

XML Technologies: Exercise 8

XML Design, RDF

This is a mandatory exercise and the result will be part of your final mark. The solution must be uploaded to OLAT by **May 20th at 15:59**. Late submissions will not be accepted.

Submit the following files in a zipped archive:

- design.txt
- rdf.xml

Make sure the archive is named [lastname]_[firstname].8.zip (for example *mueller_mathias.8.zip*). Some parts of your submission may be automatically evaluated, so make sure to name your files *precisely* as prescribed, otherwise you might not get any points.

1 XML Design

After seeing many XML technologies put to good use (among others: DTD, XML Schema, XPath, XSLT) and since the semester is coming to an end, it is now time to take a step back. When looking at the very basics of XML, we intentionally skipped one crucial stage – that is, how XML documents are designed in the first place.

XML design is a crucial step because any XML technology *builds on top of it*. In other words, bad design has hardly anything to do with an intrinsic badness, it should be understood as hindering the processing of documents (navigation, query, transformation). In most cases, it's not that the design is bad *per se*, but, considering the purpose it should serve, it is disadvantageous.

As a consequence, it is absurd to think about the design of an XML document if you have no idea what it (i.e. the document) is going to be used for. So, defining the purpose of XML actually precedes its design.

1.1 Intelligent Design (2 points)

Imagine the following. You are working for a software company and currently in charge of overseeing the design of a new kind of XML document. It should serve the following purpose:

Represent information on countries. Specifically, the data should include the country and continent codes, the current population, the area it covers and last year's GDP. Several applications will access this document to retrieve information on specific countries. Therefore, e.g. retrieving the population of one specific country should be a trivial operation. On the other hand, given a GDP value, it should be simple to determine the respective country.

Two members of the team you are leading have volunteered to devise a design (sample data taken from <http://countrycode.org/>). Both drafts are now subject to your thorough scrutiny.

Draft A

```
<?xml version="1.0" encoding="UTF-8"?>
<countries>
  <country code="AO" continent="AF">
    <population>12799293</population>
    <area>1246700</area>
    <gdp currency="USD">110.3 Billion</gdp>
  </country>
  <country code="BE" continent="EU">
    <population>10414336</population>
    <area>30528</area>
    <gdp currency="USD">389.3 Billion</gdp>
  </country>
  <country code="ID" continent="AS">
    <population>240271522</population>
    <area>1904569</area>
    <gdp currency="USD">914.6 Billion</gdp>
  </country>
  <country code="MX" continent="NA">
    <population>111211789</population>
    <area>1964375</area>
    <gdp currency="USD">1.563 Trillion</gdp>
  </country>
  <country code="ZA" continent="AF">
    <population>49052489</population>
    <area>1219090</area>
    <gdp currency="USD">491 Billion</gdp>
  </country>
</countries>
```

Draft B

```
<?xml version="1.0" encoding="UTF-8"?>
<facts>
  <fact key="country" value="AO"/>
  <fact key="continent" value="AF"/>
  <fact key="population" value="12799293"/>
  <fact key="area" value="1246700"/>
  <fact key="gdp" value="110.3_Billion" currency="USD"/>
```

```

<fact key="country" value="BE"/>
<fact key="continent" value="EU"/>
<fact key="population" value="10414336"/>
<fact key="area" value="30528"/>
<fact key="gdp" value="389.3_Billion" currency="USD"/>
<fact key="country" value="ID"/>
<fact key="continent" value="AS"/>
<fact key="population" value="240271522"/>
<fact key="area" value="1904569"/>
<fact key="gdp" value="914.6_Billion" currency="USD"/>
<fact key="country" value="MX"/>
<fact key="continent" value="NA"/>
<fact key="population" value="111211789"/>
<fact key="area" value="1964375"/>
<fact key="gdp" value="1.563_Trillion" currency="USD"/>
<fact key="country" value="ZA"/>
<fact key="continent" value="AF"/>
<fact key="population" value="49052489"/>
<fact key="area" value="1219090"/>
<fact key="gdp" value="491_Billion" currency="USD"/>
</facts>

```

Both drafts can also be found in the exercise zip folder, as `draft_a.xml` and `draft_b.xml`.

Study drafts A and B very closely. Decide which of the drafts is better suited for the task, given the requirements stated above. Do not rely on your intuition.

Instead, substantiate your claim with XPath expressions that contrast the effort necessary to retrieve

- the population of one specific country
- a country, using its GDP value as an identifier

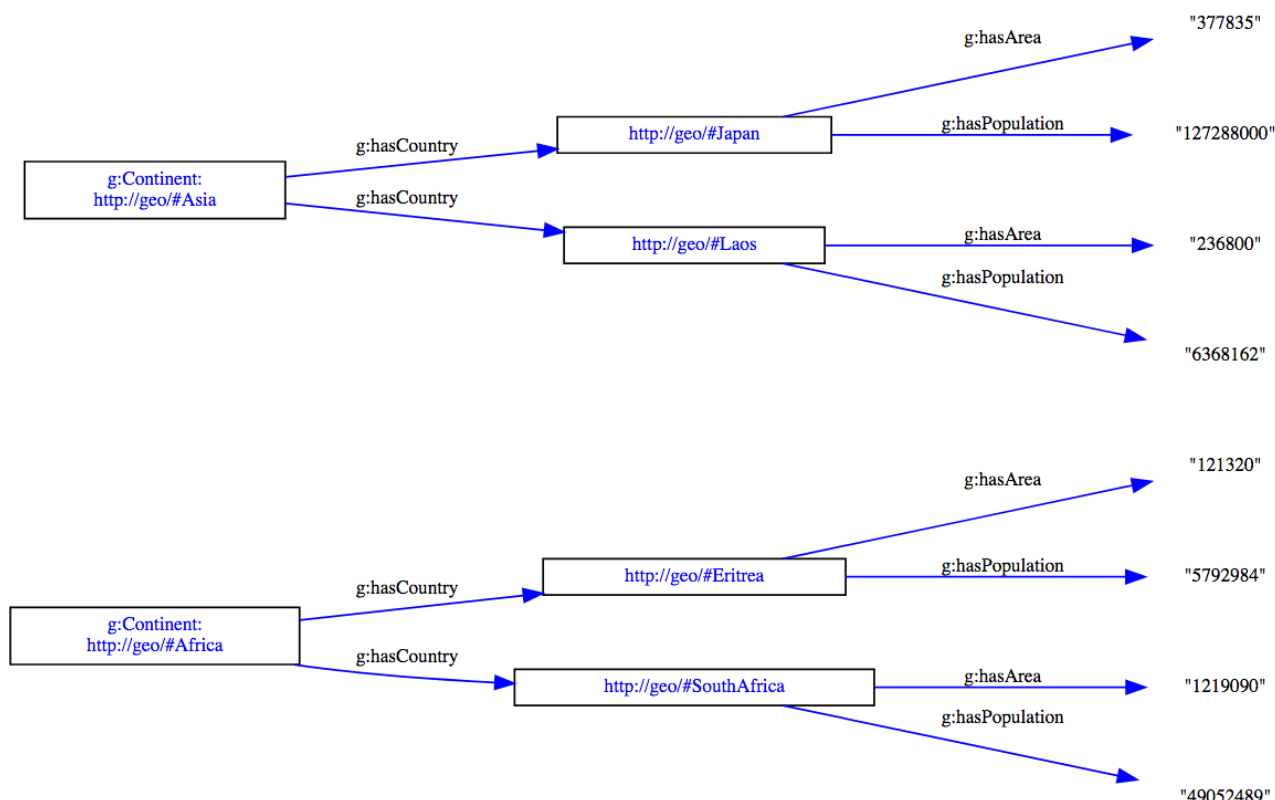
given either draft A or B as the query document. Be as clear as possible, your bosses never underwent proper XML training.

Save your findings as `design.txt`. Include your choice of draft, an explanation in complete sentences and suitable XPath expressions.

2 RDF (3 points)

The RDF standard is one of the most serious attempts to semantically enrich the web to date. It is meant to represent any kind of knowledge: facts, concepts and relationships. RDF is not limited to XML syntax, XML just happens to be one of the *serialisation formats* of RDF. Another form of serialising an RDF network of knowledge is Notation3 (N3).

Yet another way of displaying RDF is a graph, where serialisation is not an issue, since all information can be shown simultaneously.



Nodes in the graph represent objects and types, while arrows (predicates) indicate a relationship between nodes. As you can see, the kind of information displayed is similar to question 1 above. The graph is also available attached as `render.png`.

Write a serialised version of the graph above that represents all information in XML RDF syntax. Declare two namespaces, one for RDF elements:

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

Another one for the domain-specific facts:

```
xmlns:g="http://geo/#"
```

The beginning and end of your document should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:g="http://geo/#"
  <!--Your code here-->
</rdf:RDF>
```

Use the official RDF XML validator¹ to validate your solution before submitting it. Also, the validation service is useful to learn how RDF translates to either model triples or graphs. Save your RDF as `rdf.xml`.

¹Online at <http://www.w3.org/RDF/Validator/>.