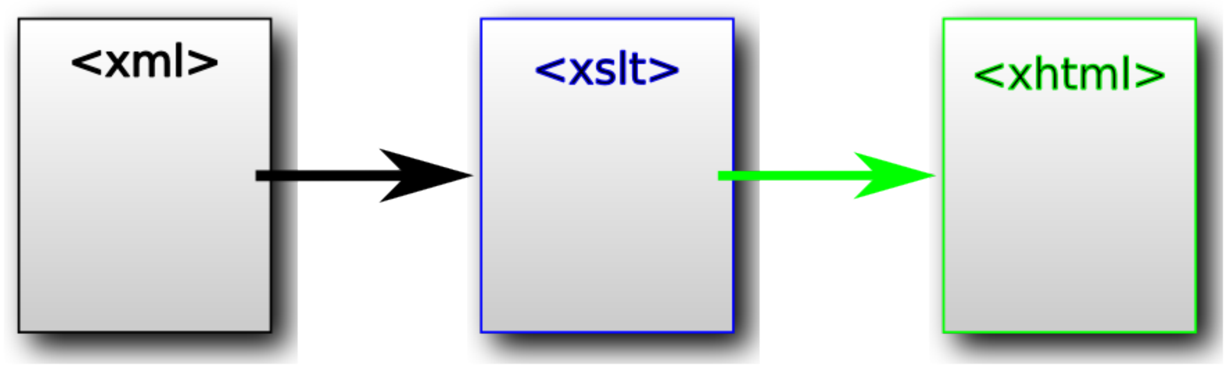
The XPath doc() function: using input from another document

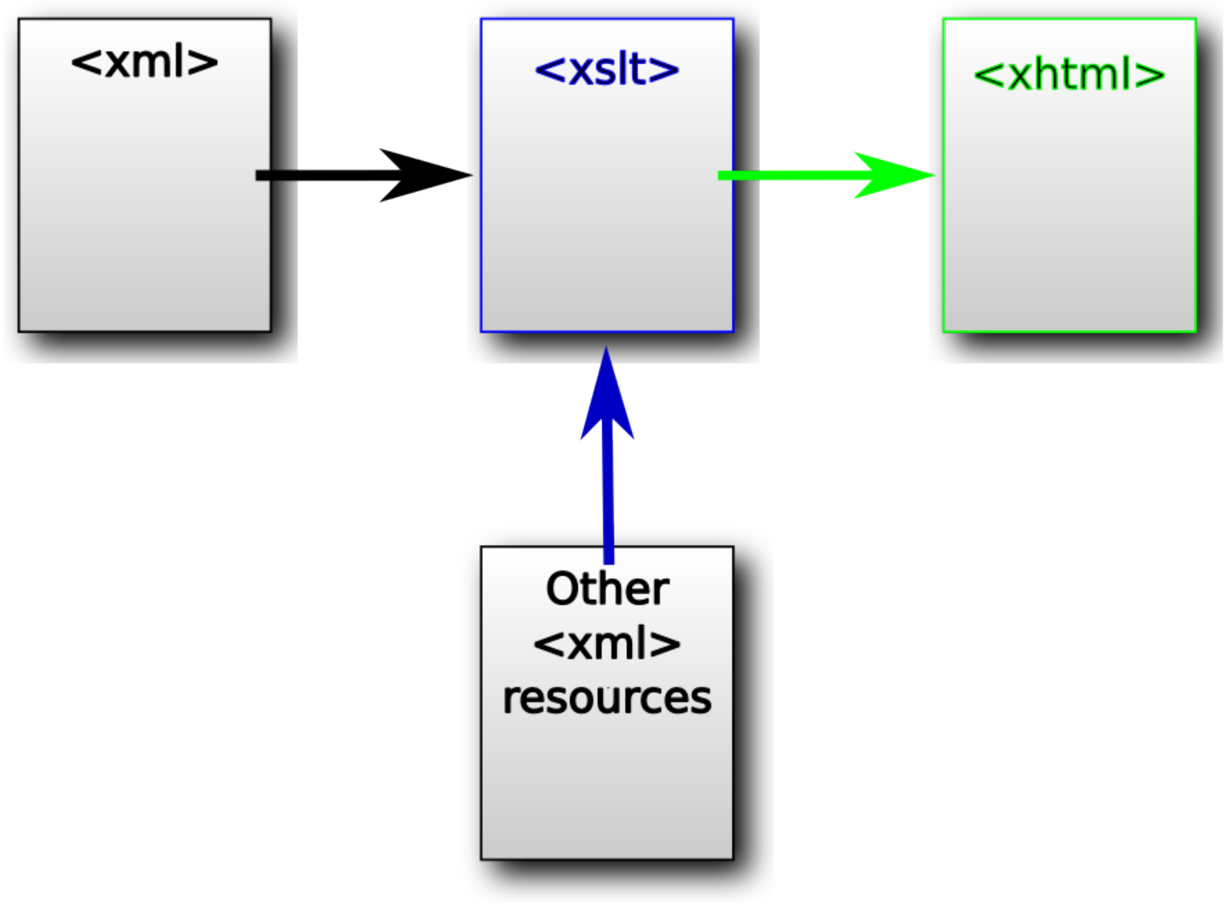
Martin Holmes

What we know so far



What's the problem with this simple scenario, especially in the case of large projects?

Information from another document



What kinds of external data might we bring in?

- personography
- placeography
- orgography
- footnotes
- editorial annotations
- critical apparatus

- another edition of the same text

The `doc()` function

The `doc()` function takes a URI parameter, pointing to the external file.

```
doc('personography.xml')
```

```
doc('../people/personography.xml')
```

```
doc('http://mapoflondon.uvic.ca/literary_personography.xml')
```

Point out that what you're providing is either a relative or an absolute URI. Generally, with your own files, what you'll be using is a relative URI, but you can easily import data from any available source on the Web, as in the last example.

A worked example

- Download the example short poem "Art" in XML: http://web.uvic.ca/~mholmes/dhoxss2013/examples/art.xml
- Download the corresponding simple XSL file: http://web.uvic.ca/~mholmes/dhoxss2013/examples/art.xsl
- Set up a transform in the XSLT debugger, or through a Transformation Scenario, whichever suits you best. We'll analyse the transformation as it now works.

First, get someone to outline how the XML file works. Make sure they point out persName and @key.

Get someone else to outline how the stylesheet is working. In particular: suppressing the teiHeader, and outputting the stanzas as lists but without bullets or numbers. Note that there's nothing we can do with persName because the key data is elsewhere.

A worked example (2)

- Now download the Tiny Personography: http://web.uvic.ca/~mholmes/dhoxss2013/examples/tiny_personography.xml
- This contains the data we need.

Now we work through the process of incorporating the data.

- Create a variable $personography for the new data. Elicit the syntax for loading the file.
- Next, address the issue of adding a footnote number after persName. This is a review of our presentation on template modes. Create a template for persName.
- Review how to calculate the footnote number by counting previous persNames. Put that information in a variable.
- Render out the persName, then create a link to a so-far non-existent footnote using fn_ + the footnote number.
- Style the footnote using the sup element.
- Check that's working.
- Now ask: how do we process the persNames again to create the footnotes?
- Create a new template for persName with mode="footnotes".
- Again, calculate the footnote number and store the value in a variable.
- Output a paragraph including the number and a period. Check this works.
- Now create a $person variable which is a pointer to the person entry in the $personography variable.
- Output the $person/persName in the footnote, followed by a colon. Check that works.
- Add xsl:apply-templates on the note element. View the results.
- Now get everyone to tell you exactly what to do about the ref element.

A worked example (3)

- You'll find a downloadable working version of what we just did here: http://web.uvic.ca/~mholmes/dhoxss2013/examples/art_plus.xsl

Summary

- The `doc()` function takes a URI parameter, pointing to the external file.

```
doc('personography.xml')
```

```
doc('../people/personography.xml')
```

```
doc('http://mapoflondon.uvic.ca/literary_personography.xml')
```

- It's helpful to load the document into a variable at the beginning of your transformation so it's easy to refer to it anywhere.
- If you want to check whether the document exists before loading it, use `doc-available()`:

```
<xsl:variable name="personography"
select="if (doc-available('tiny_personography.xml'))
then doc('tiny_personography.xml') else ()"/>
```