XQ: An XML Query Language

Introduction and Overview

Kin Ng

kn2006@columbia.edu

Background

The Extensible Markup Language (XML) is a markup language for documents containing structured information dervied from SGML (ISO 8879). The goal of XML is to provide many of SGML's benefits not available in Hyper Text Markup Language (HTML) and to provide them in a language that is easier to learn and use than complete SGML. In fact, XML is really a meta-language to provide a facility to define tags and the strutural relationship between them. Unlike HTML, XML has no pre-defined tag set and any preconceived sematics. All of the semantics of an XML document will either be defined will either be defined by the applications that process them or by stylesheets.

Over the past few years, XML has rapidly gained popularity as a formatting language for information. Key application areas for XML include information management, database management, e-commerce. In fact, Simple Object Access Protocol (SOAP), one of the fastest growing protocols adopted by many application servers, provides an open, extensible way for applications to communicate using XML-based messages over Hyper Text Transfer Protocol (HTTP). The simplicity and extensibility of XML allows SOAP to bridge heterogeneous components over the Internet and provides an open mechanism for frameworks that provide Web services.

As more businesses choose to provide access to their databases and to exchange data with related businesses and organizations. We take a database view, as opposed to document view, of XML: we consider an XML document to be a database and a DTD to be a database schema. Howerver, XML standard does not address what techniques should be used for data extracting. XML query languages are needed to address this issue.

In this project, we are present a programming language, XQ, as a small trail in design and implementing of programming language for XML queries.

Related Works

Over the past few years, a lot of effort was dedicated to XML query languages research. A few languages were proposed. The commonly known ones are: XQL[1], XML-QL[2] and QUILT[3]. Their functionalities and syntax are quite similar. Currently, the W3C picked XQuery[4] (dervied from QUILT) as the query language of XML. The language is still a work in progress under the auspices of the W3C's XML Query working group. The syntax of XQuery is similar to other query language such as SQL[5].

Goals

The main goal of the project is to sucessfully design and implement a XML query language. This language will be clean semantic, easy to setup, efficient and expandable.

Clean Semantic: The syntax of this language will able to express simple queries simply. This language will be easily pickup by an experience database programming as well as a noivce application prgrammer.

Easy to setup: This lauguage will limit its external dependence. Only minimum software and hardware will be required to installation to the language

Efficient: This language will allow programmers to create indices on the document. Programmers then can provide an index as "hint" to the search engine to speed up the retrieval process.

Expandable: The implementation of the language will use a component-based design methodology. It will simplify the process of adding future enhancements.

Main Features

Besides the design goals we mentioned above, there are a few main features we will include in this language.

Data types: This laugauge will support data type such as *integer, float, char, cursor, document* and *index*. The *cursor* data type is used to hold the result set from the *select* operation. The *document* data type is used as a document handler for XML document that is being extracted. The *index* data type is used to hold the indexing information of the document.

Control flow: The flows of control of this language will consist of selection statements such as *if-else, else-if* , iteration statements such *while* and *for* ,and jump statements such as *break* and *continue*.

Ability to return an XML: This language will able to return the result in XML format.

Ability to query attributes: This language will allow programmers to query the attributes as well as the payload.

Preserve order: This language will allow programmers to use the *order by* clause to sort the result set in a particular order.

Projection and selection queries: This lauguage will support projection and selection queries. Insert and update qeuries are not included this language.

Functions: This language will provide a number of internal functions such as *print*. It also will support user define functions.

Comment: C style comment[6] will be used for this language.

Implementation Plan

In this project, we are planning to use ANTLR[7] to perform language recognition. ANTLR will construct JAVA code for the backend operations. Xerces-J[8], a JAVA based Document Object Model (DOM)[9] enabled XML parser, will be used to parse the XML document. We will use JUnit[10] for unit testing and CVS[11] for revision control. The coding style will be base Sun JAVA coding conventions[12].

Sample Code

In this section, we are going to provide a brief sample code to demonstrate the usage of this language. Since the language reference manual is still in the development phase, the final syntax for this language might be different.

```
/* start of the sample code */
document doc;
index ind1;
cursor cur;
char name[10];
integer ssn;

/* create document handler using data and DTD files */
OPEN doc WITH ("/home/johnd/students.txt", "/home/johnd/students.dtd");

/* create index on the XML docment on base on the name tag ssn. The order of the index
is DESC */
CREATE DESC INDEX ind1 ON doc ("ssn");

/* instead of using the declare keyword to create cursor, we will use CREATE in our
language */
CREATE CURSOR cur ON SELECT s.ssn, s.name FROM student s WHERE s.ssn >
123456789 USE INDEX ind1;

for (; ;) {
 (name, ssn) = FETCH cur INTO;  /* fetch result set from cursor */
  if (SQLCODE != 0) {
          break;
  }
  print(ssn, name);
}

CLOSE doc;  /* close document handler */

/* end of the sample code */
```

Bibliography

[1] XQL, J. Robie, J. Lapp, D. Schach. *XML Query Language (XQL).* See http://www.w3.org/TandS/QL/QL98/pp/xql.html.

[2] XML-QL, Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu. *A Query Language for XML.* See http://www.research.att.com/~mff/files/final.html

[3] QUILT, Don Chamberlin, Jonathan Robie, and Daniela Florescu. *Quilt: an XML Query Language for Heterogeneous Data Sources.* In *Lecture Notes in Computer Science*, Springer-Verlag,

Dec. 2000. Also available at
http://www.almaden.ibm.com/cs/people/chamberlin/quilt.html.

[4] XQuery, World Wide Web Consortium. *XQueryX, Version 1.0*. W3C Working Draft, 7 June 2001. See http://www.w3.org/TR/xqueryx

[5] SQL, International Organization for Standardization (ISO). *Information Technology-Database Language SQL*. Standard No. ISO/IEC 9075:1999

[6] C Style Comment, Brian W. Kernighan, and Dennis M. Ritchie. *The C Programming Language, Second Edition.*

[7] ANTLR, http://www.antlr.org

[8] Xerces-J, http://xml.apache.org/xerces-j

[9] Document Object Model, http://www.w3.org/DOM

[10] Junit, http://www.junit.org

[11] CVS, tttp://www.cvshome.org

[12] Sun JAVA Coding Conventions, http://java.sun.com/docs/codeconv