

An approach to portable implementation of the XQuery language*

Paraschenko Oleg

21st August 2004

Protva XQuery project

Saint Petersburg, Russia

Saint Petersburg State University

Abstract

XQuery, a query language for the XML, can be considered as a generalized interface for querying hierarchical data. The paper provokes an idea of creation of a generalized XQuery system that can process any tree-like structures and proposes realization using a virtual machine.

1 Why usual applications need XQuery

One of the approaches to XML[2] processing is a representation of XML as a tree. Navigation over an XML tree is performed using XPath[3]. XPath is a basis for XML transformation languages XSLT[4] and XQuery[5]. XSLT is a template language, and XQuery can be considered as a highly extended XPath.

XML can be considered as an external representation of in-memory tree-like structures, and XML-related standards — as methods of processing such data. Some types of applications can benefit from these approach. For examples, such applications are

- text processors and
- compilers and interpreters.

Complex hierarchical queries are very important in such applications. Examples of possible queries are:

- find all headings that are not marked as deleted, and create a table of contents;
- find all assignments used as a truth value in conditional parts of commands if, for, while and similar.

*A thesis for the First Conference of Free Software Developers on Protva River, Obninsk, Russia, 2004. The original Russian version is available at <http://xmlhack.ru/protva/xquery-thesis.pdf>.

Straightforward realization of such applications is often a hard work because a programmer concentrates on low-level details instead of concentrating on a real task. Instead of low-level coding we suggest to use an embedded XQuery. We suppose that this approach simplifies application development.

2 An approach to implementation

The main problem in implementation of an XQuery system is a big amount of work. It is not reasonable to write XQuery each time from scratch. Instead, it is reasonable to create an portable implementation and port this implementation with small enough efforts.

Our approach is to develop a virtual machine for tree processing and an XQuery system on top of this virtual machine. In case some application need embedded XQuery support, it is enough to realize only the virtual machine.

In order to simplify development, we write and debug code using the Common Lisp language¹. The development is based on the CL-XML[6] system. James Anderson, the author of the CL-XML, already have done a lot, so CL-XML can process examples from the XQuery specification.

CL-XML translates XQuery into an internal representation in a form of an abstract syntax tree (AST). AST can be used for high-level optimization of queries.

Most of the language constructions of Common Lisp are realized using macros. If man takes CL-XML and expands all macros, then man gets a program which consists of primitives only. A set of the produced primitives is a draft version of a virtual machine.

Realization of a dialect of a minimal Lisp (or Scheme) is a simple enough task because this task is well known and investigated[7].

A special interest is a translation of our virtual machine into the LLVM[8] — an intermediate representation and a set of tools for program optimization. Due to LLVM and high-level optimization, an automatically generated code for XQuery evaluation may be more efficient than a hand crafted code.

3 Status

The project status is an idea. Right now only a plan of work, first experiments and selection of tools are done. Intermediate results will be uploaded on the home page of the Protva XQuery[1] project. Results will be distributed under the terms of the LGPL license.

The conference talk will demonstrate a draft of a virtual machine and the first results.

A first planned milestone² of the project is addition of an XQuery support into the XSLT-processor xsltproc[9].

¹15 August 2004: Now we work with Scheme and SXPath/SXQuery instead of Common Lisp and CL-XML. Anyway, all the following ideas are still the same.

²15 August 2004. Now it is a second milestone. The first one is an XQuery support for the Python AST.

References

- [1] Project Protva XQuery <http://xmlhack.ru/protva/>
- [2] Extensible Markup Language (XML) <http://www.w3.org/XML/>
- [3] XML Path Language (XPath) <http://www.w3.org/TR/xpath>
- [4] The Extensible Stylesheet Language Family (XSL) <http://www.w3.org/Style/XSL/>
- [5] XML Query (XQuery) <http://www.w3.org/XML/Query>
- [6] CL-XML: Common Lisp support for XML <http://cl-xml.org/>
- [7] Scheme Compiler Technology/Implementation Techniques and Optimization <http://library.readscheme.org/page8.html>
- [8] The LLVM Compiler Infrastructure Project. <http://llvm.cs.uiuc.edu/>
- [9] The XSLT C library for Gnome <http://xmlsoft.org/XSLT/>