Exercise: Describing a Manuscript in TEI XML

March 2009

1 Before you start

This exercise is designed to further familiarise you with both the oXygen editing environment and the TEI manuscript description module by taking a very basic manuscript description and giving it some real structure.

2 Loading a document

Once oXygen has loaded and you've dismissed any helpful tips it wants to give you, load a file (File -> Open) by browsing to the your working directory and opening the file 'msDesc.xml'.

3 Our basic `<msDesc>`

As you will notice this contains a fully complete, valid, and well-formed manuscript description. Technically it validates against the ENRICH schema, which is a pure subset of the TEI designed for standardising manuscript descriptions internationally. However, this one consists only of the required elements and attributes! The msDesc element has not only the TEI namespace delcaration but the required xml:id and xml:lang elements. The msDesc contains inside it only two children, the `<msIdentifier>` element (with the required `<repository>`) and a `<p>` element containing the prose of the manuscript description from your first exercise. The materials directory also contains 'f101-19a.jpg' which is an image of this manuscript if you are curious.

4 Making a real `<msIdentifier>`

1. Begin by highlighting and deleting the `<repository>` element and its contents. To do this move the cursor to before the start of the `<repository>` tag, hold down the shift key, move to the end of the closing `</repository>` tag, and then press delete. When you do so notice that closing `</msIdentifier>` tag gets underlined in red, that a red line appears across from it on the right-hand bar, and there is a red box in the upper right-hand corner. These are all signs to you that there is an error and where it is. Hovering your mouse over the red box in the upper right-hand side of the editor will provide a tooltip pop-up indicating the nature of the error. Alternatively, there should be an error message in the status bar at the bottom. In this case oXygen is complaining because the ENRICH schema requires there to be either`<repository>` or a `<msName>`

2. Return to the paragraph and delete the phrase 'Stored in'.

3. Just after that highlight from the word 'Lithuania' through until the end of the word 'Department.' including the period. 'Cut' this text by pressing (usually) 'control-x', and then after moving back

up to the <msIdentifier> paste it by pressing 'control-v'. oXygen should still complain about you putting text here because it expects some structure

4. Highlight the word 'Lithuania', and then select the menu item 'Document -> XML Refactoring -> Surround with Tags' (note that the shortcut key for this is 'control-e'). This should pop up a window which allows you to choose any element valid at this point to surround this bit of text with. Type 'country' and notice how it completes the word as you type it. Pressing enter will inside the element around the text 'Lithuania'.

5. Delete the comma after Lithuania, press enter to move 'Vilnius' to a new line, highlight 'Vilnius', press 'control-e' and type 'settlement'. Press enter to insert this element.

6. Delete the comma and press enter to move the rest of the text down to a new line.

7. Use this same technique to add a <institution> element around the text: 'Lithuanian National M. Mazvydas Library'; and a <repository> element around 'Rare Book and Manuscript Department'. Delete any unnecessary commas or periods.

8. Your document should now be well-formed and valid. (There should be no red lines or error messages, and that box in the upper right-hand corner should be green. If that isn't the case, make sure you have deleted any stray punctuation in between the elements.

9. Move to just before the closing </msIdentifier> tag and press enter to insert a blank line inside it. Type the open angle bracket '<' and notice what happens. oXygen should provide you with a drop down list of elements which it is valid to insert at this point. Choose <idno> and notice that it inserts the element and leaves you inside the opening tag in case you wanted to add some attributes. Move down and cut and paste 'F101-19' in between the starting <idno> tag and its closing </idno> tag. Delete the stray colon that you did not cut and paste.

10. Your <msIdentifier> should now look something like:

```
<msIdentifier>
 <country>Lithuania</country>
 <settlement>Vilnius</settlement>
 <institution>Lithuanian National M. Mazvydas Library</institution>
 <repository>Rare Book and Manuscript Department</repository>
 <idno>F101-19</idno>
</msIdentifier>
```

and your document should still be well-formed and valid. If it isn't, correct it before continuing. When it is you can choose the menu item 'Document -> XML Document -> Format and Indent' to tidy up the indenting of the elements.

5 Creating an <msContents> element

1. Immediately after the closing </msIdentifier> tag, type an angle bracket: '<' to get oXygen to present a drop-down list of elements valid at this point. Select <msContents>. Notice that the <p> below is now underlined in red, because the ENRICH schema doesn't allow a vague pararaph and structural divisions to co-exist in this manner.

2. Move in between the opening and closing tags of the <msContents> element and again type '<' to see a list of possible elements. In this case choose <summary>.

3. Go to the paragraph down below and copy and paste everything from 'An original copy of a' all the way to 'Bug (Bugk) river.' into the <summary> element you have just created. This is a prose description of the intellectual content of the manuscript.

4. Press enter between the closing </summary> tag and the closing </msContents> tag and then add a new <textLang> element by typing '<'. Notice how oXygen (usually) automatically adds the required mainLang attribute and places your cursor and just the right place to type in the ISO language code 'lat' for Latin. Type it in.

5. Move to just after the closing double-quote of the mainLang attribute. Press space. oXygen should provide a drop down menu of all other attributes valid on this element. Scroll down and choose 'otherLangs', and since the descriptive paragraph below tells us it contains Polish as well as Latin, add the code 'pol' as the otherLangs attribute value.

6. Cut and paste the text 'Latin. There are postscripts in Polish.' into the middle of the <textLangs> element.

7. After the <textLangs> element but before the closing </msContents> tag, add in an <msItem> element by typing it in. Inside the <msItem> element create an <incipit> element in a similar manner. Cut and paste the text 'In nomine Domini amen. Ne orror obliuionis...' in between the <incipit> tags, including the final ellipsis.

8. While the document still isn't valid, because of that <p> down below, there should be no red lines inside your <msContents> which should look something like:

```
<msContents>
 <summary>An original copy of a charter where Alexander (Alexander), King
   of Poland and Grand Duke of Lithuania, transumes and confirms the
   privilege of January 26, 1380, by which Vytautas (Vitholdus), Grand
   Duke of Lithuania, gives permission to establish a monastery of the
   Augustinian Order in Brest (Bresth) and founds plots of land and
   Kostomoloty (Koszthomlothy) village near the Bug (Bugk)
   river.</summary>
 <textLang mainLang="lat" otherLangs="pol">Latin.  There are postscripts
   in Polish.</textLang>
 <msItem>
  <incipit>In nomine Domini amen.  Ne orror obliuionis...</incipit>
 </msItem>
</msContents>
```

6 Structuring a <physDesc>

Using the techniques you've learned above of wrapping an element around some text (control-e) or just starting to type the element in ('<'), separate the physical description information into a <physDesc> element and its possible children, something like below:

```
<physDesc>
 <objectDesc form="leaf">
  <supportDesc material="perg">
   <support>Parchment.  Single sheet format:  485 x 385 + 115 mm.</support>
  </supportDesc>
  <layoutDesc>
   <layout columns="1">Text:  375 x 225 mm.</layout>
  </layoutDesc>
 </objectDesc>
 <decoDesc>
  <decoNote>The ornamented initial "I" takes up 30 lines, 15 cm.</decoNote>
 </decoDesc>
 <sealDesc>
  <seal>
   <p>The lesser red wax attached seal of GDL (dia.  4 cm) in a black wax
bowl (dia.  8.5 cm) is affixed to the document with a string of rosy thread
(parchment fold-up is 11.5 cm, two slits for the string are spaced 8 cm
apart).</p>
  </seal>
 </sealDesc>
</physDesc>
```

Pay particular attention to the required attributes on some of the elements. These are not required in TEI generally, but here we are using the ENRICH schema, which holds us to a higher standard. Thus, if these elements are present, we must provide a form attribute on <objectDesc>, a material attribute on <supportDesc> and a columns attribute on <layout>.

7 The importance of <history>

Most of the text of the paragraph below should now be gone. Use the remaining text to create a <history> element containing the crucially important origin date and place (<origDate> and <origPlace>). Your history element should look something like:

```
<history>
 <origin>
  <origDate when="1502-06-24">[June 24], 1502.</origDate>
  <origPlace xml:lang="lat">Kamenec (Kamieniecz)</origPlace>
  <p>Authenticity - original.</p>
 </origin>
</history>
```

Here we have also added a paragraph stating that this is an original of the charter (rather than a contemporary copy).

8 Making this a real TEI file

What we need to do now is take the <msDesc> we've created and build a TEI file around it. This will also help us to understand how the description fits into the concept of a complete digital edition.

9 Wrapping it in <TEI>

1. Highlight your entire <msDesc> element using either the keyboard or the mouse, making sure **not** to include the processing instructions or xml declaration at the top.

2. Press (usually) 'control-e' to surround this with an element. Looking at the drop-down menu in the dialog box you should have an option for 'TEI#http://www.tei-c.org/ns/1.0'. Select it!

3. Notice that not only the <TEI> element but its namespace declaration has been added.

As you can see, although the file is well-formed, it is invalid because it doesn't validate against the schema. To be specific, in needs a <teiHeader> element, optionally a <facsimile> element and/or a <text> element.

10 Adding the <teiHeader>

1. Inside the <TEI> element, but before the <msDesc> element, add a <teiHeader> element

2. The <teiHeader> requires a <fileDesc>, add it

3. A <fileDesc> requires a <titleStmt>, <publicationStmt>, and <sourceDesc>. Add the <titleStmt> and inside that a <title> element. Add in a title for this document, perhaps 'F101-19: An electronic edition'. If you move outside the <title> and type '<' you can see a list of other elements allowed inside <titleStmt>, look at the options but delete the '<' before moving on.

4. Outside the <titleStmt> add a <publicationStmt> and also look at what elements are allowed here. Inside the <publicationStmt> add a <publisher> element with the content 'Lithuanian National M. Mazvydas Library'. The are other elements one could add here, or we could have just given this information in a <p> instead.

5. After the <publicationStmt> element, add a <sourceDesc> element to describe the source, inside put a single, empty, <p/> element. This is just a placeholder, we will replace it in a minute.

6. If everything has gone well, then you should have no red lines inside your <teiHeader> (though overall the document will still be invalid because we don't have a <text> element). It should look something like:

```
<teiHeader>
 <fileDesc>
  <titleStmt>
   <title>F101-19:  An electronic edition</title>
  </titleStmt>
  <publicationStmt>
   <publisher>Lithuanian National M. Mazvydas Library</publisher>
  </publicationStmt>
  <sourceDesc>
   <p/>
  </sourceDesc>
 </fileDesc>
</teiHeader>
```

Note that we haven't added any of the other (optional) aspects of the teiHeader that are not required. You may wish to explore these at some point.

11 Adding our <msDesc>

We now want to move our <msDesc> that we created earlier to the <sourceDesc>

1. Delete the empty <p/> element inside the <sourceDesc>
2. Highlight the <msDesc> you made and using the edit menu, or keyboard shortcuts, cut and paste it into the <sourceDesc> element
3. Your document should now have no red lines inside the <teiHeader> section of your document (some further down is fine for now).

12 Adding a <facsimile> element

We're not going to do much with the <facsimile> element, but it is good to include since this allows a way to reference graphics associated with the text. In our case we will just use the <graphic> element. (We could also have added this to our <msDesc> using a <surrogates> element inside the <additional> section.)

1. Go to after the closing </teiHeader> tag and insert a blank line.
2. Insert a <facsimile> element
3. Inside this, add an empty <graphic/> element
4. As part of the <graphic/> element add a url attribute with a value of 'f101-19a.jpg'. This could take any URL pointing anywhere on the internet, however in this case the graphic file is in the same place as the other materials.
5. Your <facsimile> element should now look something like:

```
<facsimile>
 <graphic url="materials/f101-19a.jpg"/>
</facsimile>
```

It is important to note that adding this <facsimile> element does not 'do' anything with regard to linking the text to the image or parts of the image. It is simply providing the information in a standard place so that in processing or rendering the document one can have access to it.

13 Adding the text

Your file should now be completely valid against the ENRICH schema at this point. If it isn't, try to find what it thinks is wrong and fix it before proceeding! Although you can make a valid TEI file with only a `<facsimile>` element to form an electronic facsimile and no `<text>` element, we do not tend to view these are real digital editions. To get improve this edition we must add a `<text>` element!

1. Move to just after the closing `</facsimile>` tag, and hit enter to add a new line
2. Add a `<text>` element
3. Inside this add a `<body>` element
4. Move in between the starting and closing tags of the `<body>` element and go to the menu item 'Document -> File -> Insert File', and select the file 'F101-19_text.xml'. This file contains the text of the charter already marked up in XML. This markup has been provided by someone else, and we might have encoded some things differently, see what you think.
5. Your file should be valid!

14 Save your file!

Just a reminder to make sure you have saved your file before you finish! Perhaps save it as 'exercise2.xml'.

15 Other things to try ...

1. Looking at the image of the manuscript, what things have we not included in this basic manuscript description?
2. What TEI elements might exist to record this information?
3. What TEI phrase level elements might you use to mark up the information further?
4. Explore the markup we've added by inserting the 'F101-19_text.xml' file. What is marked up, and what is not?
5. Experiment with adding more markup to the file.
6. Try out the oXygen 'Author' mode on this text. It has a customised CSS stylesheet which displays the manuscript description for editing.