MATHML IN PRACTICE: ISSUES AND PROMISE

T. W. Cole*[1]

*[1] *Mathematics Librarian, University of Illinois at Urbana-Champaign, 1409 W. Green St., Urbana, IL 61801*
Email: t-cole3@uiuc.edu

ABSTRACT

This paper discusses MathML, an XML-based standard for expressing mathematics - everything from elementary mathematics to undergraduate college-level mathematics. Limitations of pre-existing options led to the creation of MathML. MathML is designed to be useful for authoring and publishing, for creating online interactive math resources, and as a non-proprietary approach for archiving. MathML is supported by the W3C and by multiple scholarly publishers and vendors of computer-based mathematics software. The question now is whether MathML can achieve greater acceptance among authors and better integration with standards in related domains, either through cross-walks or through direct incorporation into other domain schemas.

Keywords: MathML, XML, Digital libraries, Science and technology publishing.

1 INTRODUCTION

The history of mathematical notation is nearly as long and circuitous as the history of mathematics itself and has been well chronicled (e.g., Cajori, 1928). The relatively high-degree of international standardization in printed mathematical notation today facilitates the volume and level of scientific scholarly discourse we currently enjoy. (Though even now, mathematics notation is far from static. Each year a handful of new symbols and new notations are introduced to accommodate cutting-edge research in mathematics and in related scientific fields, such as mathematical physics.) Until recently, the primary challenge in publishing formal scholarly writings involving high-level mathematics has been largely a matter of typography, i.e., an issue of how best to represent on the printed page and for human consumption the mathematics intended by the author. Conventions developed over the centuries have led to relatively compact formats optimized for the printed page and designed to be easily read and understood by a human readership.

The advent of ubiquitous distributed computing, however, has led to additional complexities. Of first priority was the development of a means to convey unambiguously the conventions of mathematical printed page typography to computer-based publishing systems. In the late 1970s, Donald Knuth developed TEX, a computer language for mathematical typography. It has been immensely successful and remains, in one variant or another, the predominant way in which academic mathematicians format and submit research mathematics for print publication. Knuth's philosophy in developing TEX was that typography, when considered “as the servant of mathematics,” should have as a goal “to communicate mathematics effectively by making it possible to publish mathematical papers and books of high quality, without excessive cost.” (Knuth, 1979)

However, while TEX has facilitated the print publication of scholarly mathematics, it was not designed to fully leverage the power of modern computing systems in other ways. Computers are more than just traditional books on steroids. Pre-dating the Web by more than a decade, TEX is not especially well-suited for use in Web browsers, nor does it facilitate the creation and use of distributed, dynamic and interactive publication formats enabled by the ubiquitous nature of modern PCs and by the Web itself. Consider classroom or practical laboratory settings where direct interaction with published mathematics could be desirable, e.g., a researcher who wants to plug his or her data directly into a published mathematical formula and then graph the results or an engineering student who wants to evaluate over a particular range of values an integral or function found on a Web page. TEX was not designed (primarily at least) with this use in mind. TEX also has limitations for archiving literature in digital form. For extensibility, TEX allows each author to extend the language with home-grown macros. Identifying and retaining long-term the custom macros required for each published article or paper on an article-by-article basis becomes cumbersome. Ownership and versioning of such custom macros can also become an issue when archiving.

The crux of the problem is that humans and computers process information differently. While $\TeX$ is extremely good at allowing an author to specify exactly how his or her equation should look when printed on a page in a journal or book to be read by a human reader in the context of an article or chapter, it is not as effective in conveying the meaning of mathematics in a way that can reliably and unambiguously be understood by computer applications needing to evaluate, graph, or otherwise exploit the math interactively. Compactness, visual balance, and fitting correctly within the bounds of the printed page are all desirable characteristics when trying to convey the overall meaning and structure of mathematics to a human reading a print document. But trade-offs made to accomplish such objectives in print can lead to ambiguities or loss of precision in conveying more complete semantic meaning to a computer program. What is needed in the digital realm is a way to describe not only how the math should look on the screen to a human reader, but also what is meant by the mathematics in a way that can be understood unambiguously by computer applications designed both to present the mathematics to the user and to allow direct and robust user interaction with the mathematics itself.

MathML, an XML-based standard for expressing mathematics, was created in an effort to address this need. MathML focuses more on the meaning of the mathematics than any previous non-proprietary standard for expressing mathematics in digital format. The intent is to provide a sort of *lingua franca* for mathematics up through the underclass undergraduate level, as a way to facilitate interoperability among and between a wide range of scientific tools and data applications, and as a way to advance more broadly scholarly communication (and the archiving of that communication) in mathematics and in the sciences. MathML has significant potential. How well that potential will be realized remains to be seen. This paper summarizes a presentation about MathML from the *Workshop on International Scientific Data Standards and Digital Libraries* held June 10-11, 2005, in Denver, CO, in conjunction with the 5th Annual ACM/IEEE Joint Conference on Digital Libraries. The intent here is to introduce the influences that drove the early evolution of MathML, the basics of its structure, the current state-of-the-art, and the priorities and issues for MathML going forward. The focus is on practical aspects of MathML and issues relevant to the implementation of MathML in the context of digital libraries.

2 WHERE MATHML CAME FROM AND WHAT IT IS

2.1 Mathematics in SGML & HTML

Concurrent with the introduction of $\TeX$ in the domain of mathematics, independent efforts were underway to better encode the general intellectual structure and meaning of digitized content. Much of this effort centered on the development and exploitation of Standard Generalized Markup Language (SGML). As a precursor of both HTML and XML, SGML was designed to provide a way to better describe and delineate the intellectual structure of document-like digital content. A number of science and technology publishers began exploiting SGML in the 1980s, and during that decade, the American Association of Publishers (AAP) introduced a set of SGML Document Type Definitions (DTDs) defining standard ways to mark-up article and book content in SGML. They included a specific component DTD for marking up mathematical content in SGML. By 1994 an international standard DTD set (ISO 12083) for articles and books in SGML had been promulgated. ISO 12083 also included its own DTD component for mathematics markup (available at <http://www.xmlxperts.com/mathdtd.htm>). Also by the mid-1990s, SGML was being evaluated for its potential as an infrastructure component for digital library systems (Cole & Kazmer, 1995).

Experience with the standard SGML DTDs for mathematics was mixed. In spite of SGML's focus on delineating intellectual structure and semantics, the early SGML DTDs for mathematics took a very presentational approach to the markup of mathematics. Both the AAP and the ISO 12083 versions of the mathematics DTD were still mostly concerned with describing how the mathematics was meant to look on the printed page. Even in that regard, they were found insufficient for higher level mathematics of the kind often found in scholarly science and technology publishing. Most publishers found it necessary to extend the baseline AAP and ISO 12083 SGML DTDs for mathematics. They did so in idiosyncratic ways, which exacerbated difficulties in interoperability and efforts to clarify the meaning of the mathematics (Mischo & Cole, 2000). SGML also proved difficult to implement on the Web. Major Web browser vendors decided that the SGML standard was too heavy weight to implement within the browser. The most successful browser plug-in for SGML, SoftQuad's Panorama, while adequate for general text in SGML, was not adequate to render high-end mathematics found in scholarly publications.

Even as ISO 12083 was adopted, the W3C, the consortium that serves as home for most major pan-Web standards and initiatives, was considering including more advanced mathematics within HTML. Early work on HTML version 3 aimed to add a number of specialized tags intended especially for rendering mathematics. The approach taken, as would be expected given the nature of HTML, was focused on the presentation of the mathematics on the screen, not on the meaning of the mathematics. In any event, the effort to augment HTML with specialized tags for the mathematics community ran into some resistance from HTML users and Web browser vendors. One of the strengths of HTML lay in its limited element set. The scope of proposals to augment HTML with additional mathematics elements was seen as going against the grain. HTML version 3.2, as finally ratified, did not include many of the math-specific elements that had been considered, and plans for further mathematics-related additions to HTML were dropped. The math-related limitations for the Web of SGML, $\mathrm{T_EX}$, and HTML suggested that another approach was needed. (For further discussions of these limitations see Ion, 1999 and Cole, 2001.)

2.2 The W3C Mathematics Working Group

By 1997 the W3C was committed to two related initiatives - the introduction of XML as a Web-friendly alternative to SGML and the creation of a new mathematics markup language as the first major example of a community-based XML implementation. The W3C Mathematics Working Group (home page at <http://www.w3.org/Math/>) was chartered and charged with the task of creating MathML. This working group, chaired by Patrick Ion of Mathematical Reviews, brought together publishers, markup language experts, and vendors of mathematical tools and software (e.g., Wolfram Research, MapleSoft, and Design Science). By this time, mathematicians and software engineers creating mathematical tools for the academic market had made great strides in understanding ways to successfully represent mathematical notation in forms that could be understood and acted on by computers (Wolfram, 2000). As consumers and users of SGML, science and technology publishers as well had learned a great deal about the practical limitations of early attempts to define markup schemes for mathematics.

However, though the issues and problems were better understood, there remained controversy as to how the new standard should address past concerns. In particular there were questions about scope (how complete and comprehensive MathML should be), the balance between the need to unambiguously define how a mathematical expression should be rendered on the screen (and potentially in print), and the need to describe the meaning (semantics) of the mathematics in a manner that could be understood by computer applications reliably and unambiguously. In the end, compromises were struck in regard to both issues.

2.3 The Two Components of MathML

Version 1.0 of MathML was released in 1998, not long after the XML 1.0 specification itself was released. As a compromise between simplicity and comprehensiveness, the scope of MathML was defined as elementary mathematics through elementary calculus, consistent at the high end with mathematics taught through underclass undergraduate university level. Though some extension mechanisms were included, MathML does not claim to be adequate for all mathematics needed for graduate-level work and cutting-edge research in the field, trying instead to be compatible with and a good foundation for ongoing and future work by others in that direction.

MathML includes two, non-overlapping element sets, one consisting of about 30 "Presentation MathML" elements (as of MathML version 2.0) designed to express how the mathematics should be presented on the screen or page and the other consisting of about 120 "Content MathML" elements (as of MathML version 2.0) designed to express the meaning or semantics of the mathematical content. MathML allows implementers to use either set of elements singly or both tag sets in combination. Obviously using both together provides the most complete information, but at times the cost (labor) of doing so can be an issue. With legacy mathematics as well (e.g., SGML mathematics), it may not be possible to automate a reliable transformation that can create more than just Presentation MathML. It should be recognized, however, that Presentation MathML is not completely void of semantic meaning (and vice versa). Frequently Presentation MathML contains meaning adequate for most purposes. Wolfram's *Mathematica* tool, for example, imports presentation MathML and does a pretty good job, albeit not always perfect, of inferring sufficient meaning to exploit the transformed Presentation MathML within the *Mathematica* environment. Figures 1 and 2 provide simple illustrations of Presentation and Content MathML. The current MathML specification (version 2, second edition) is available at <http://www.w3.org/TR/MathML2/>.

$$\sqrt{z} = z^{\frac{1}{2}}$$

Presentation MathML:

```
<math xmlns='http://www.w3.org/1998/Math/MathML‘>
   <mrow>
    <msqrt> <mi>z</mi> </msqrt>
    <mo>=</mo>
    <msup> <mi>z</mi>
      <mfrac><mn>1</mn><mn>2</mn></mfrac>
    </msup>
  </mrow>
</math>
```

Figure 1. An example of Presentation MathML encoding of a simple mathematical equation.

.

$$\sqrt[2]{z} = z^{\frac{1}{2}}$$

Content MathML:

```
<math xmlns='http://www.w3.org/1998/Math/MathML‘>
<apply><eq/>
  <apply><root/>
             <degree><cn type='integer'>2</cn></degree><ci>z</ci></apply>
  <apply><power/>
             <ci>z</ci><cn type='rational'>1<sep/>2</cn></apply>
</apply>
</math>
```

Figure 2. An example of Content MathML encoding of a simple mathematical equation.

Note that both kinds of MathML are verbose relative to the amount of content they describe. Also order can be significant in MathML, unlike in some XML implementations. Presentation MathML is used to describe the layout structure of mathematical notation and so focuses on visual constructs. It includes both Token elements (i.e., elements that rely primarily on simple Parsed Character Data content models) and Layout Schemata (for building expressions using element-based content models). To provide some semantic richness even in Presentation MathML, semantic hints via invisible characters (e.g., implicit multiplication) are encouraged. Content MathML is intended to support encoding of the underlying mathematical structure of an expression, rather than any particular rendering for the expression. Mathematical semantics and grammar is as complicated as any language. (Increasingly it is being studied using classic linguistic tools and analytical methods). To keep scope and complexity manageable, Content MathML is limited in scope, designed to express “commonplace mathematical constructs,” i.e., through about the first 2 years of college. Content MathML was built with mechanisms for extensibility. For instance, MathML can be extended through the use of OpenMath <http://www.openmath.org/>, which provides additional semantics for describing more complex mathematical expressions.

3 MATHML -- THE STATE OF THE ART TODAY

3.1 Using MathML on the Web

MathML is neither as complete nor as powerful as the internal mathematics notational schemes used by sophisticated computer algebra systems like *Mathematica* or *Maple*. On the other hand, MathML is not a proprietary format tied to a particular rendering engine. This makes it valuable as a means to share mathematics between such tools and for direct use on the Web. The current Mozilla family of Web browsers, including Firefox, provides native support for rendering MathML. Microsoft Internet Explorer does not provide native support for MathML, however, specialized plug-ins are now available from Integre and Design Science that exploit advanced features of Internet Explorer to render MathML seamlessly within the browser window. More importantly, tools now exist for Web browsers which can exploit the semantic richness of MathML to allow the user to interact with the mathematics within the confines of the Web browser client, e.g., to evaluate or graphically display mathematical expressions over an arbitrary range of variable values in the browser window. Robust MathML-aware Web services are also available (e.g., Wolfram's MathML Central service, mentioned below).

Quality and completeness of MathML rendering in both Firefox and Internet Explorer is dependent upon the availability of essential fonts and glyphs on the client computer. While math-specific font sets have long been commercially available, there has been a lack of standardization in how to reference individual characters needed for advanced mathematics and limited availability in the public domain of more advanced glyphs. Over the last several years, the STIX Fonts Project <http://www.stixfonts.org/>, a collaboration of several leading science and technology publishers working in concert with the W3C Mathematics Working Group, has led the effort to alleviate these problems. Lobbying of the Unicode Technical Committee by the STIX Project led to the inclusion between Unicode release 3.0 and release 3.2 of more than 1,600 additional code points covering advanced mathematics (Unicode Consortium, 2002). In addition, the STIX Project has undertaken to create more than 5,000 new glyphs of mathematics symbols which will be available to the general public without cost. (More than 4,800 glyphs have been created so far with release of the full set expected later in 2006.) With the combination of the several hundred math code points already in Unicode as of release 3.0 and the more than 2,000 math-related glyphs already in the public domain or freely available from other sources, a robust and very complete set of glyphs and unambiguous Unicode code points is available for use in rendering mathematics in Web browsers and other client tools.

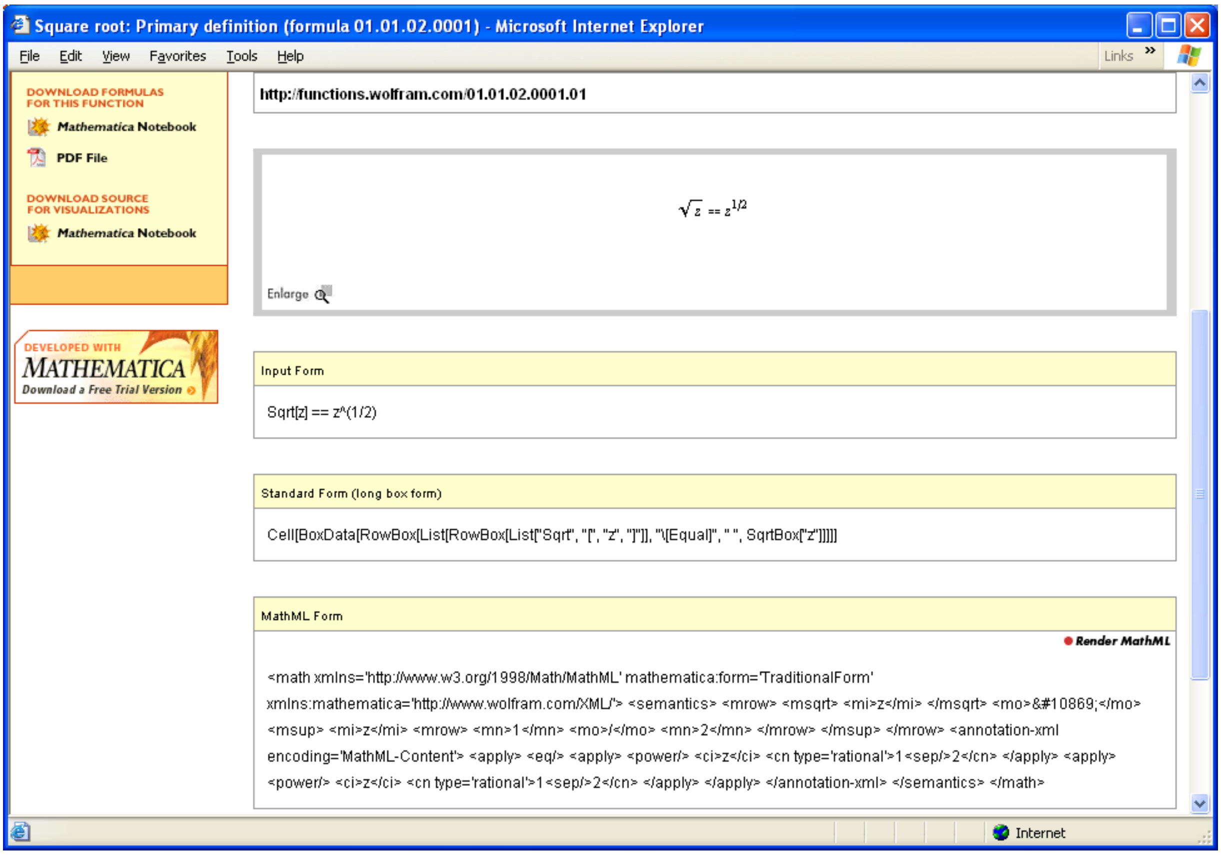
MathML also experienced some early growing pains due to differences and variations in how MathML was implemented by Internet Explorer, Mozilla-based, and the W3C's own Amaya Web browser platforms installed under various client operating systems. These differences in implementation have largely been neutralized through the efforts of the W3C Math Working group which has tested and thoroughly documented MathML implementation details in various browser platforms. To help authors of Web documents containing MathML overcome variations in Web browser support, the Working Group provides <http://www.w3.org/Math/XSL/> illustrative examples, test pages, and a transforming XSL style sheet that can be used to insure MathML implementations are compatible with the full range of popular browsers and plug-ins advertising support for MathML. By following the Working Group's guidelines and invoking their style sheet as recommended, it is possible to create XHTML documents containing MathML, which can be viewed correctly using any of the major browser platforms supporting MathML.

Finally, in addition to direct support for MathML in popular Web browsers, MathML is also supported today by a number of other software tools and applications. Most specialized editors for mathematics, including selected tools from Integre and Design Science, provide MathML support. As mentioned above, major computer algebra systems such as *Mathematica* and *Maple* provide MathML support. Such tools generally allow the import and/or export of MathML, though support may sometimes be limited to Presentation MathML only. In addition, a number of other tools, many freely available, will transform mathematics authored in $\TeX$ into MathML.

3.2 An Example: The Wolfram Functions Site

One may ask, seven years after its initial introduction, in what ways is MathML being used? The greatest inroads to date have been behind the scenes. A number of journal publishers and large content providers (e.g., the U.S. Patent Office), already committed to the use of XML-based technologies for publishing and archiving activities, have begun experimenting with the incorporation of MathML into their workflows and XML DTDs or schemas. Both scholarly society publishers (e.g., the American Institute of Physics) and commercial sector science and technology publishers (e.g., Elsevier, Blackwell) have been looking at or are incorporating MathML into publishing workflows (Wusterman, 2003). Typically, MathML (and XML itself) is used primarily or only in the back room. At this time, authors usually still submit in Microsoft Word, $\mathrm{T_EX}$, or other, non-XML formats. Final dissemination is more frequently in HTML, Adobe PDF, or some other proprietary format than in MathML, though MathML output may be an option. XML and MathML are used during editorial processing and archiving and may be used behind the scenes to support more robust search and discovery or the development of linkages between and within documents.

In some ways more interesting than the use of MathML in the traditional scholarly publication cycle are the ways it is being used in new kinds of publications engendered by the Web itself. An illustrative example is the inclusion of MathML in the Wolfram Functions Website <http://functions.wolfram.com/>. As of December 2005, this Website includes over 87,000 formulas for and almost 11,000 visualizations of canonical mathematical functions. Functions are thoroughly classified and indexed and range from the elementary to the quite complex. In addition to providing a GIF image representation of each formula (in 2 sizes), the Website provides *Mathematica* Input and Standard Forms of each formula and a MathML form of each formula. Figure 3 shows a simple formula expressed in these various forms.


Square root: Primary definition (formula 01.01.02.0001) - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Links
DOWNLOAD FORMULAS FOR THIS FUNCTION
Mathematica Notebook
PDF File
DOWNLOAD SOURCE FOR VISUALIZATIONS
Mathematica Notebook
DEVELOPED WITH
MATHEMATICA
Download a Free Trial Version
Enlarge
Render MathML
Internet

http://functions.wolfram.com/01.01.02.0001.01

$\sqrt{z} == z^{1/2}$

Input Form

```
Sqrt[z] == z^(1/2)
```

Standard Form (long box form)

```
Cell[BoxData[RowBox[List[RowBox[List["Sqrt", "[", "z", "]"]], "\[Equal]", " ", SqrtBox["z"]]]]]
```

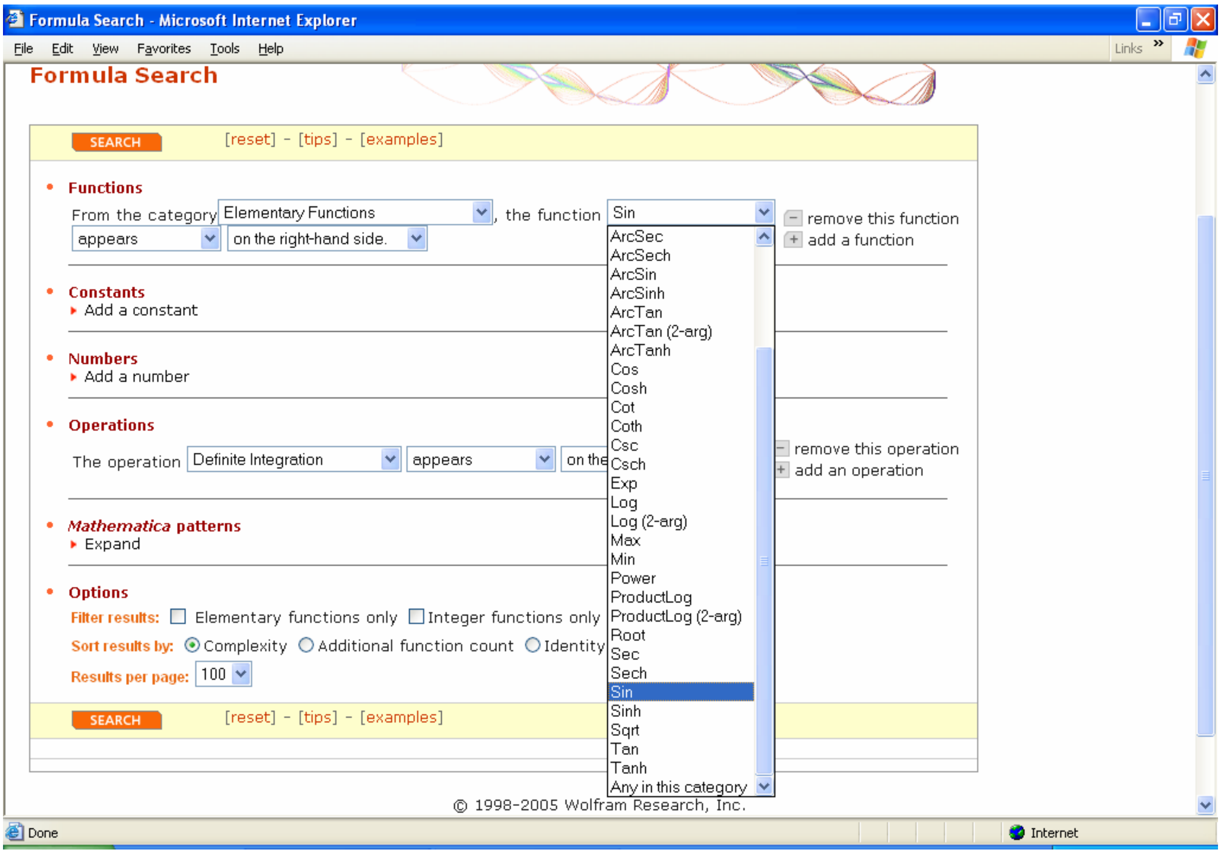
MathML Form

```
<math xmlns='http://www.w3.org/1998/Math/MathML' mathematica:form='TraditionalForm'
xmlns:mathematica='http://www.wolfram.com/XML/'> <semantics> <mrow> <msqrt> <mi>z</mi> </msqrt> <mo>&#10869;</mo>
<msup> <mi>z</mi> <mrow> <mn>1</mn> <mo>/</mo> <mn>2</mn> </mrow> </msup> </mrow> <annotation-xml
encoding='MathML-Content'> <apply> <eq/> <apply> <power/> <ci>z</ci> <cn type='rational'>1<sep/>2</cn> </apply> <apply>
<power/> <ci>z</ci> <cn type='rational'>1<sep/>2</cn> </apply> </apply> </annotation-xml> </semantics> </math>
```

Figure 3. An example of different forms of encoding (including MathML) for a simple mathematical formula.

Notice that both the Presentation MathML form for the expression (the <*mrow*> element) and the Content MathML form for the expression (the <*annotation-xml encoding='MathML-Content'*> element) are provided. To further illustrate the interactive nature of MathML, the Wolfram Functions Site provides a link ('Render MathML') to the Wolfram MathML Central Website <http://www.mathmlcentral.com/> where arbitrary valid MathML can be transformed into a GIF image, plotted, or integrated on the fly. The Wolfram Function Site is also instructive as to the range of what MathML can do. For simple function expressions, such as that shown in Figure 3, MathML is wholly up to the task. The formula can be adequately and unambiguously expressed in both Presentation and

Content MathML. At the other end of the spectrum, however, i.e., for complex formulations of complex functions, the current incarnation of MathML is not adequate. Additional hints must be embedded within the MathML to completely describe the appearance and/or meaning of the formula. MathML enables this in a couple of ways, including through an alternative use of the MathML <*annotation-xml*> element. Using this approach, XML can be borrowed from other namespaces to express more complex mathematical constructs. Thus an implementer could include semantics from the OpenMath namespace using an <*annotation-xml encoding="OpenMath" ...*> element. Support is provided within the MathML specification for doing this with XLink pointers and even including references to additional style sheets associated with the external namespace. Not surprisingly, on the Functions Site, expressions too complex for MathML alone typically include <*annotation encoding='Mathematica'*> elements. When imported into a *Mathematica*, these more subtle parts of the formula can then be understood correctly.



From the category Elementary Functions, the function Sin
appears on the right-hand side.

The operation Definite Integration appears on the

Filter results: □ Elementary functions only □ Integer functions only

Sort results by: ● Complexity ○ Additional function count ○ Identity

Results per page: 100

Figure 4. Form for searching for mathematical expressions contained in the Wolfram Functions Site.

Finally, another special feature of the Wolfram Functions Site enabled by the rich semantic description of the mathematics contained in the site is an interface for searching using mathematical expressions rather than English language text. Figure 4 shows the basic search interface for searching for mathematical expressions within the Wolfram Functions Site. Currently this functionality exploits the *Mathematica* forms of the formulas indexed (rather than the MathML forms); however, this example still illustrates what is possible by exploiting semantically rich mathematical notation forms like MathML. Search arguments are entered via pull down menus and manually entered values and patterns. Once initial results are found, users also have the option to search for "more like this one." To implement this feature, the application analyzes formulas of interest, identifying the different defining characteristics of the formula. Then a search is automatically conducted relying on the most important characteristics of the formula already found in order to find semantically similar formulas.

3.3 An Experiment in the Context of the National Science Digital Library

Some of the recently added features and MathML-related functionality of the Wolfram Functions Site were developed with the support of a grant awarded to Wolfram Research and the University of Illinois under the auspices of the National Science Digital Library program <http://nsdl.org/> (NSDL). This grant-funded project also included preliminary experiments and investigations regarding possible ways mathematical expressions and math-related text might be used to help tie published scholarly literature to Web resources, such as the Wolfram Functions Site. The testbed of published scholarly literature used for this project consisted of approximately 75,000 articles on computer science and physics published in 60 different science and engineering journal titles between 1995 and 2001. These materials were part of a larger testbed hosted at Illinois during this period as part of our DLI-1 and D-Lib Test Suite projects (Mischo & Cole, 2000). Articles were provided originally in SGML by publisher partners and had previously been converted to XML by researchers at the University of Illinois, including transformations of the embedded SGML mathematics to MathML (Cole et al., 2000).

Two approaches were investigated for this part of our NSDL grant project. In the first approach, metadata keyword phrases and function names derived from the pages of the Wolfram Functions Site were searched for in the full-text of testbed journal articles. At the time of the experiment (2004), the Wolfram Functions Site collectively contained approximately 7,500 unique keyword phrases and function names. Not all were suitable for searching in natural language text. *Mathematica* forms, numerals and phrases containing only one word (other than function names) were eliminated as not being useful for searching in natural language texts. Semantically duplicative phrases, i.e., sets of phrases where the only differences between members were word order or English language stop words, were represented in final listing of search terms by a single version. After this analysis, about 1,000 unique useful phrases culled from Wolfram Functions Site metadata remained. Just over a third (367) of these phrases appeared at least once in full-text of documents contained in the testbed. Approximately 44,000 of the 75,000 testbed journal articles contained at least one match. On average, matching documents matched two different Functions Site metadata keyword phrases. Random sampling of matches suggested that most matches were meaningful, at least to the extent that the documents did in fact make use of the function mentioned in some fashion. However, almost all of the Functions Site keyword phrases found in full-text documents linked back to more than one page in the Wolfram Functions Site. Because function names and similar descriptions can appear in multiple branches of the Functions Site hierarchy, there was no reliable way to link from document matches to the most useful node in the Functions Site. Frequently, among the limited sample of matches examined manually, the Functions Site page most relevant to one document matching a particular function name or descriptive phrase would not be the most relevant for another document matching the same name or phrase.

A second approach attempted to link journal articles to Function Site pages through MathML matches to see if authors used MathML forms of mathematical expression that could be recognized as synonymous with or equivalent to forms of mathematical functions found in the Functions Site. A sample of approximately 70,000 MathML instances was extracted from the journal testbed and searched for in the Wolfram Functions Site. Almost no matches were found. This is not really surprising. Though MathML can be used to describe mathematical expressions unambiguously, there is frequently more than one MathML form that can be used to describe the same mathematical expression. Moreover, authors may arbitrarily choose to use different letters for variable names when writing mathematical expressions that are otherwise identical or very similar in meaning. Depending on the mathematical expression, order sometimes can be varied without affecting mathematical meaning. Where distinctions such as these are not important, as in the case of trying to discover potentially useful linkages between journal articles and canonical forms of mathematical functions, extensive normalization of MathML is required. Heuristics for doing such normalization are being developed but are still relatively experimental. Additional community agreements on how to normalize MathML and on the critical "parts of speech" of mathematics grammar must be reached to enable higher levels of mathematical semantic interoperability. Such agreements also will improve our ability to search on mathematics itself. (For examples of ongoing efforts to develop better ways to support direct mathematics searching, see <http://www.ima.umn.edu/complex/spring/searching.html>.)

These results highlight a limitation of MathML. While MathML is proving useful for authoring and publishing mathematics for use on the Web, it is not yet being used in a manner consistent enough to allow recognition of documents from disparate sources that contain related mathematical expressions. For now, the natural language text that typically surrounds published mathematics is more reliable for establishing linkages between related resources.

4 ISSUES FOR MATHML GOING FORWARD

MathML version 2.0 is stable, available, and increasing in use. It is proving useful not only in the behind-the-scenes world of traditional scholarly publishing in science and technology, but also in the creation and publication of new kinds of Web-based information resources. It is well-tailored for use on the Web, and its potential for creating interactive mathematical resources, e.g., to facilitate instruction, has been demonstrated. Ancillary infrastructure in the form of publicly accessible glyphs for most mathematical notations and better support for mathematics within Unicode has been addressed by the STIX Project in conjunction with the W3C Math Working Group and the major vendors of mathematics tools and software. Because it was created and remains under the auspices of the W3C and is broadly supported by major publishers and vendors of academic mathematics software, there is a realistic expectation that it will stay up-to-date, non-proprietary, and largely backward compatible.

However there remain a number of questions and issues that are holding MathML back from wider acceptance among many individual users, especially among the larger part of those who author research mathematics. MathML on its own is not adequate to describe the most advanced, cutting-edge expressions used in research mathematics. Special extensions are needed to ensure proper representation (both visually and semantically) of more complex mathematics. As long as such extensions remain proprietary, the need to use them will detract from the interoperability benefit of MathML and will give some authors pause. Continued work on OpenMath shows promise and may eventually help address these limitations of MathML in a non-proprietary way (though that is far from certain at this point in time). Many also feel that control of mathematical presentation using MathML, though markedly better than with HTML or previous SGML mathematics schemas, still lags significantly behind $\TeX$. The same MathML encoded expression rendered in two different Web browsers can look different in small but arguably significant ways. There also remains the simple fact that $\TeX$ is seen as adequate for traditional print publication of scholarly research. Authors will change to MathML only as the benefits for instructional use and more efficient research mount and only as new forms of online publication requiring MathML begin to supplant traditional print-on-paper publication in importance as measures of scholarly achievement. MathML authoring tools also must continue to evolve and improve.

In other ways as well, MathML is at a critical stage in its evolution. Initial development of MathML was driven by the need to have a way to unambiguously and reliably express mathematics for use on the Web. However, while this need has largely been met, experience to date also shows that there are multiple valid ways of using MathML to encode the same mathematics. Determining similarities or equivalences between different MathML instances is (at present) difficult. While this issue goes beyond MathML itself, to realize the full potential of MathML as a semantically rich, Web-friendly notation for describing mathematics, more work on mathematics grammar, MathML normalization and related issues is needed.

Finally, MathML has grown up largely isolated within the academic mathematics community. Given the ubiquitous nature of mathematics across the science domains, it is critical that the sciences develop a broader consensus on how mathematics should be expressed for use with computers and on the Web. Ideally XML-based markup languages being developed for other scientific domains should either borrow from MathML directly or at least strive to implement semantically compatible ways to express mathematics, but for the most part, this is not happening. More work is needed in this area. Such efforts will need to coordinate closely not only with markup language developments in mathematics and other disciplines, but also with nascent work now underway on the development of formal mathematics ontologies and their relationships to XML-based schemes for mathematical notation.

5 ACKNOWLEDGEMENTS

Some of the work described in this article was supported under a grant from the National Science Foundation: "Second Generation Digital Mathematics Resources with Innovative Content for Metadata Harvesting and Courseware Development," NSF-DUE Award #0226327, September 2002 - August 2005, Principal Investigator: William H. Mischo, Grainger Engineering Library Information Center, University of Illinois at Urbana-Champaign.

6 REFERENCES

Cajori, F. (1928) *A History of Mathematical Notations* (2 vols.), La Salle, IL: The Open Court Publishing Company.

Cole, T. W. (2001) Publishing Mathematics on the Web. *Science & Technology Libraries* 20 (2/3), 27-44. (Published simultaneously as chapter in Schlembach, M. C. & Mischo, W. H., (Eds.), *Electronic Resources and Services in Sci-Tech Libraries*, Binghamton, NY: Haworth Press, 27-44.)

Cole, T. W., Mischo, W. H., Ferrer, R. & Habing, T. G. (2000) Using XML, XSLT, and CSS in a Digital Library. In Kraft, D. H. (Ed.) *ASIS 2000: Knowledge Innovations, Proceedings of the 63rd American Society for Information Science Annual Meeting, November 12-16*, Medford, NJ: Information Today, 430-439.

Cole, T. W. & Kazmer, M. M. (1995) SGML as a Component of the Digital Library. *Library Hi Tech* 13 (4), 75-90.

Ion, P. (1999) MathML: A Key to Math on the Web [pre-print of paper presented at 1999 $\TeX$ Users Group annual meeting]. Retrieved 13 September, 2006 from the World Wide Web <http://www.tug.org/TUG99-web/pdf/ion.pdf>.

Knuth, D. (1979) Mathematical Typography. *Bulletin (New Series) of the American Mathematical Society* 1 (2): 337-372.

Mischo, W. H. & Cole, T. W. (2000) Processing and Access Issues for Full-Text Journals. In Harum, S. & Twidale, M., (Eds.), *Successes and Failures of Digital Libraries [Papers Presented at the 35th Annual Clinic on Library Applications of Data Processing, March 22-24, 1998]*, Urbana, IL: Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign, 21-40.

Unicode Consortium. (2002) Unicode for Mathematics and Technical Publishing [Press Release]. Retrieved 13 September, 2006 from the World Wide Web: <http://www.unicode.org/press/pr-3.2.html>.

Wolfram, S. (2000) Mathematical Notation: Past and Future [Transcript of a keynote address presented at MathML and Math on the Web: MathML International Conference 2000]. Retrieved 13 September, 2006 from the World Wide Web: <http://www.stephenwolfram.com/publications/talks/mathml/index.html>.

Wusterman, J. (2003) XML and E-Journals: The State of Play. *Library Hi Tech* 21 (1): 21-33. (DOI: 10.1108/07378830310467373)