Lab 5: XPath Practice

**Due date:** Thursday, May 8, midnight.

## Lab Assignment

The purpose of this assignment is 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.

### 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.

### eXist Installation

**eXist** is an open-source light-weight native XML DBMS. It is distributed in the from of a Java .jar file, which is available for download from http://exist.sourceforge.net/. The links to the home page of eXist, download page and documentation are available on the course web page.

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. You may use either lab machines, or personal computers (laptops, home desktops) to complete this lab. eXist distribution comes either as a Windows .exe file (required for Vista) or as a Java .jar file (for installatio under Linux)[1].
2. Run exist, 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[2].
   - Exit program.

## 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. `<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.

[1]Please note, I have not tried installing eXist on Macs.

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

**lab02-acmsigmod.xml**

The full url for the file is:

`http://users.csc.calpoly.edu/~mlrobert/lab02/lab02-acmsigmod.xml`

This file (created by Mark Wazny of the Mean Greens group) documents the conference program of the ACM SIGMOD 2005 conference.

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

1. Find the name of the first presenter at the 2:30 session in the `Harborview I` room (place). (return as the relevant XML element node).
2. Find all XML elements containing information about SIGMOD sessions that start at 2:00pm on all days.
3. Find all starting times associated with the sessions in which `Dan Suciu` and `Nilesh Dalvi (Univ. of Washington)` are presenters. (Return the appropriate XML elements).
4. Find all sessions with at least four presenters.

**CakeryCafe.xml**

This file (created by Yuri Kapulkin) is available at the following url:

`http://wiki.csc.calpoly.edu/krgy468/browser/docs/lab2/CakeryCafe.xml`

It describes the menu of the Cakery Cafe on Foothill Boulevard in San Luis Obispo.

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

1. Find all menu items that cost between $3 and 4$.
2. Find all menu items which contain `mayo` as one of the ingredients.
3. Find all menu items that feature add-ons.
4. Find all add-on items in the `California BLT` sandwich.
5. Find any vegeterian sadwiches which contain `red onion`.

**MilestoneRCPribyshchuk.xml**

This file (created by the now defunct Blue Moon group) is available at

`http://users.csc.calpoly.edu/~mmayorga/MilestoneRCPribyshchuk.xml`

The file describes the contents of a business card.

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

1. Find the phone number that follows in the XML document the business card-holder's `Cell` phone number.
2. Retrieve every child node of the `<address>` node except for the node containing the zip code.
3. Find all phone numbers except for the `Office` number.
4. Find all phone numbers preceeding the `FAX` number.
5. For each XML element node in the tree, find its first child. Retrieve the list of first child elmenets.

### AlbumCollection.xml

This document, prepared by the members of the Lindisfarne (now, CompuGlobalMegaHyperNet), is available at:

`http://users.csc.calpoly.edu/~yong/468/AlbumCollection.xml`

The document describes a number of CDs.

The authors of this document should remove all text (comments, white space) before the `<?xml .... ?>` declaration.

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

1. Find all albums with exactly 12 tracks.
2. Find all performers (artists) whose songs are recorded on ther `Sometimes God Smiles` CD.
3. Find the last track for each album.
4. Find the name of the recording company which released the CD, which has a track named `All I ask of you`.
5. Find all tracks on which `Luciano Povarotti` has multiple cosingers.
6. Find all tracks preceding `Rudy Can't Fail` on the Clash tribute album `Burning London`.