

# The Importance of MathML to Mathematics Communication

*Robert Miner*

Compared to plain text, mathematical notation is hard to represent, edit, display, and process with computers. On the surface the formalism and rigor of mathematics seem as if they should lend themselves to computing. However, advances in user interfaces and electronic communication have led us to expect that information should flow seamlessly from application to application, intelligently doing the right thing for the given context. Double-click on a picture and the expectation is that it will open in a photo editor or viewer. Drag that same image into your word processor and it should be imported automatically into the document. For mathematics notation, these expectations are particularly demanding, since math is used in a wide variety of applications with very different requirements. Not only should double-clicking on an equation open it in an equation editor, but dragging it onto a computer algebra system or graphing program should be supported as well. Similarly, math should “work” with screen readers for the visually impaired, page composition engines in publishing workflows, search engines, online testing systems, and more. Devising a worldwide software infrastructure that facilitates the use of mathematics in all these contexts is a challenging problem with a long history. Recent years, however, have witnessed a significant step forward in the form of a standardized encoding for mathematical notation called Mathematical Markup Language, or MathML.

---

*Robert Miner is director of New Product Development at Design Science, Inc. His email address is robertm@dessci.com.*

MathML is an XML-based encoding for mathematics. XML (short for eXtensible Markup Language) has emerged as the dominant data format underlying the information infrastructure of the World Wide Web. XML defines a method of representing structured data types, essentially the now familiar pointy-bracket tagging of HTML. XML itself merely defines the common syntax and leaves it to specific areas of application to define appropriate data types. MathML is one such data type. XML is significant because it is becoming deeply embedded in the software systems and workflows that will shape the information landscape for years to come. Because of MathML, mathematics is a full-fledged part of that information landscape, and this bodes well for the scientific community.

In this article we will look closer at MathML and examine some of the implications it has for scientific communication. We will describe some of the ways it is currently being put to use in publishing, e-learning, searching, knowledge management, and accessibility for the visually impaired and learning disabled. We will also describe areas of active research and suggest some of the possibilities that may lie ahead.

## **The Mathematical Software Landscape**

The challenges of dealing with math notation have long had a polarizing effect on mathematical software development. On one hand, because supporting math is difficult and expensive, generic applications such as word processors, databases and web browsers have not usually supported math directly on economic grounds. Instead, math functionality is delegated to third-party math add-ons. On the other hand, within particular communities with a strong need for math, software packages

such as  $\text{T}_\text{E}\text{X}$ , Maple, and Mathematica have arisen, with integrated, specialized support for math.

The appeal of specialized software is that it is less constrained by the requirements of a diverse user group and can therefore focus on the math processing, adding value through ease of authoring, quality of output, computational power, and so on. But the downside of specialized software is that to the extent its functionality overlaps that of generic applications, it tends to get left behind, struggling to keep up with desirable new features as they evolve in mainstream generic applications. More importantly, specialized systems tend to be difficult to interface with other software, a key consideration in networked information environments. By contrast, adding math functionality to generic applications via third-party add-ons allows math components to benefit from the functionality of the highly engineered host application, and component architectures lend themselves well to reuse in networked environments. However, the add-on component approach has pitfalls of its own, since extension mechanisms in generic applications have tended to be inadequate for math and lack of standards has hindered interoperability.

As a result, prior to the rise of the Web, the math software landscape was balkanized, consisting of a collection of specialized applications and math add-on components that couldn't easily share data or interoperate to any great degree. A number of these software packages achieved success within particular communities, such as commercial publishing, mathematical computation, and research mathematics. An obvious example is  $\text{T}_\text{E}\text{X}$ , which came closer to providing a standard electronic format for math than anything before it. To the limited extent that math applications of this era talked to each other, they probably exchanged  $\text{T}_\text{E}\text{X}$  code. However, in nonnetworked desktop computing environments before the Internet and the Web became ubiquitous, there was more emphasis on self-contained end-to-end solutions for particular tasks and less demand for interoperability.

The success of the Web, however, dramatically changes the value of information *exchange*. The process of taking information and moving it around via email, publishing it in print and in electronic format, and making it almost instantly accessible using search engines and web browsers has tremendous value and offers thought-provoking opportunities. By enabling this kind of information exchange, the Web has had a profound impact on society in general and the way scholarship is conducted in particular. As a measure of its success, we now take it for granted that by typing "soliton" and "Cameroon" into Google, within seconds one can discover that there is a physicist at the Université Yaoundé I who is interested in the topic and whose latest paper can be read online for a fee.

The technological key to information exchange in the context of the Web has been standards. Standard protocols and formats for exchanging structured data, such as HTTP, HTML, and XML are fundamental to the software architecture of the Web. In the case of math, MathML has become established as the standard format for math in XML, and this stands to benefit mathematical software development in a number of ways. Just by existing, MathML draws attention to the requirements of mathematics in XML and Web contexts, which is significant. Having a standard format also provides a clear direction for developing conversion and import and export capabilities in specialized math applications. A standard also encourages better, more uniform extension architectures in generic applications, which benefit add-on math components as well. Finally, and not least significantly, a standard decreases the risk of investment and increases the potential for return, which stimulates software development. As a measure of the effectiveness of MathML in this regard, the World Wide Web Consortium (W3C) maintains a list of over fifty MathML-aware software packages, about half of which have appeared in the last year.

### About MathML

To better understand how MathML is being used, it is useful to know a bit about MathML itself. MathML is an encoding of the visual presentation and semantic content of mathematical expressions. It was developed under the auspices of the W3C, the body that has responsibility for most Web-related standards such as HTML, XML, and many others. The initial impetus behind the Math Activity at W3C was to provide a better way of displaying equations in Web pages than as graphics. However, MathML rapidly evolved into a general means of representing and communicating mathematical expressions in XML.

MathML was first published as a W3C Recommendation in 1998, which means it is comparatively old and mature for a Web technology. MathML appeared within a few weeks of the initial release of the XML specification itself and was the first major application of XML. Because of this it has had a high profile within the XML community and has been an important test case for many subsequent Web technologies. In particular, math was a motivating example when the namespace mechanism for mixing XML markup languages in a single document was devised.

In order to support the diverse demands placed on math in different contexts, MathML is an information-rich encoding. This reflects a fundamental design decision to make low-level information explicitly available in the markup, information that is usually inferred from context by human readers. While this is usually advantageous and

sometimes essential for machine processing, it means MathML is not suitable for writing by hand. It is a text-based format, however, so it can be read by persistent humans. In this regard, it is somewhat analogous to other low-level structured formats such as PostScript, for example.

One particularly notable feature of MathML that distinguishes it from most other math encodings is that it contains two separate vocabularies, which can either be used alone or in conjunction. One vocabulary, termed *presentation markup*, describes the visual appearance of an expression. By contrast, *content markup* attempts to capture the meaning, or mathematical semantics, of an expression. An example is useful for illustrating the differences between the two. Consider the expression  $(x + 2)^3$ . The presentation markup for this expression is:

```
<math>
  <msup>
    <mrow>
      <mo>(</mo>
      <mrow>
        <mi>x</mi>
        <mo>+</mo>
        <mn>2</mn>
      </mrow>
    </mrow>
  </msup>
</math>
```

The content markup for the same expression is:

```
<math>
  <apply>
    <power/>
    <apply>
      <plus/>
      <ci>x</ci>
      <cn>2</cn>
    </apply>
    <cn>3</cn>
  </apply>
</math>
```

There are several interesting points to be made. The first is that MathML requires that all *tokens*, or indivisible units of text, be explicitly tagged to indicate their roles. In presentation markup, the markup elements `<mi>`, `<mn>`, and `<mo>` indicate identifiers, numbers, and operators respectively. Identifying the structural roles of tokens in markup is important for use with XSLT, a powerful stylesheet-driven transformation language for XML data types. XSLT has little facility for parsing text data, so XSLT-based applications would have difficulty applying standard math typesetting

conventions to tokens if it weren't for the low-level token tagging in MathML. Operator tokens are quite general in Presentation MathML. They include "grouping operators" such as parentheses as well as more traditional operators like "+". Operators can be "stretchy" in the case of notations such as arrows, bars, and parentheses.

A second notable feature of presentation markup is that it goes to some lengths to make the hierarchical markup structure reflect the underlying mathematical structure. Thus, for example, the `<msup>` element denoting the superscript construct has two child elements representing the base and the script. This is in contrast to encodings such as T<sub>E</sub>X, where the script would typically just embellish the final parenthesis. Similarly, the MathML markup introduces invisible `<mrow>` elements to group the operands together with the "+" operator. Stretchy operators such as parentheses also tend to encourage proper grouping of arguments, since they automatically stretch to the height of the enclosing element. This usually has the effect of requiring that an expression semantically grouped by parentheses is also structurally enclosed by a corresponding `<mrow>` element.

The content markup for  $(x + 2)^3$  is quite different. Here we have a functional, LISP-like representation, where operators are applied to arguments. Note that the parentheses do not appear directly in this representation. They are artifacts of a particular visual presentation of the expression. This abstraction gives one the ability to generate multiple presentations of the same expression. A common approach for associating a particular visual rendering with a Content MathML expression is to use XSLT stylesheets to transform it to Presentation MathML. As noted above, XSLT is a rule-based transformation language for XML, and it is supported by most contemporary web browsers. The Connexions Project at Rice University [1] uses this technique to allow users to choose notational preferences. Other groups have used it to localize math expressions: for example, using "tan" for the tangent function in the U.S. and "tg" in France.

An obvious limitation of Content MathML is its scope. It covers only a modest collection of mathematical concepts, covering up to roughly the first year of calculus in the U.S. curriculum. However, Content MathML defines a mechanism for extending its usage by referencing external repositories of semantic definitions. In particular, there is a close relationship between Content MathML and OpenMath, an organization which maintains such "content dictionaries" in a standard format. Content MathML and OpenMath are being used in a number of formal systems and theorem-proving research projects. The EU-funded MowGLI project [2] and Ontario Research Center for Computer Algebra are particularly noteworthy in this regard.

## Science, Technical, and Medical (STM) Publishing

As many commentators have noted, mathematics research literature has a long lifespan. An article can remain relevant for many decades. Consequently, making more effective use of the breadth and depth of the literature is an enticing possibility. By connecting ideas and thinkers through time and space, information technology has enormous potential for mathematics. Organizations such as MathSciNet, JSTOR, the arXiv, and many others have begun to suggest the outlines of what is possible. A large amount of material has already been made available in electronic format, and retro-digitization projects continue to push the frontier of electronically accessible documents back into the past. Schemes for durable references to electronic publications have been established by STM publishers, and articles and bibliographic databases have been cross-linked. To make this substantial achievement happen, STM publishers have developed cross-media publishing workflows, which involve creating and managing content in print, over the Web, in databases, and so on.

Across the publishing industry as a whole, *XML-centric workflows* have become the strategy of choice to meet the challenges of cross-media publishing. In the XML-centric model, articles are stored in XML format in a central repository and formatted for various media by stylesheet-driven composition engines. Fueled by public and private investment throughout the dot-com boom, there is widespread support for XML in current publishing software systems. But in order for STM publishers to take advantage of these XML-centric publishing systems, they must be able to deal with math, and that's where MathML comes in.

In the last several years there has been a proliferation of MathML software targeting STM publishing, reflecting a significant shift toward XML workflows amongst major content providers. STM composition software, XML editors, and conversion software have added MathML support. This has been greatly facilitated by the fact that MathML is a completely integrated XML data type that can be accessed and manipulated through standard APIs in XML software systems. This enables math add-ons to do a better job more easily and in an interoperable way.

At the same time, MathML has been incorporated into a number of XML document types used in STM publishing such as DocBook and the Journal Archiving and Interchange format used by the National Library of Medicine and its PubMed system. As a result, a number of publishers are now running pilot projects using MathML, and a few have embarked on plans to shift major workflows to an XML-centric model using MathML. John Wiley & Sons is conducting pilot projects using MathML. The Amer-

ican Physical Society and American Institute of Physics have already begun using MathML directly for production purposes. The oldest and probably largest volume MathML-based workflow is that of the U.S. Patent Office, operated under contract by Reed-Elsevier. That workflow handles thousands of equations per week. Other projects are currently gearing up, and the next year or two promise a substantial increase in MathML use in production workflows.

Gaps in software support remain. Ironically, MathML support in web browsers continues to present challenges. The newer Netscape and Mozilla-based browsers such as Firefox now have built-in MathML support, and the free MathPlayer extension from Design Science adds native-quality MathML support to Internet Explorer for Windows. But the Safari browser for Macintosh does not support MathML yet. Support for MathML in page layout programs has yet to be developed, and  $\TeX$  conversion is another area requiring further work. The Hermes  $\LaTeX$ -to-MathML translator, being developed as part of the MowGLI project, seems promising in this regard.

## Math-Aware Searching

One of the most interesting possibilities of MathML is the potential for enhancing searching of technical literature and educational material. By integrating mathematics with the surrounding document in a highly structured way, MathML opens the door to mathematical keyword searching: type an equation into a search engine and get back a list of papers in which it occurs. MathML could also play a role in automated or semiautomated creation of metadata, where the content of a document is analyzed by a software agent to suggest keywords from subject taxonomies or other metadata ontologies. In this way, MathML may have a role in enhancing existing search systems geared toward bibliographic metadata.

In most cases, current searching of online STM content is limited to keyword searches on text. As a result, a researcher typically needs to know appropriate keywords in advance to search for the desired mathematical subject matter. This is limiting in three ways. First, searching is frequently restricted to abstracts and bibliographic metadata, and appropriate keywords may not appear there, even though the full document may contain the desired information. This is especially likely for material in secondary topics, background information, and introductory material. This is the classic metadata problem of insuring appropriate keywords are accessible and appear in close proximity to the resources that they describe.

Secondly, keywords describing mathematical objects are typically overgeneral. One can search for "quadratic polynomial", but there is no effec-

tive way to narrow the search to a particular polynomial or class of polynomials. This is particularly limiting for educational resources, where the same generic label applies to many different treatments of the same material. Searching for “rate of work” with Google produces some 20,000 references. Finding out which of these documents might shed light on the particular rate-of-work problem in your child’s homework assignment is a laborious, and likely fruitless, task.

Finally, it is commonplace in technical subjects to be confronted with mathematical problems of a type beyond one’s experience for which one does not know appropriate keywords at all. A variant of this problem arises when different fields of study have different terminology for identical mathematical objects. In such instances, a problem may in fact be well understood, but the researcher has no way to discover what keywords will find the answer. While the problem of differing nomenclature also affects mathematical notation, at least in some cases the problem is more tractable.

At the Future of Math Communication II workshop [3], held at MSRI in 1999, Rob Corless of the Ontario Research Centre for Computer Algebra related an incident that makes this point well. In the course of working on a nonlinear initial value that arose in conjunction with a dynamical system he was studying, he needed to understand the behavior of a certain power series. He knew that Neil Sloane of AT&T Research had recently set up a website where one could search the *Handbook of Integer Sequences and Series* [4], so he entered the coefficients of his power series. It turned out that the series was known, and the initial value problem had been solved in generality by Gilbert Labelle in a paper in the *European Journal of Combinatorics*. Not being a combinatorialist, Corless thought it unlikely he would have found this information in any other way. In particular, Corless would not have found it using text-based keyword searches, since he did not know in advance that the solution had anything to do with combinatorics.

Of course, a key point in this anecdote is that integer sequences and series have an obvious, unique, easy-to-type canonical form that make mathematical keyword searches particularly easy in this very narrow area. However, it is important to note that many other kinds of mathematical objects also have easily computed canonical forms. Similarly, other techniques involving normalizing expressions and pattern matching can be quite effective for suitably restricted kinds of mathematical content, especially when augmented by metadata. Consequently, it is reasonable to suppose that mathematical search technology can be extended without undue effort to the point where search success stories of the kind Corless describes

could at least be replicated for many categories of content, if not in full generality.

However, to scale up mathematical searching and integrate it with text searching to any appreciable degree, the ability to automatically identify and normalize large classes of mathematical expressions is essential. Here MathML plays an important role in two ways. First, since it takes pains to insure markup structure generally reflects mathematical structure, it significantly simplifies the computational complexity of recognizing and manipulating expressions. This facilitates the creation of specialized, math-aware search engines. Second, by integrating math with text in a common XML-based format, MathML makes math accessible to generic XML search technologies. The XQuery standard currently under development, for example, is expected to make a substantial impact in this area. Because of their obvious commercial potential, generic XML search technologies are attracting widespread investment and support, so the potential benefit to STM searching is highly leveraged.

The potential benefit of math-aware searching has been recognized in many quarters. The National Science Foundation awarded Design Science a grant through the National Science Digital Library (NSDL) program to investigate ways of enhancing math searching. The MowGLI project is also investigating applications of MathML to math searching. A number of other projects, both commercial and academic, are under way. For example, a workshop on the topic, funded by Design Science, was held at the Institute for Math and its Applications at the University of Minnesota in April 2004 [5]. Searching and related topics also featured prominently at a conference on Mathematical Knowledge Management, held in conjunction with the Joint Mathematics Meetings in January 2004.

## E-learning

E-learning is another area where MathML has natural applications. In many ways, MathML has its roots in online learning, as much of the original motivation for MathML was to provide a better means of incorporating mathematics into Web pages for educational purposes. However, in practice, MathML has probably had a larger impact to date as a backend technology used to add math support in course management systems, online assessment systems, and the like.

From the outset, the Web has had a strong appeal as a medium where educational content can be dynamic and interactive and where concepts can be presented in multiple educational modalities: text, images, sound, animation, and even manipulatives and simulations. Much effort has been invested in the creation of such content, in many cases with highly engaging and effective results. Different projects have used a wide variety

of Web technologies, including some that make use of MathML. One of the larger projects making extensive use of MathML to create dynamic educational content is the Homework Help feature of Microsoft's MSN Premium service.

However, highly dynamic educational Web content is problematic in several ways. First, it is quite difficult to create. It requires not only skill in educational design but also substantial technical skill. Dynamic Web content also places great demands on bandwidth and browser technology. But education is precisely where one finds the broadest diversity of browsers and platforms and where the requirement of universal access is strongest. As a consequence, efforts to use the Web as a means of integrating rich media into math curricula have generally had mixed success.

At the same time, use of the Web to provide a learning environment where students and teachers can interact via various modes of electronic communication has grown to the point where it is now commonplace. Such uses range from email and simple course homepages, where syllabi, assignments, and office hours are posted, to sophisticated learning management systems (LMS) such as WebCT, Blackboard, eCollege, and others. Because electronic communication has an entirely different dynamic than face-to-face interactions, it can lower barriers to participation for students who otherwise might sit silent and unnoticed in the back row. Some students may be more comfortable interacting with teachers and their peers via electronic means, where there is a slight element of anonymity and the ability to consider questions and responses without the real-time pressure of face-to-face interaction.

MathML is now widely used behind the scenes for adding math support to online collaboration tools. To a large extent, this has been a consequence of the fact that MathML fostered the development of a number of interoperable math-aware components that LMS vendors have been able to utilize. WebEQ, webMathematica, techexplorer, and MathIWYG are among the most common. In particular, a number of LMS vendors now provide math support in their whiteboard and message board systems using these components. Typically such systems use a MathML equation editor component that can be embedded in a Web page for authoring, together with server-side components for processing MathML for display.

Another area where MathML is being used is online assessment. For lower-level mathematics, where MathML is most successful at capturing mathematical semantics, several systems utilize computer algebra systems and other techniques to perform automatic scoring. Some systems, such as MapleTA, are even able to analyze student errors to provide adaptive hints and customize tutorials

to target student skill deficits. In a similar vein, Integre Technical Publishing is investigating ways of using standard XML tools to take advantage of the structured nature of MathML to analyze student work through an NSF-funded NSDL grant.

### Accessibility

One last area where MathML is making a noteworthy contribution is accessibility for the visually impaired and learning disabled. As the work force has aged, disabilities and impairments of all sorts have grown to affect nearly two-thirds of adults. Most are mild, but, according to a survey by Forster Research commissioned by Microsoft, 17 percent of computer users have a mild visual difficulty or impairment and 9 percent have a severe visual difficulty or impairment.

The needs of users vary substantially depending on the nature and degree of the impairment. Individuals with severe visual impairments often rely on tactile feedback through braille displays and embossers, as well as audio renderings of the math. Individuals with low vision typically require audio renderings, in conjunction with conventional typeset representations using large font sizes or high-contrast colors. Those with learning disabilities such as dyslexia benefit most from synchronized highlighting of visual display along with an audio rendering.

In the United States, the American Disabilities Act (ADA), the Individuals with Disabilities Education Act (IDEA), and other federal legislation require schools and publishers to provide accessible versions of texts in many circumstances. A majority of states have similar laws mandating accessibility of content for organizations that receive state funding, and a majority of states require that textbook publishers provide versions of their textbooks for the blind. Similar laws also apply in Europe and elsewhere. Currently, in the case of mathematics, these requirements are typically met by providing a text equivalent for equations. For example, in an HTML page, this typically means equations are displayed using images, with a textual ALT description of an image.

Unfortunately, text descriptions of mathematical expressions only meet the letter of the law and do not really address user needs. At a practical level, the preparation of text descriptions is labor intensive and error prone. At a deeper level, for audio rendering of mathematics, the ability to "navigate" around a long expression is critical to comprehension. Moreover, static text cannot take advantage of locale or user preference information to choose the language or customize the vocabulary.

MathML facilitates solutions to all of these problems. Most assistive technology utilizes standard software APIs developed for HTML and XML. By making the math notation available through these

standard methods, MathML enables screen readers and other assistive technologies to properly handle the math with minimal additional effort. And since MathML tries to insure that markup structure reflects the underlying semantics, navigation is also greatly facilitated. Finally, since the audio rendering of a MathML expression is generated on the client machine, it can take full advantage of locale and user preference information.

Design Science has received an Small Business Innovation Research grant from the National Science Foundation to add accessibility functionality to its MathPlayer extension for Internet Explorer. MathPlayer 2.0 has a demo “speak expression” feature that works with leading screen readers, and support for expression navigation and synchronized highlighting is under way. A number of other groups are also exploring ways of using MathML for accessibility, and it is anticipated MathML will soon be incorporated into standard XML formats for accessibility currently under development.

## Conclusion

From a certain point of view, MathML is merely another data format for math notation—not the first, and assuredly not the last. However, as has so often been the case in the history of technology, the larger significance of MathML depends only tangentially on its particular strengths and weaknesses as a technology. Instead, MathML is significant because of the opportunity it represents for math and science to participate more fully in the information revolution that is one of the great intellectual movements of our time. MathML reserves a place at the XML technology table for the interests of math and science.

Many groups have already seized upon the opportunity presented by MathML and are using it in innovative ways in STM publishing, searching and knowledge management, e-learning, and accessibility. Moreover, momentum is building, as development efforts cross-pollinate and reinforce one another. While many of these projects are still in their early stages, the future generally looks bright for electronic communication in the sciences. In a world where math phobia is pervasive and the cold calculus of the marketplace rarely favors academia, that is no small achievement.

## References

- [1] The Connexions Project, <http://cnx.rice.edu>.
- [2] MowGLI, <http://mowgli.cs.unibo.it>.
- [3] The Future of Mathematical Communication II, [http://www.msri.org/activities/events/9900/fmc99/fmc\\_ABS.html](http://www.msri.org/activities/events/9900/fmc99/fmc_ABS.html).
- [4] NEIL J. A. SLOANE, The On-Line Encyclopedia of Integer Sequences, <http://www.research.att.com/~njas/sequences>.
- [5] Enhancing the Searching of Mathematics, <http://www.ima.umn.edu/complex/spring/searching.html>.