****

**Faculty of Science  
Department of Computing**

ITEC830: XML Technologies

**Report for Assignment Two**

Semester 2, 2012

**Student Name**: Shamim Ahmed

**Student ID**: 42756561

## Introduction

The report accompanies the python program that was implemented as part of assignment two. The main goal of this assignment is to create a set of browsable HTML files from a set of XML files using DOM. The report provides instructions on how to use the python program and also offers a glimpse into the issues that were faced during the implementation.

The input XML dataset for this assignment contains medical data, which was obtained from the Journal of Family Practice for research purposes. The starting point for the data is the ‘ClinicalInquiries.xml’ file that contains a set of questions, along with several possible answers. Each answer is backed up by a set of references. There is an XML file for each reference that summarizes its contents (e.g. abstract, authors, publishers).

Our goal is to extract information from these XML files and convert them into a set of HTML files that are hyperlinked. The starting page shows a list of questions, along with a link to an answer page for each question. Clinking on the answer link displays the possible answers for a particular question, along with links to the references that support each answer. By clicking on the link for a particular reference, the user can view the information that summarizes the reference.

This task is challenging because the XML data related to the references is not entirely standardized. There is no DTD or XML schema to describe their structure, so it is necessary to explore them in order to decide how to extract the information to be displayed.

## Instructions

Before the conversion is done, the user must download the archive ‘EBMSummariserCorpus.zip’ that contains the input XML files. The archive contains ‘ClinicalInquiries.xml’, along with a folder named ‘Abstracts’ that contains all the summaries for the references.

The python program ‘clinical.py’ should be placed in the same directory as ‘ClinicalInquiries.xml’. After that, the conversion can be initiated by running the following command:

python clinical.py

The output HTML files will be available in a directory named ‘output\_42756561’. The user should start browsing through the questions and answers by opening the ‘index.html’ file in that directory.

## Description

The following is a high level overview of how the python script works:

1. The script first loads ‘ClinicalInquiries.xml’ using the DOM API.
2. It then retrieves the list of questions, along with their answers.
3. The script creates a directory named ‘output\_42756561’.
4. After this, the script creates the home page (‘index.html’) in the output folder. The home page contains the list of questions retrieved in step 2. However, the answers are not shown in the home page. Instead, a separate HTML file is created for the possible answers to each question. The home page contains links to these answer pages.
5. Each possible answer to a question is backed up by one or more references. The script needs to create a separate HTML page for each reference. So, using the reference id, it loads the XML file that contains reference data (from the ‘Abstracts’ directory). It then extracts the relevant information and creates an HTML file with that information. All such reference-specific HTML files are placed under a directory named ‘references’ (which is created inside the ‘output\_42756561’ directory).
6. The script ensures that the answer pages are linked with the reference pages so that the user can easily browse for the information he needs.

## Discussion

### Issues

1. There is no DTD or XML schema associated with the reference XML files. As a result, we had to explore them first in order to understand their structure and how we can extract the required information. The BaseX database turned out to be a valuable tool in this regard, because it allowed us to run various XQuery statements against the input XML files and determine if certain assumptions about the structure of these files are indeed correct.
2. Another issue was related to the creation of the HTML files corresponding to the references. Our script loads the XML file corresponding to a particular reference only when it is mentioned in an answer. It then creates an HTML page summarizing reference data. However, the same reference may be cited in answers to different questions. Care should be taken not to regenerate the HTML file for such a reference. Our script ensures that the HTML file for each reference is created only once and linked to all the answer pages that cite that reference.
3. It would be more convenient if we could use a third-party python library like ‘lxml’ for XML processing. In that case, we could use XPath to extract the required set of nodes easily. When we are using DOM, the only option is to access the nodes level by level and perform all the checks manually, which can sometimes be tedious.

### Limitations

1. More attention has been spent on the content of the HTML pages, rather than the presentation.
2. Some CSS is embedded in the python code.
3. Since there is no XML or DTD associated with the reference XML files, it cannot be guaranteed that this script will work when new references are added to the data set, because the new reference may violate the assumptions that have been made as part of the implementation.