**XPath Processing**

**Purpose:**

C++ standard library does not provide any facilities for managing XML. In CSE687 we’ve used some code I developed, XmlTran and XmlDocument. XmlDocument is a fairly complete XML DOM implementation I developed as an example solution for the first project in CSE687 in 2010. Some of you have used it for Project #4 this year as well. The primary issue with XmlDocument is that it is fairly complex.

The XmlTran package is much simpler, but very limited. It does not do recursive descent parsing so it will get confused if there are elements with the same tag name at different levels in the XML document.

The purpose of the XmlReaderWriter project is to fix that with a simple recursive descent parser for reading and processing an XML file or string. The basic idea is to develop parseElement(iterator iter) that, in the course of parsing an element may encounter a child element at which time it will recursively call itself on the child.

This project adds XPath navigation facilities to the XmlReaderWriter and/or to XmlDocument.

**Requirements:**

For the XPath project you will:

1. Select a subset of the XPath language to implement
2. Develop processing to navigate through an Xml string returning a std::vector of all elements that match an XPath query.
3. Implement a set of test cases to show how your processing functions.

**Here are some references:**

<http://www.ecs.syr.edu/faculty/fawcett/handouts/SummerProjects/>

<http://www.w3schools.com/xpath/>

<http://oreilly.com/perl/excerpts/system-admin-with-perl/ten-minute-xpath-utorial.html>

<http://www.ecs.syr.edu/faculty/fawcett/handouts/CSE687/code/XMLTran-TextRW/>

<http://www.ecs.syr.edu/faculty/fawcett/handouts/CSE687/Presentations/xml.ppt>

<http://www.w3schools.com/xml/default.asp>