

Project Proposal and Literature Review

Eliza Brock

April 8, 2008

Contents

1	XPath Evaluator Project Proposal	2
1.1	Objectives:	2
1.2	Options for Further Work:	3
2	Team & Organizational Plan	4
2.1	Group Organization	4
2.2	Major Tasks	4
2.3	Proposed Timeline	5
3	Initial Reference List	6

1 XPath Evaluator Project Proposal

1.1 Objectives:

The primary objective of the XPath Evaluator is to serve as a pedagogical tool to illustrate the basic principles of querying XML using XPath.

The XPath Evaluator will be used to make visual connections between sections of the XPath query and XML elements that are selected by the queries. It will generate a dynamic visual representation of XML data and XPath expressions in order to represent the effect of each individual XPath subquery on the transaction as a whole. The idea is to allow students to understand the results of each portion of their XPath query as it is evaluated, thus preventing some of the confusion surrounding the basic question of “How did these results happen?”

The project will be accomplished using a Haskell based server process, and an ajax/javascript driven browser interface. See Table 1 for details on the proposed client-browser interaction. Using a web browser as the main interface will make the project more accessible to learners and usable from anywhere.

Client On the client side, the user will be able to select an XML document, and input an XPath expression in order to see the results of the query evaluation. The goal is for these interactions to be dynamic and fluid, such that the interactions between user, browser, and server are near real-time. With near real-time interactions, the browser and server will be able to collaborate to show the most current results of query evaluation as each sub-expression is completed.

Portions of the HTML-based UI will be sourced from open source projects, in order to minimize time spent on basic client-side functionality.

Server On the server side, the XML will be parsed into a form suitable for use by the server process and will then be divided via the various XPath sub-expressions. At the request of the client, the server will send information

about the entire XML document (for display by the client) and the results of various XPath sub-expressions.

Planned Implementation of the XPath Standard: The portions of the XPath standard that I intend to implement are:

- Location Steps, including axes and node tests
- Predicates
- Standard operators¹, excluding “|”
- Portions of the core library, as time permits. The implementation of the core library will focus on the following common functions:
 - position
 - count
 - last
 - first
 - contains
 - name
 - local-name

1.2 Options for Further Work:

Additional work on this project, apart from the basic features outlined above could include any of the following:

- Calculating the XPath expression whose result would be a given set of nodes.²
- Implementing the full XPath standard
- Determining whether a query that returns no results for a given XML document could be satisfiable given the document’s DTD.

¹As shown at: http://www.w3schools.com/xpath/xpath_operators.asp

²This could be a corollary to the functionality provided for regular expressions at: <http://www.txt2re.com/?s=29:Mar:2008%20%22This%20is%20an%20Example!%22&-19>

Table 1: Proposed Workflow

Client	Server
Requests page	
	Server sends page
User selects XML file to use	
	Graphical representation of the XML is generated
User inputs XPath, which is sent to the server on a per-token basis	
	Identifiers of elements selected by individual sub-expression are returned to the client
The browser highlights the elements and corresponding sub-expressions	

2 Team & Organizational Plan

2.1 Group Organization

Name	Task/Role
Eliza Brock	Jack-of-all-Trades

2.2 Major Tasks

1. XML Parsing
 - (a) Parse XML
 - (b) Transform XML into convenient data structure
2. XPath Parsing
 - (a) Allow users to specify query
 - (b) Search XML based on XPath query
 - (c) (if time allows) An item selected from Section 1.2
3. Browser Interaction

- (a) Specify XML to use
- (b) Display XML in visual form
- (c) Connections between XPath and XML elements

4. Paperwork

- (a) Project Proposal (Wk. 5)
- (b) Milestone Report (Wk. 6)
- (c) Status Report (Wk. 8)
- (d) Final Report (Wk. 10)
- (e) Research Paper (Wk. 10)
- (f) Final Code (Wk.10)

2.3 Proposed Timeline

Week	Goals [Tasks]	Deadlines [Tasks]
4.5th	Templates of all reports [4*] Finding suitable XML with DTDs [1*]	
5th	Parsing XML into datastructure [1a, 1b] Parsing XPath by token [2a]	Project Proposal [4a]
6th	Visual display of XML elements [3a, 3b]	Literature/Design [4b]
7th	Connect XPath & XML [2b, 3c]	
8th	<i>Margin of Error</i>	Status Report [4c] Project Demonstration
9th	Final features[2c]	
10th	<i>Margin of Error</i>	Project Delivery[4d-f] Project Demonstration

3 Initial Reference List

- [1] BARTON, C., CHARLES, P., GOYAL, D., RAGHAVACHARI, M., FONTOURA, M., AND JOSIFOVSKI, V. Streaming XPath processing with forward and backward axes. *Data Engineering, 2003. Proceedings. 19th International Conference on* (5-8 March 2003), 455–466.
- [2] BENEDIKT, M., FAN, W., AND GEERTS, F. XPath satisfiability in the presence of DTDs. In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2005), ACM, pp. 25–36.
- [3] GENEVÈS, P., AND LAYAÏDA, N. A system for the static analysis of XPath. *ACM Trans. Inf. Syst.* 24, 4 (2006), 475–502.
- [4] GOTTLÖB, G., KOCH, C., AND PICHLER, R. The complexity of XPath query evaluation. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2003), ACM, pp. 179–190.
- [5] GOTTLÖB, G., KOCH, C., AND PICHLER, R. Efficient algorithms for processing XPath queries. *ACM Trans. Database Syst.* 30, 2 (2005), 444–491.
- [6] GROPE, J., AND GROPE, S. Filtering unsatisfiable XPath queries. *Data I& Knowledge Engineering* 64, 1 (January 2008), 134–169.
- [7] HUANG, C.-H., CHUANG, T.-R., AND LEE, H.-M. Prefiltering techniques for efficient XML document processing. In *DocEng '05: Proceedings of the 2005 ACM symposium on Document engineering* (New York, NY, USA, 2005), ACM, pp. 149–158.
- [8] KOCH, C. Processing queries on tree-structured data efficiently. In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2006), ACM, pp. 213–224.
- [9] MARX, M. Conditional XPath, the first order complete XPath dialect. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-*

- SIGART symposium on Principles of database systems* (New York, NY, USA, 2004), ACM, pp. 13–22.
- [10] MARX, M. Conditional XPath. *ACM Trans. Database Syst.* 30, 4 (2005), 929–959.
 - [11] MARX, M., AND DE RIJKE, M. Semantic characterizations of navigational XPath. *SIGMOD Rec.* 34, 2 (2005), 41–46.
 - [12] MICHAEL BENEDIKT, W. F., AND KUPER, G. Structural properties of XPath fragments. *Theoretical Computer Science* 336, 1 (May 2005), 3–31.
 - [13] QUN CHEN, ANDREW LIM, K. O., AND TANG, J. Indexing XML documents for XPath query processing in external memory. *Data & Knowledge Engineering* 59, 3 (Decemeber 2006), 681–699.
 - [14] SANJAY MADRIA, YAN CHEN, K. P., AND BHOWMICK, S. Efficient processing of XPath queries using indexes. *Information Systems* 32, 1 (March 2007), 134–169.
 - [15] SCARDINA, M., AND WANG, J. Building an XPath-powered framework for XML data processing. In *XML 2004 Proceedings* (2004).
 - [16] TAJIMA, K., AND FUKUI, Y. Answering XPath queries over networks by sending minimal views. In *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases* (2004), VLDB Endowment, pp. 48–59.
 - [17] TEN CATE, B., AND MARX, M. Navigational XPath: calculus and algebra. *SIGMOD Rec.* 36, 2 (2007), 19–26.
 - [18] W3C. *XML Path Language (XPath)*, 1.0 ed., November 1999.
 - [19] W3C. *Extensible Markup Language (XML) 1.0*, 4.0 ed., September 2006.