Entities

Magdalena Turska   *April 2015*

1   Handy oXygen tricks

Enclose the selection with a tag:

- highlight the characters which you want to tag

- type CTRL+E to display the menu of available tags

- type 'pe' (for `<persName>`) or 'pl' (for `<placeName>`) and then press RETURN

Split the long chunk of text into a sequence of elements of the same kind:

- highlight the long chunk and wrap it with your desired element (say `<p>`)

- move the cursor to a place within that chunk that is the start of next element of the same kind

- type ALT+SHIFT+D to split elements (it inserts closing and starting tags at the cursor position)

- repeat as many times as needed

If you forget the key combination to perform the trick try the right-click and see what's in the Refactoring section

It can help a lot to format and indent your work automatically via CTRL+SHIFT+P or clicking the Format and indent icon.

2   Find your files

In this exercise, we will use the letter from Dantiscus' correspondence.

The aim is to encode:

- people & places mentioned in the text

- and the metadata about these entities

3   Marking Names

Now that we have the file, note there are people and places in them, so we should mark their names. Go through the document and any time you come across the name of a person, place, or organisation mark them up using `<persName>`, `<placeName>`, and `<orgName>` respectively.

You can do this quite quickly if you remember the oXygen trick of enclosing the selection with a tag with CTRL+E combination

4   Referencing strings

You might notice that there are mentions of named entities that not necessarily use names. Formulaic expressions like `secretario nostro` or `caesaream maiestatem` but also `she`, `my love` are good examples. In such cases you should use the `<rs>` element that otherwise behaves just like `<persName>` or `<placeName>`.

5 Identifying

Next you need to disambiguate the names and referring strings. `secretario nostro` for example is the same person as `Ioannes Dantiscus`. To do this, and for other reasons, you need to allocate unique codes to each unique person or place you identify. In a real life project, you would probably use special software for this purpose, but for the moment proceed as follows:

- Create metadata entry for each person or place in the teiHeader (how to on the next slide)

- Use the *@xml:id* of this entry as a value of *@ref* attribute on each `<persName>` and `<placeName>`. This value should be a pointer, starting with a # and followed by any code you cared to make up when creating the entry. You could use sequential numbering, (`pers01, perso2, place01, place02` etc), or you could follow our suggestion of using two initial letters taken from the name in question, followed by two digits.

- For example, if we give the queen 'Bona' the code `BS01` and we give the city of 'Cracovia' the code `CR01`

then wherever they appear in the text we need to use

```
<persName ref="#BS01">Bona regina</persName> or
<rs ref="#BS01">nostra regina</rs>
```

6 Adding Metadata for Places

In our `<teiHeader>` we're able to store metadata about the people, places, and organisations mentioned in the text. The location for storing these is inside the `<profileDesc>` element we added to the header. For places we add a `<settingDesc>` and for people and organisations we add a `<particDesc>`.

- 1. After the closing `<langUsage>` inside `<profileDesc>` add a `<settingDesc>` element and inside that a `<listPlace>`. (Technically the `<listPlace>` is unnecessary, but I think it is a good habit as it allows you to group related places.)

- 2. Inside the `<listPlace>` add a `<place>` element with an *@xml:id* attribute of 'CR01'. This is an arbitrary ID number based on the first four letters and two incremental digits – obviously if we were dealing with even more places we'd come up with something more robust.

- 3. Inside this first place add a `<placeName>` (Cracovia), a `<region>` (Lesser Poland) and a `<note>` ('Cracovia' was a capital city of the Kingdom of Poland in Jagiellonian times.)

- 4. Add another place next to the first one in a similar manner

- 5. Your `<settingDesc>` might now look something like:

```
<settingDesc>
 <listPlace>
  <place xml:id="CR01">
   <placeName>Cracow</placeName>
   <placeName>Cracovia</placeName>
   <region>Lesser Poland</region>
   <note>'Cracovia' was a capital city of the Kingdom of Poland in Jagiellonian
       times. Situated on the Vistula River the city dates back to 7th
       century.</note>
```

```
     </place>
   <place xml:id="HE01">
    <placeName>Heilsberg</placeName>
    <placeName>Lidzbark Warminski</placeName>
    <region>Warmia</region>
    <note>Town of Old Prussian origins, formerly the capital of Warmia and its
        largest city.</note>
   </place>
  </listPlace>
 </settingDesc>
```

7 Adding Metadata for People

- 1. After the closing `<settingDesc>` inside `<profileDesc>` add a `<particDesc>` element and inside that a `<listPerson>`.

- 2. Let's start with one of the people mentioned that we know most about, Ioannes Dantiscus. Add a `<person>` element with an *@xml:id* of 'ID01'. Inside this we can add all sorts of information about the person. Let's start with a persName, with a *@xml:lang* attribute of 'la', containing 'Ioannes Dantiscus'. The reason we're saying that this is Latin version of the name is he was known under many variants of his name, depending on the country: Johannes Flaschbinder (or von Höfen) to German speakers and Jan Dantyszek to Poles.

- 3. Next to this persName(s) we could add a birth element with a date element containing '1 November 1485' and a @when attribute. Inside this birth we can also provide a placeName of 'Danzig'. Yes, we could (and should) add this place to our `<listPlace>` in `<teiHeader>` if we wanted.

- 4. Let's note our protagonist's occupation and other information we might fish from available resources.

- 5. If you're really dilligent your first person might now look something like:

```
<person xml:id="ID1485">
 <persName xml:lang="la">Ioannes Dantiscus</persName>
 <persName xml:lang="de">Johannes von Höfen</persName>
 <persName xml:lang="pl">Jan Dantyszek</persName>
 <persName xml:lang="de">Johannes Flachsbinder</persName>
 <persName xml:lang="la">Ioannes de Curiis</persName>
 <birth notBefore="1485-01-01"
  notAfter="1485-12-31">1485</birth>
 <death when="1548-10-27">†1548-10-27</death>
 <occupation>diplomat, neo-Latin poet and traveller</occupation>
 <occupation when="1504">1504 royal scribe</occupation>
 <occupation when="1507">1507 referendary for Prussian affairs at the court of
   Sigismund Jagiellon; </occupation>
 <occupation from="1508" to="1513">1508-1513 royal envoy to Prussian towns and to
   the Prussian assemblies;</occupation>
 <occupation from="1515">1515 secretary of the Polish legation at the imperial
   court; </occupation>
 <occupation from="1516" to="1532">in 1516-1532 envoy in the service of the king of
    Poland Sigismund Jagiellon and emperors Maximilian and Charles V of Habsburg;
</occupation>
 <event when="1529">Kulm canon; </event>
 <occupation from="1530" to="1537">1530-1537 bishop of Kulm; </occupation>
 <occupation from="1537" to="1548">1537-1548 bishop of Ermland</occupation>
</person>
```

Add entries for other people

If we won't bother to look up much information about the other people they will go a lot quicker!

8 Linking Names and Metadata

Having marked all these names, and created metadata about them, it seems a shame not to link the names to the metadata. So let's do that!

- 1. Go to the first persName you marked – probably that of Ioannes Dantiscus shown above. Move the cursor into tag name just before the closing '' and press space. oXygen should prompt you with a list of attributes allowed at this point. Add the @ref attribute and when you do you should get a drop down list of all the @xml:id values in which the value '#ID01' should appear. Select it!

- 2. Continue on and for each `<persName>`, `<placeName>`, and `<orgName>` (for which there is a `<person>`, `<place>` or `<org>` element) go through and add a *@ref* attribute pointing to the correct *@xml:id* or add the necessary entries to `<listPerson>`, `<listPlace>` or `<listOrg>` first.

- 3. The value of *@ref* is a URI, which includes URLs, and in this case a 'fragmentary URL'. It starts with a '#' to let us know that the place it is pointing to is in the same document. You could also have stored the listPerson in a separate document, in which case we would put something like 'people.xml#ID01', or stored this online somewhere 'http://www.example.com/people.xml#ID01'. This makes more sense if you are encoding many documents which might involve the same people, places, or organisations. While it is best if this points to a TEI person element, it can in fact point to anything which documents the name, such as a wikipedia article. (One reason it is better for this to point to a person element is that inside that you could indeed point to more than one external source of information, and change this is one place when the resources change.)

- 4. The benefit of an encoder doing all this work is that for each instance of a name someone processing the text could find a standardised form of it, and other metadata, when generating other outputs. (e.g. for help in searching, linking, or displaying this information)

- 5. One of the things we've not done is mark all the names mentioned in the metadata itself and have them point to their person records. While this would be a good idea if we were generating sophisticated output from this metadata, we probably don't need to do that for this exercise.

9 Another person example

'Bona' refers to Bona Sforza, the queen of Poland and duchess of Lithuania and Bari, wife of the king Sigismund I Jagiellon (you can read about her on wikipedia at http://en.wikipedia.org/wiki/Bona_Sforza) who exchanged almost 300 letters with Dantiscus.

```xml
<listPerson>
 <person xml:id="BS01">
  <persName>Bona Sforza</persName>
 </person>
 <person xml:id="ID01">
  <persName>Ioannes Dantiscus</persName>
 </person>
</listPerson>
```

Try to create at least minimal `<person>` and `<place>` elements for each of the different people and places whose names you have tagged. Of course, it's up to you how much time you spend researching these named entities and transferring the information you find into TEI form! Use elements such as `<birth>`, `<death>`, `<occupation>`, `<event>` to record for example these facts about Bona Sforza, which we have copied from Wikipedia:

```
   Bona Sforza (2 February 1494[1] or 2 February 1493[2] — 19 November 1557) was a
member of the powerful Milanese House of Sforza. In 1518, she became the second wife
of Sigismund I the Old, the King of Poland and Grand Duke of Lithuania, and became
the Queen of Poland and Grand Duchess of Lithuania.

She was the third child of Gian Galeazzo Sforza and his wife Isabella of Naples.[3]
Her older brother was Francesco Sforza and her sisters were Ippolita Maria and Bianca
Maria. All of Bona's siblings died young.

When her mother Isabella of Naples died in 1524, Bona succeeded to the titles
Duchess of Bari and Princess of Rossano. She also became the holder of the Brienne
claim to the title of King of Jerusalem.
```

10 We still need more!

We have a nice TEI XML document which a) marks up names and places, and b) provides a list of disambiguated people and places.

How can we improve it

- We can geolocate all the places in it

- We can put in information which can be used to create linked data

some of these are impractical today, so lets concentrate on three simple tasks.

11 Find locations for your places

The task is to edit your file, and add a) `<location>` with `<geo>` elements, and b) `<idno>` elements to each `<place>`. Use geonames.org, or some other source (eg Google Maps), to locate your places. Use simple point location. For `<idno>` elements try VIAF: The Virtual International Authority File (viaf.org)

12 In real life

Most probably in real-life project there will be many TEI documents that should not repeat all the shared metadata about people and place. In such cases you might want to have one 'master' file for all people etc.

What to do then? Instead of putting `<listPerson>` &co in the teiHeader just put in `<include>` element in its place, that points to a file that has `<listPerson>` as a root.

```
<profileDesc>
 <particDesc> <include xmlns="http://www.w3.org/2001/XInclude" href="people.xml"/>
 </particDesc>
</profileDesc>
```

Of course, this requires moving all your metadata into separate file. Remember to make root element of that file belong to TEI namespace!

13   Self-Assessment

Check if you understand some of the core principles of this exercise by answering the following questions:

- Which elements are used to mark personal, place, and organizational names?

- How do you store metadata in the header about the entities these names refer to?

- What values does the @ref attribute allow? How can this be used to point to external files or URLs?

- How do you mark up strings of text which reference named entities, but aren't names themselves?

14   Save often!

Don't forget to save the file you have created!