XML Data Management **Fall-Winter 2011/12**

Coursework **Werner Nutt**

Coursework C3: XSLT

For this piece of coursework you are asked to use XSLT to express natural language queries and transformations over documents with information about movies. The queries or transformations should be formulated over the same DTD as the queries for the last coursework on XQuery. You find both, the DTD and an example document, on the course website.
Again, you can develop and test the queries using the Kernow front end to Saxon. Note that to run Kernow under Linux, you need the official Java JDK from the Java website at Oracle.

Hint: Look at the XSLT sample solutions on the course website to understand how to write your stylesheets.

Comment: This is an extra piece of coursework. If you complete it, it will raise the percentage of the coursework that counts towards the final mark to 40%.

Movie Transformations

1. Return an element "`movie_titles`" that contains a list of alphabetically sorted titles of all movies in the document.

   Write two stylesheets, one where iteration is expressed by template calls and another one where iteration is expressed by `<xsl:for-each>` elements.

2. Return an element "`movies`" with a list of "`movie`" elements, containing in turn attributes title and year, ordered by year in descending order. Use static element and attribute constructors.

3. Rewrite the preceding stylesheet so that you use copying and dynamic element and attribute constructors wherever possible.

4. Restructure the movies document so that

   - movies appear according to their year, with the most recent years first, and, within the same year, according to their title
   - countries appear in alphabetical order

- for each director, the first name is given before the last name
- actors of a movie appear according to their last name and, for actors with the same last name, according to their first name.

Everything else should remain as before.

Hint: To sort according to two keys, say, first according to `key1`, and in case of equal `key1`-values according to `key2`, write two `<xsl:sort>`-elements:

```
<xsl:sort select="key1"/>
<xsl:sort select="key2"/>
```

5. Return an element "`countries`" containing an alphabetically sorted list of "`country`" elements, where each country from the document occurs exactly once.

   Use the two approaches shown in the lecture.

6. Output movies (title and year) and within each movie those actors whose role occurs in the summary of the movie.

7. Return an element "`actors`" with a list of "`actor`" elements, alphabetically sorted by last name and first name, where each actor from the movies document occurs exactly once.

   Each actor element should contain last name, first name, and year of birth of the actor.

   Hint: You may need an `<xsl:variable>` element to remember the last name when you sort according to first name.

8. Create a stylesheet that produces an HTML layout of the movies document. Follow the example of the recipes layout from the lecture.

Deliverable

For this coursework, you can work in pairs. Your deliverable will consist of

- a `.txt` file that contains for each query your formalization in XQuery.

The file `movie-transforms.txt` on the website contains a numbered list of all queries/transformations. Write your answers into that file, each answer below the corresponding question.
Please, submit your work by email to `nutt` AT `inf.unibz.it` no later than

Fri, 17 February 2012, 23:30 hours.