Exercise: Processing Linking with XSLT

James Cummings *January 2010*

1 Using XSLT to follow links

In this short exercise, we will use the name and person markup we introduced earlier. You'll need a file which includes `<person>` and `<persName>` elements linked together.

1. Using OxyGen, open the file sample-1.xml which you prepared in the previous exercise, and check that it is still a valid TEI XML document. If you've lost or corrupted it, you can use the file we prepared for you instead: it is called sample.xml (if you do this, remember that you will need to associate it with the ex1.xsl XSLT stylesheet).
2. In this exercise, we will enhance the display of the sample file by having a tool-tip style popup appear when we mouse over each `<persName>` element. The popup will contain text taken from the `<person>` element associated with that name, if we have one. If we don't, it will simply replace the name by a link to an external Wikipedia or DNB entry for the relevant person.
3. Open the ex1.xsl stylesheet in OxYgen. We will add two new templates, one for the case where the ref attribute on `<persName>` points to a `<person>` element in the header, and one for the case where it points to some other web page.
4. The second case is easy. You can assume that if the value of the ref attribute on a `<persName>` element starts with http, all you need to do is turn the `<persName>` into an HTML `<a>` element, with an appropriate value for its href attribute. That requires a template which will match `persName[starts-with(@ref,'http:')]`
5. Now for the other part. You can do it all with a template which will match only the relevant `<persName>` elements (`persName[starts-with(@ref,'#')]`). This time however, we replace the `<persName>` by an HTML `<span>` element, whose title attribute contains the text we want to display. That text is built up by concatenating relevant children of the `<person>` element whose xml:id has the same value as the ref attribute (or rather, the substring following its initial #, ie `substring-after(@ref, '#')` in XPath). You can get to the corresponding person with the select attribute `select="//person[@xml:id=substring-after(current()/@ref,'#')]"` on `<xsl:for-each>`
6. For extra points, you might like to add a final template which will just pop-up the text Don't know this person for any other `<persName>` element.