

Dissertation Title: Enabling Search for Mathematical Expressions

Abstract by: Moody Ebrahim Al-Tamimi
Email: moody@gwu.edu
Doctoral Candidate at the Department of Computer Science
The George Washington University
Washington, DC 20052

Research Advisor: Dr. Abdou Youssef
Professor at the Department of Computer Science
The George Washington University
Washington, DC 20052
Tel: (202) 994-6569
Fax: (202) 994-4875
Email: ayoussef@gwu.edu
Office: Academic Center, Room 708 Phillips Hall

Abstract

Problem Description

Searching for mathematical expressions on the web is impossible at this time. Even though search technology today is mature, it only facilitates text search only and doesn't handle non-alphabetic characters. A user cannot search for a simple equation like $y = x + 3$ in a search engine like Google for example. HTML, the most popular language used to describe text on the web, is not rich enough to describe the highly structured nature of mathematical expressions. The nature of text is different from that of mathematical expressions; text is a linear organization of letters while math expressions are a 2 dimensional arrangement of symbols. The only math content search available is to search for keywords (text) describing the mathematical content on a web document. That is not enough to fully utilize the many mathematical resources out there today. Until recently, a standard language to properly describe mathematical content didn't exist. Most data formats were propriety or focus only on encoding the presentational aspect of expressions like $L^A_T E_X$ for example.

Math search is a new area of research with many challenges. Some of the challenges are technological; standard encoding that enables processing of mathematics is recent due to the nature of the mathematical writing system. A mathematical expression is used to communicate ideas using an arrangement of symbols. Different ideas can be expressed using the same notation, and different notation can be used to describe the same idea. As a result, to describe mathematical content properly, a description of how the symbols are arranged needs to be encoded in addition to a separate encoding of the underlying organization of ideas. The conceptual structure eliminates the inconsistency and ambiguity problems of presentational structures. This is necessary when designing applications that process mathematical content. A search tool cannot simply "guess" what the user really intended when searching for $g(x+y)$, is the search for an arithmetic operation (multiplying a variable g by the sum of variables x and y) or is it for an algebraic function (function g with the sum of variables x and y as an argument). The recent development of MathML, an XML based language, for encoding mathematics was a needed technology for any attempt to solve the math search problem. Another needed technology is a standard implementation of an XML Query engine that enables extracting information from XML based documents.

The focus of this research is to develop novel techniques that allow the user to describe a mathematical formula using a simple math query language that will then be executed against a data repository of files containing mathematical formulas. There are many issues that are addressed in the dissertation, to name a few: A user should be able to enter a math sentence that can have different representations in different files. In additions, math laws need to be taken into account, for example: priority, commutative, and associative laws.

Approach

- 1- A comparison was made between the existing two standards for encoding the conceptual level of mathematical expressions, Content MathML and OpenMath. Content MathML was chosen because it is the W3C standard and has more vendor support.
- 2- A decision was made to integrate an existing implementation of the latest version of the XML Query specification. An evaluation of existing engines was made focusing on the following criteria: Open source, java based, implemented most of the latest XML Query specification, and finally enables the execution of XML queries against XML files and not just proprietary XML databases. Qizx/open was selected and was successfully integrated in the search tool.
- 3- A Math query language has been developed that allows the user to describe mathematical expressions. The language is brief, easy to use, and flexible. Priority and association laws are recognized by the parser without requiring the user to enter additional symbols. For example: $x + y * z$ is interpreted accordingly and the user is not required to write the expression as $x *(y +z)$. In the case of operators and special characters that don't exist on the keyboard, the user is not required to use additional software. For example, in the expression $a \equiv b$, the \equiv symbol can be represented as $= = =$. In addition, the language allows the user to enter partial expressions. For example: search for $/b$ is permitted.
- 4- A translator that will parse the user query into an XPath expression has been developed. The objective is to write an XPath expression that will find the MathML tree that corresponds to the math sentence in a document and also to point to the main element that encompasses the entire tree in the document. Pointing to the main element is important so that we can later on apply some highlighting technique to highlight the main element and its children that correspond to the math sentence and nothing else.
- 5- The capability to allow searches using Boolean operators is to be added. The user should be allowed to enter complex expressions that include Boolean operators. For example: $a (+ \text{ or } -) b$. In this case, "not" "or" "and" will be used in order to differentiate the Boolean operators from the corresponding operators used in the query language "~" "||" "&&"
- 6- A more flexible user interface that allows the user to express his needs is under development. Tools are being developed to help the user in writing a query. Such tools must allow the user to be precise and also given flexibility. For example, when the user enters the letter e, it has different interpretations: e-notation, exponential E, or 2.71828. The user must have the option of choosing what he/she intended.
- 7- A thesaurus for reserved symbols and their various interpretations is under development. This will then later be used in relaxing the query based on a math sentence. For example, Z can have a strict interpretation of an identifier called z or can be synonymous to the set of integers.
- 8- A content transformation module is under development. This module will take user authored MathML documents and transform them into a "normalized" canonical form according to well defined specifications. The user query is already normalized during the parsing phase. The canonical form deals with issues like distributive and communicative laws to name a few.
- 9- Outcome metrics to measure recall, precision, and expressiveness of (XPath vs. XQuery) are under development. In addition, process metrics like speed are also needed.

Contribution

- The syntax and grammar of simple math query language
- Parser + translator from "simple math query" to "XQuery/XPath expression"
- Thesaurus for reserved symbols and their various interpretations
- Algorithms to convert user query to a form that conforms to the normalized set of files. In addition to algorithms that will convert user authored documents into the normalized form: i.e. "query \rightarrow normalized query" against "normalized set of MathML docs \leftarrow user authored MathML docs"
- Identify where the line ends for XPath and starts for XQuery, what mathematical expressions can be expressed fully in XPath and what expressions cannot. This is important since XPath is a subset of XQuery.

Keywords

Document search, XQuery, Mathematical queries, Mathematical expressions

Resume

MOODY EBRAHEM AL-TAMIMI

*Doctoral Candidate
Department of Computer Science
The George Washington University
Washington, D.C.
me_altamimi@yahoo.com
(703) 629 8904*

SUMMARY

Ms. Al-Tamimi is currently working on finishing her doctoral dissertation. She is well underway in research in the area of search and retrieval of mathematical content. In particular, she is focusing on providing a math appropriate query language and user interface that enables users to express their information needs, which often involve math symbols and structure. Search of mathematical content is a new area and presents new research challenges to the search community. This is because conventional search systems handle text and are not designed to handle content that involves non-alphabetical symbols.

Ms. Al-Tamimi has also extensive experience in Object Oriented methodologies. She has a Masters of Science degree from The George Washington University in Software Engineering, and her skills in custom software development range from languages like Java, C, and C++, to PERL and Eiffel. Her expertise in commerce-enabled Web technologies is evident in several commercial Internet storefronts where she has augmented the shopping experience with automated purchasing processes. In addition, she has considerable skill in software testing and character/pattern recognition algorithms and systems.

TECHNICAL SKILLS

Languages: Java, XML, XPath, XQuery, PERL, JavaScript, VBScript, C/C++, Eiffel, XHTML, MathML, OpenMath

Software: Sun Java JDK, IBM Aglets SDK, Visual C++, Borland C++, Active Server Pages, Photoshop, Image Composer, dBase IV, Object Domain, Word Perfect, Microsoft Office, Tcov, ADL, BusinessObjects, MS Site Server 3.0, Kawa

Databases: Oracle 8, SQL Server 6.5, and Access 97

Search Engines: XQuery search engines: Quip, Qizxopen, and Kawa

Operating Systems: UNIX (System V/ Solaris), Windows XP, Windows NT Server and Workstation 3.51/4.0, Windows 95, Windows 3.11, Novell Netware 4.1, MS DOS

Hardware PC and Sun

CERTIFICATIONS

Java certified programmer (March 1999)

PROFESSIONAL EXPERIENCE

Network Associate Labs

Apr - Oct 2001

Researcher

Glenwood, Maryland

Self-Protecting Mobile Agents (SPMA) Project

- Participated in the development of effective techniques that allow mobile agents to protect themselves from malicious host computers.

- Conducted intensive debugging of the “Distributed Agent State” technique developed for SPMA.
- Researched possible techniques to enhance fault-tolerance in SPMA.

Noblestar Systems Corporation

1997 – May 2000

Consultant

Reston, Virginia

Financial Passport

- Improved and enhanced/ augmented system documentation for new team members.
- Conducted intensive debugging of one of the system’s components.
- Re-designed and redeveloped an existing module to allow the object oriented framework.

Net2000 Communications

- Lead the requirements analysis effort and interviewed potential user groups to understand their business requirements, which resulted in a detailed requirement document.
- Designed and built Business Objects V4.0 and V5.0 repository/ universe to give user groups the ability to report directly off of Saville and the ability to report off of Net2000 Billing Data-mart.
- Set up users groups and access right to the universes, and created Business Objects reports required by each user group.
- Created test plan, user test sheets for the universes, and user guides; conducted training for the different user groups.

Riverbed Technologies

- Built a frame based order web site, which consists of an Order Center, an Order Fulfillment Center, and an Admin Center.
- Authored Active Server Pages (ASP) code for dynamic HTML generation using JavaScript V2.0 and VBScript V5.0 and to perform client side validation of data and business rules enforcement.
- Connected the Web site to an Access V97 database using ODBC.
- Installed, configured, and integrated MS IIS V4.0 Secure Server.
- Installed MCK V3.2 (Merchant Connection Kit) from CyberCash and integrated it as part of the credit card application for credit validation by modifying MCK ASP scripts.

Freddie Mac

- Advised Freddie Mac on future directions and technologies for their corporate Intranet.
- Developed prototypes and recommended technology suites (Java servlets and front-end applications) for the revamp and deployment of new applications.
- Designed access limits, dynamic contact, on-line process and multimedia features within each of these groups.

Multifamily Housing Institute

- Built a credit card application for Multifamily Housing Institute as part of their AptData application for user credit card entry on-line.
- Authored LiveWire code for dynamic HTML generation using JavaScript V1.2.
- Connected the Web site to an Oracle 8 database using ODBC.
- Devised JavaScript code to perform client side validation of data and business rules enforcement.
- Installed, configured, and integrated MCK V3.2 (Merchant Connection Kit) from CyberCash as part of the credit card application for credit validation by modifying MCK C scripts.
- Wrote procedures to dynamically read the Oracle system catalog and create HTML input and edit forms for any user table in the database.
- Installed and configured ArcView (ESRI mapping server V3.0).

Nationwide Insurance

- Designed and built a framed-based Online Purchasing Handbooks site for Nationwide Insurance as part of their shopping cart application using HTML V3.2.
- Designed all graphics for the site using Adobe Photoshop.
- Used Active Server Pages (ASP) to generate all HTML dynamically using VBScript and Jscript, and used ADO to connect to a SQL Server database, retrieving catalog information and posting purchasing requests.
- Built an Intranet shopping cart application for Nationwide Insurance that enables purchasers for the organization to buy hardware, software, peripherals, office furniture, and other office supplies on-line.
- Built application interfaces with an EDI system that propagates all purchasing data directly to suppliers of these goods and services.
- Authored Active Server Pages code for dynamic HTML generation using a combination of VBScript, JScript, and Perl.
- Connected the web site to SQL Server V6.5 using ADODB components of ASP.
- Devised JavaScript V1.1 code to perform client side validation of data and business rules enforcement.
- Installed and configured Windows NT Server V4.0 and Internet Information Server V3.0.
- Designed all graphics for the site using Microsoft Image Composer and Adobe Photoshop.

Gatekeepers Internet Marketing

1996 - 1997

Web site Developer

Washington, DC

- Designed and implemented “Calculate and Display Customer Request” scripts for an online restaurant using C to track and retrieve user navigational data.
- Performed graphic design, page layout, page coding, script coding, and server configuration and deployment.
- Developed an online shopping-basket-style shoe store on the Web that allowed users to browse shoe types, sizes, and models and order online using PERL.

KAAU

1990 - 1994

Undergraduate Student

Computer Science Department
Saudi Arabia

- Lectured to the Computer Science Department staff on Software Quality Assurance.
- Designed and developed a system for Arabic character recognition using Visual C++ (senior project), where an original branching algorithm for pattern identification was implemented.
- Specialized in Object Oriented design, implementation of applications using C++.
- Led application development review and testing of the “School Registration System”. Load, stress, penetration, and logical testing were conducted using TCOV tools in concert with manual methods.
- Served as a computer science tutor in three University departments – Math, Statistics, and Computer Science.
- Utilized computer software knowledge to create smoother documentation processes.

EDUCATION

George Washington University

Washington DC

D.Sc. Computer Science with focus on the area of Information Retrieval (expected May 2005)

George Washington University

Washington DC

MS Software Engineering (1997)

King Abdul Aziz University

Saudi Arabia

BS Computer Science (1994)

Awards

Recipient of Educational and Research Funding Organization, Inc. Scholarship (1995)