Practical Lab 1

Web Data Models

September 18th, 2017

The goal of this lab session is to practice checking whether an XML is well-formed and valid in regard to a DTD, and to parse an XML using DOM/SAX and extract useful information from it.

1 Checking XML

Consider the following XML, called `tp.xml`:

```
<?xml version="1.0" encoding="utf-8">
<contributors>
  <contribution cname="DarkAngel80" cname="TryHarder">
    <text date="12-09-2016">The necessary threshold is <20 & >10
    <status>Approved<status>
    </text>
  <contribution cname="HelloKitty" />
  <contribution cname="Anonymous">
    <text date="13-09-2016">The revised text is here</text>
    <text date="14-09-2016">The second revised text is here</text>
    <status>Reviewed</status>
    <assignedTo>HelloKitty</assignedTo>
  </contribution>
</contributors>
```

Question 1. Give *all* the reasons why the XML above is not well-formed. Now, check this via the `xmllint` Unix command, by typing:

```
xmllint -noout tp.xml
```

If the XML is well-formed, the command will not output anything. If, on the other hand, the XML is not well-formed, an error will be output.

Question 2. Fix the above XML – write another XML, called `tpfixed.xml` that is well-formed and contains the same information as the original XML. Again, check it with `xmllint`.

Question 3. Draw the tree representation of the XML document in `tpfixed.xml`.

Question 4. Create a DTD file which validates your `tpfixed.xml`. Call it `tp.dtd`. Validate it using `xmllint`:

```
xmllint -noout -dtd-valid tp.dtd tpfixed.xml
```

2 Parsing XML

In this section, we will use DOM or SAX to parse an input XML file and extract information from it.

Question 1. Read the Java tutorials about DOM at `https://docs.oracle.com/javase/tutorial/jaxp/dom/readingXML.html` and about SAX at `https://docs.oracle.com/javase/tutorial/jaxp/sax/parsing.html`. In the following, it is your choice what type of XML parser you use.

Question 2. Open the OpenStreetMap website at `http://www.openstreetmap.org/`, and select a zone in the world (by zooming in and out). Then, export it using the *Export* button on the webpage. You will obtain a file called `map` in the OSM XML format. Take the time to familiarize yourself with the OSM XML format by looking over the file and consulting `http://wiki.openstreetmap.org/wiki/OSM_XML`.

Question 3. Use a Java DOM or SAX parser to load and parse the `map` XML file. Use the program to identify and print all the names appearing in the map.

Question 4. For each place name identified at Question 3, extract its Wikipedia short description, using the Wikipedia API by reading the URL `https://en.wikipedia.org/w/api.php?format=xml&action=query&prop=extracts&exintro=&explaintext=&titles=YOURTITLE`, where `YOURTITLE` is the place name you extract. Identify and print the short description in the resulting XML.