TÉCNICO LISBOA

Gestão e Tratamento de Informação

1st Project

Deadline at 24 Oct. 2014 :: Online submission at IST/Fénix

The XML document available at https://fenix.ist.utl.pt/downloadFile/3779580002803 contains a data sample of transcripts for parliament interventions, taken from the Portuguese *Assembleia da República*, together with profile information regarding the Portuguese politicians involved in these parliamentary discussions. Figure 1 illustrates the contents of the XML document in the above URL.

```
<?xml version='1.0' encoding='UTF-8'?>
<parliament xmlns="http://www.parlamento.pt">

  <parliament-interventions>
   <session date="2014-03-01">
    <speech order="1" politician="1">Text from the speech of a given politician.</speech>
    <speech order="2" politician="2">Reply to the 1st speech from another politician.</speech>
    <speech order="3" politician="1">Reply to the previous reply.</speech>
    <speech order="4" politician="3">Another intervention from a politician.</speech>
    <speech order="5" politician="2">Reply from a given politician.</speech>
    <speech order="6" politician="1">Reply to the previous reply.</speech>
   </session>
   <session date="2014-03-02">
    <speech order="1" politician="1">Intervention from a politician.</speech>
    <speech order="2" politician="2">Speech about "ensino superior" and about "educação".</speech>
    <speech order="3" politician="1">The reply from a second politician.</speech>
   </session>
   <session date="2014-04-02">
    <speech order="1" politician="3">A speech given by a particular politician.</speech>
   </session>
   <!-- list of remaining parliament sessions/speeches -->
  </parliament-interventions>

  <politicians>
   <politician code="1" party="PSD" age="50">Pedro Passos Coelho</politician>
   <politician code="2" party="PS" age="52">José Seguro</politician>
   <politician code="3" party="PS" age="65">João Soares</politician>
   <!-- list of remaining politicians -->
  </politicians>

</parliament>
```

Figure 1 : Sample from the XML document used in the exercises.

Exercise 1

1.1 - Create an XML Schema for validating an XML document with information regarding politicians and parliamentary interventions, similar to that from Figure 1. When developing the XML Schema, take the following particular aspects into consideration:

- Ensure that the developed XML Schema uses global data types, instead of local type definitions encapsulated within the elements, in order to promote modularity and code re-usage.
- Ensure that the data types for the contents of attributes named *order* and *age* correspond to positive integer numbers.
- Ensure that politician names start with an uppercase letter, and that they are composed of at least two separate words.
- Ensure that the attribute named *code* is of mandatory occurrence in the elements named *politician*, whereas the attribute named *politician*, in the elements named *speech,* is optional.

- Ensure that the elements descending from *parliament* (i.e., the elements named *parliament-interventions* or *politicians*) can appear in any order (e.g., the list of politicians may appear first in these XML documents).

1.2 - Extend the XML Schema from the previous exercise in order to take referential integrity into account, when validating the document. Specifically, verify if the *politician* attributes used in the elements named *speech*, refer to politicians that exist within the XML document.

Exercise 2

Create an XML Transformation (i.e., an XSLT document) for representing the politicians listed in the XML document from the previous exercise, as an XHTML document that can be displayed in a Web browser. The developed XSLT should use structural recursion to process the input XML document, including at least two different templates.

The resulting XHTML document should contain one table, where rows correspond to the different politicians. This table should include six separate columns, corresponding to the politician's code, its party, its age, its name, the number of parliament interventions where the politician was involved, and the number of sessions with interventions by the politician.

The table should also include a header, listing the names for the different columns.

Exercise 3

Create XPath expressions for addressing each of the following information needs.

1. Find how many different politicians intervened in the session that took place during the month of March 2014.
2. Find the names for politicians that have made interventions regarding the subject of *ensino superior* or *educação* (i.e., interventions where those particular expressions have been used).
3. Find the names of all politicians that have made interventions replying to the politician named *José Seguro* (i.e., the names for politicians that have made interventions in a same session and with an order that corresponds to exactly an intervention after another from the politician named *José Seguro)*.
4. Find the average age of the politician(s) that have made interventions in the session from 2014-03-01, after the first intervention by a politician from the PSD political party at that same day.

If required, your XPath expressions may assume that (i) sessions are always ordered chronologically in these XML documents, and (ii) speeches are always ordered according to the *order* attribute in these XML documents.

Exercise 4

Create XQuery FLWOR expressions for addressing each of the following information needs. Although XPath expressions can be used as part of each answer, note that in this exercise you should always write the queries using XQuery FLWOR expressions, eventually also using XQuery updating expressions whenever the exercise involves updating the XML dataset.

1. Find the name(s) of the politicians(s) who have made more than two interventions replying to the politician named the *José Seguro,* together with the corresponding party name(s).

2. Find the top five longest parliament interventions (i.e., the ones containing more words in their contents), showing the contents of each of these interventions according to the XML format that is shown next.

```
<interventions>

     <intervention session-date="2014-01-01"
                   party="PS"
                   politician="José Seguro"
                   rank-order="1">
       Message from a politician.
     </intervention>

     <intervention session-date="2014-01-02"
                   rank-order="2"
                   party="PS"
                   politician="José Seguro">
       Message from a politician.
     </intervention>

     <!-- remaining interventions in the list of top five -->

</interventions>
```

Figure 2 : Example of the format produced as output by the XQuery expression.

3. Change the XML dataset in order to explicitly store:

   a) An attribute named *num-interventions,* associated to the *politician* elements, and encoding the number of parliament interventions where the politician was involved;

   b) An attribute named *num-sessions,* associated to the *politician* elements, and encoding the number of sessions with interventions by the politician;

   c) An attribute named *most-frequent,* associated to the *session* elements, and encoding the party that has made more interventions in the session.

   Notice that you should use a single XQuery expression for performing all modifications.

4. Change the XML dataset so as to remove all politicians with less than 3 interventions, removing also all interventions from these less-active politicians.

Exercise 5

Consider the following two tree structures (see Figure 3), each representing an XML element encoding information about a politician named *José Seguro*.

Compute the similarity (i.e., the number of matching nodes) and the alignment between both trees in Figure 3, using the *Simple Tree Matching* algorithm, and considering that two nodes can be aligned if they share the same name/content.

Present all the calculations involved in finding the number of matching nodes in the trees. Present also the alignment between the trees, with basis on the results from the *Simple Tree Matching* algorithm.

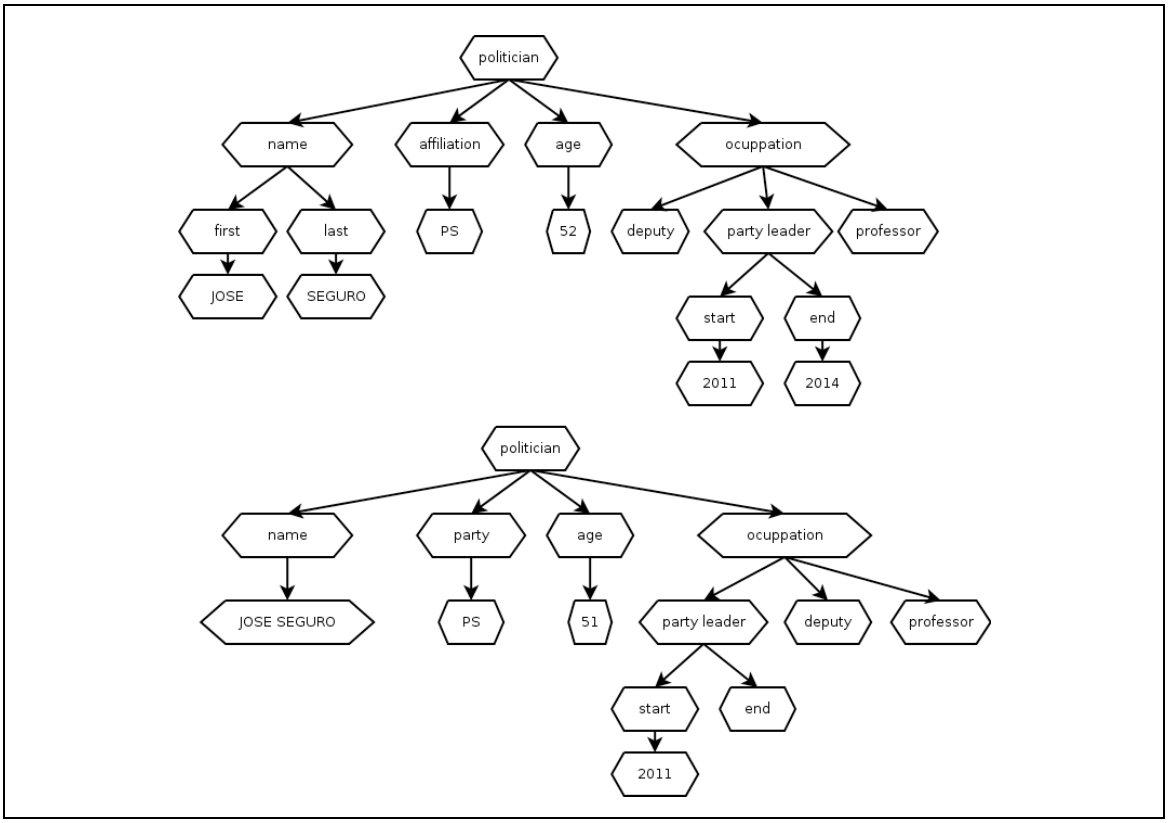

Figure 3 : Two example tree structures.

Submitting the project

The solutions to the project should be submitted through the *Fénix* system, in the form of a .zip file containing a PDF report with the solutions for the exercises, as well as individual text files with the solutions for each of the exercises (i.e., documents with the XML, XSD, XSLT or XPath/XQuery code).

In the course Webpage, you can find a Microsoft word template for the project report.

In the theoretical class following the electronic submission, a printed copy of the report, with the solutions for each exercise, should also be delivered.

We will not accept deliveries through e-mail, with reports not conforming to the supplied template, or without the text files with the solutions for the exercises.

Good luck!