Lab 5: XPath Practice

Due date: Friday, November 19, beginning of lab period.

Lab Assignment

The purpose of this assignment is for you to learn XPath. XPath is an XML path expression language. The course project will have you develop an XPath processor on top of the native XML index structures you have built. This requires some preliminary knowledge of XPath.

In the lab you will use eXist, an open-source native XML DBMS, which includes full XPath (and XQuery) support. You will download eXist, install it (it is distributed as a Java .jar file), and will work with it in interactive mode.

This is an **individual lab**: each student is expected to submit his/her own solutions. Only one aspect of this lab, pertaining to data preparation, can be done in teams.

The Tasks

There are two basic tasks in this lab: (1) download, installation and learning your way around eXist and (2) design of XPath expressions for XML data and information needs provided below. Additionally, each team will construct a single XML repository file out of its Lab 2 XML files.

Preparation of data

The lab will use four datasets: Nutrition, WiiGames, Horses and Seinfeld. Each dataset shall consist of a single XML file, `NutritionRepository.xml`, `WiiGamesRepository.xml`, `HorsesRepository.xml` and `SeinfeldRepository.xml`.

Each of the repository files is organized as follows. The root of each file is an XML element `<repository>`. The children of the `<repository>` element are the top-level objects from the Lab 2 XML files for each respective team:

- For the Nutrition repository, the children of the <repository> element are the <nutrition> elements corresponding to the nutrition information described by the Poccnn Number 1 team in Lab 2.

- For the WiiGames repository, the children of the <repository> element are the <WiiGame> elements corresponding to descriptions of individual Wii games created by Team Too Hot in Lab 2.

- For the Horses repository, the the children of the <repository> element are the <horse> elements describing the horse statues from Lexington, KY, described by Team Equestrian in Lab 2.

- For the Seinfeld repository, the the children of the <repository> element are the <season> elements describing the seasons of the TV show Seinfeld, as created by Team USA in Lab 2.

Each group shall prepare the appropriate repository file by merging the content of all their Lab 2 XML files as described above and renaming the root elelemt where appropriate. The resulting files shall be loaded onto the course wiki. Links to the files shall be placed on the front page of the wiki, to ensure that everyone has easy access to the files. Feel free to add a link to the raw file (`"?format=raw"`) in addition to the regular link. Once the links are posted, I will also put them on the Lab 5 data page.

eXist Installation

eXist is an open-source light-weight native XML DBMS. It is distributed in the from of a Java .jar file.

eXist comes with a variety of access modes: client-server architecture, embedded mode (eXist provides API that can be directly queried from Java code) and interactive mode. In this lab, you will be using eXist's GUI client application, and will perform all actions within it.

eXist implements XQuery as the main method of access to stored XML data. XPath is the subset of XQuery designed to provide access to XML data. In the class we discuss XPath version 1.0, and this is the language you will be using in this lab.

Prior to start work on XPath you need to:

1. Download eXist. On CSL machines running Linux or on Macs, you will be using eXist version 1.4.0, revision 10440 shiped as a .jar file `eXist-setup-1.4.0-rev10440.jar`.

   This file can be obtained

   - from the Lab 5 data page:

     http://www.csc.calpoly.edu/∼dekhtyar/468-Fall2010/labs/lab05.html

   - from the Sourceforge eXist page:

http://sourceforge.net/projects/exist/files/Stable/1.4/eXist-setup-1.4.0-rev10440.jar/download

On your own PCs you will be using the `.exe` version of the eXist DB version 1.4.0 revision 10440. You can download it from either the lab page:

http://www.csc.calpoly.edu/∼dekhtyar/468-Fall2010/labs/lab05.html

or from the Sourceforge eXist page:

http://sourceforge.net/projects/exist/files/Stable/1.4/eXist-setup-1.4.0-rev10440.exe/download

2. Install eXist. Run the downloaded .exe file or the .jar file. Select necessary installation options (you do not need sources and javadocs, but feel free in include them). Select the admin password when prompted.

3. Run eXist. eXist comes with a client-server setup. On CSL machines, to run the server change to the directory in which eXist is installed and enter the command

   ```
   bin/startup.sh &
   ```

   To run the client, enter

   ```
   bin/client.sh&
   ```

   To shut down the server, enter

   ```
   bin/shutdown.sh
   ```

   On Windows systems, you can use `bin`
   `startup.bat`, `bin`
   `client.bat` and `bin`
   `shutdown.bat`, or you can find eXist programs in your system's "Start" menu.

4. Become familiar with the client's interface. You need to learn how to

   - Load an XML file from disk.
   - Browse an existing XML file.
   - Enter query mode.
   - Enter queries and observe answers[1].
   - Exit program.

[1]Note, that since the project involves implementing a(n almost) proper subset of XPath, you can use eXist later on for debugging purposes - it will allow you to establish the correct answer to any XPath query.

XPath queries

Each file loaded on eXist is treated as a separate (unique) XML document/repository. Because eXist supports handling multiple documents, any XPath expression must be prefaced with the designation of the XML document over which this expression shall range. This is done in eXist using XQuery's built-in `document()` function. The format of a full XPath expression understood by eXist is thus:

```
document(<documentName>)/<Relative-XPath-Expression>
```

Here, `<documentName` is the name of the document — typically the name of the file from which the XML document was read (in double quotes). `<Relative-XPath-Expression>` is any relative XPath expression.

We use four XML files for this lab. You need to load each of them as a separate document into eXist's database. Please take some time to study the structure of the XML files — without knowing it, it may be hard to compose XPath expressions.

NutritionRepository.xml

For this file, you need to write XPath expressions retrieving the following information:

1. Report all ingredients of miso soup.
2. Report the names of all products that have Vitamin A in them.
3. Report the names of all products where one serving contains less than 110 calories.
4. Find the product(s) which conatin(s) exactly four ingredients. (retrun the entire XML element for this product)
5. Report the first three ingredients for each fat-free product.

WiiGamesRepository.xml

1. Report the names of all Wii games produced by Activision.
2. Report the names of all Wii games which can be played with four or more different Wii controllers.
3. Report the company name for each game that can be played using a Nunchuck controller.
4. For every game rated for everyone ages 10 and above, report the descriptipion.
5. Report the names of all games described in the repository after Guitar Hero V.

HorsesRepository.xml

1. Report the names of the first 10 horse sculptures.
2. Report the information about each scuplture created by more than one author.
3. Report the information about the scuplture created by Betteye Brookfield.
4. Report the author and the location of the sculpture described the last in the XML document.
5. List the sponsors of horse scupltures numbered 20 through 29.

SeinfeldRepository.xml

1. Report the text of the synopsis for the "The phone message" episode.
2. Report the names of all season 6 episodes directed by Ackerman.
3. Report the names and the ratings of all episodes in seasons 1 and 2 that have a "historic moment".
4. Find all episodes that aired later than "The Bizzaro Jerry" and report their names and air dates.
5. Find all episodes rated A- or higher.

Submission

Put all XPath queries into a single file. Name the file `lab05-<userName>.xq`, where `<userName>` is your Cal Poly login Id. Use `"(:"` and `":)"` symbols to place comments in the file. E.g.,

```
(: Lab 5.  Alex Dekhtyar :)
(: Horses Repository      :)

(: test query            :)

document("HorsesRepository.xml")/descendant::name

(: End of File           :)
```

Submit the `.xq` file you have created using handin:

```
$ handin dekhtyar lab05 lab05-<userName>.xq
```