Mathematics on the Web with MathML

Max Froumentin
World Wide Web Consortium
mf@w3.org

Abstract

This paper paper discusses the publication of mathematics on the World Wide Web, in particular using the Mathematical Markup Language (MathML), an open standard designed to overcome the difficulties encountered so far in putting mathematics on the Web. After a summary of previous methods of publishing scientific documents on−line, the language itself is presented. The following sections show how MathML makes mathematics accessible for all browsing environments, is interoperable with related standards, and benefits from very good implementation and community support.

Introduction

Although mathematical notation has made its way into computer−generated documents with quality publishing software now correctly handling mathematics, showing mathematics on the Web has been a problem since the beginning of HTML since support for displaying equations was minimal. Shortly after the birth of the Extensible Markup Language (XML) in 1998, the World Wide Web Consortium (W3C) started a mathematics activity in order to design MathML, an XML language for expressing mathematics, with the goal to make it possible to display formulas in Web browsers as well as to provide a interchange format for mathematical software. MathML 2.0 became a W3C Recommendation in February 2001 and is now widely supported, making it at last possible to distribute scientific material on the Web. The rest of this paper explains why and how.

Mathematics on the Web

Mathematical notation inherits from centuries of refinement, resulting in strict rules regarding the layout of mathematical equations and formulas. These rigorous rules make it a challenge to design satisfactory mathematics typesetting software. From the beginning of the Internet the common practice for scientists was to exchange equations ans formulas in some encoded form based on the ASCII character set. Later graphical displays became popular while computer−aided publishing software grew more and more capable and allowed better−looking presentations of mathematics. This culminated with the advent of TeX, currently the de−facto standard for exchanging scientific documents.

As the use of the Internet for scientific use increased, in academia first and then in industrial environments, it became important to include mathematical representation in Web pages. However, the HyperText Markup Language (HTML) was not designed to describe a complete set of tags for mathematical notation, as it only defined subscripts and superscripts, such as `x <sup>2</sup> + y <sub>j</sub>` for: $x^2 + y_j$

Content authors then went back to rendering mathematics using "ASCII art", such as

```
i
----  j
\  -1
/  -------------
----  (D-j)j!(i-j)!
j=0
```

or they would include picture files of TeX renderings in their page:

$$\sum_{j=0}^{i} \frac{-1^{j}}{(D-j)j!(i-j)!}$$

The first option was quickly dropped, in favour of the second. Indeed, software automatically converting LaTeX documents to HTML handle the rendering of equations in that manner.

Although this ad–hoc way of showing mathematics in Web pages was the best alternative until not so long ago, it is breaking fundamental principles of the Web: accessibility, usability and device–independence. Indeed, although this equation probably displays correctly if one is reading this article either on a paper printout or on a standard browser, it is probably not readable if one is reading it using a text or a Braille browser. Or, if the document is read on a portable device (not such an improbable scenario with today's scientific calculators), the image might be too big to render on a small screen.

Even with a desktop computer, the user's abilities or preferences may require that the document be displayed using a bigger font or using a different colour. Moreover, even if most people will still be able to understand the equation from the picture, software such as database indexers will simply ignore it, dismissing the pleasant possibility of having one's equations being found on the Web through the use of a search engine.

The MathML Language

Shortly after XML was created to facilitate the use of structured documents on the Web, a W3C Working Group was chartered to create an XML language to describe mathematical notation: MathML. MathML version 1.0 was published in 1998, while version 2.0 was released in February 2001, adding new markup, a Document Object Model, and updated character sets.

The MathML specification defines an XML document type to describe mathematical equations and formulas, introducing markup elements such as <msqrt> or <mn> to annotate text with mathematical semantics (see examples below). The obvious advantage over using a picture of a particular equation in a Web page is that the equation is now encoded in the text of the page, and can now be processed in order to solve the problems outlined in the previous section. A MathML–enabled browser, upon loading the page, will be able to render an equation appropriately, following accessibility principles: the MathML specification does not mandate a particular rendering for a given element, leaving it to the browser so it can be customised according to the device capabilities, or the user's preferences or language.

The specification defines two sets of elements and attributes: Presentation MathML describes an equation somewhat similarly to the way one would read it, defining elements such as subscripts, fractions or operators. Although the syntax is quite different, markup elements are similar to those of TeX, but it adds a few additional tags to mark identifiers, numbers and operators. For instance the equation $1+\sqrt{b}$, written in TeX as `$1+\sqrt{b}$`, corresponds to the following Presentation MathML markup:

```
<math> <!-- starting an equation -->
  <mrow> <!-- starting a row -->
  <mn>1</mn> <!-- 1: a number -->
  <mo>+</mo> <!-- +: an operator -->
  <msqrt><mi>b</mi></msqrt> <!-- the square root of identifier b -->
  </mrow>
</math>
```

The other type of MathML markup, called Content MathML, is meant to carry more information on the equation, in particular for exposing the semantics of functions. Our example could be written as:

```
<math>
  <apply>
  <plus/>
  <cn>1<cn>
  <root>
  <degree><cn>2</cn></degree>
  <ci>b</ci>
  </root>
  </apply>
</math>
```

Content MathML adds more meaning to its description of a formula, consequently allowing its use as an interchange format between mathematical software. However the range of mathematics covered by this markup is necessarily limited, and it was chosen to include the basic set of most standard areas of mathematics, such as arithmetic, algebra, logic, set theory, calculus, sequences and series, linear algebra, and statistics. Extension mechanisms are defined to complete this list with additional mathematical constructs. On the other hand, while conveying less semantics, Presentation MathML provides a way to represent an equation from almost all areas of mathematics. Hence the choice of a particular flavour of MathML depends on the area of mathematics concerned but also on any subsequent processing of the document. It is also possible to combine the two, giving alternative descriptions for a piece of mathematics.

MathML and the Web

Because MathML is based on XML and other W3C specifications such as Cascading Style Sheets, it is fully integrated to standard Web technologies and solves the problems encountered hitherto: the rendering of MathML can be adapted to either the device used, such as a desktop computer, a calculator or a Braille device, or to the user's preferences and abilities (font size, colour) through the use of CSS, without modifying the original markup. For instance a user can configure their client to display an equation using a 24–point font by specifying in their CSS user stylesheet:

```
math { font-size: 24pt }
```

In an environment that supports CSS, the mathematics will be displayed according to the user specified rules, overriding the author's settings.

Moreover MathML markup can be annotated using additional elements in order to add more information and facilitate operations such as Web searches for a particular piece of mathematics. Finally MathML supports standard hyperlinking mechanisms. Part of an equation can be linked to another document that explains how it was derived. Rendered mathematics can also be interactive. For example an expression can be 'folded', i.e. a renderer might allow a reader to toggle between an ellipsis and a much longer expression that it represents.

Another standard upon which MathML is built is Unicode, a character repository including alphabets from most of the world languages as well as hundreds of mathematical symbols. Unicode characters and symbols can either be included directly within the markup with an unicode–enabled authoring tool or using 'character entities', such as `α`

Yet another standard with which MathML was designed to be interoperable is OpenMath, the open standard for the exchange of mathematical objects between applications. For example the MathML element `csymbol` can be used to insert a symbol within an equation, the semantics of which are specified by a link to an OpenMath URL:

```
<csymbol encoding="OpenMath"
  definitionURL="http://www.openmath.org/cd/setname1.ocd">
N
</csymbol>
```

OpenMath's XML syntax can also be included in annotation markup, to provide a semantic mapping between a given piece of MathML and other mathematical notations. The example below shows the use of the `annotation` and `annotation-xml` elements to map the semantics of the Presentation MathML for 123/456 to the notations used by Mathematica, TeX, Maple, MathML Presentation and OpenMath.

```
<semantics>
  <apply>
  <divide/>
  <cn>123</cn>
  <cn>456</cn>
  </apply>
```

```
    <annotation encoding="Mathematica">
    N[123/456, 39]
    </annotation>
    <annotation encoding="TeX">
     $0.269736842105263157894736842105263157894\ldots$
    </annotation>
    <annotation encoding="Maple">
    evalf(123/456, 39);
    </annotation>
    <annotation-xml encoding="MathML-Presentation">
    <mrow>
    <mn> 0.269736842105263157894 </mn>
    <mover accent='true'>
    <mn> 736842105263157894 </mn>
    <mo> &OverBar; </mo>
    </mover>
    </mrow>
    </annotation-xml>
    <annotation-xml encoding="OpenMath">
    <OMA xmlns="http://www.openmath.org/OpenMath">
    <OMS cd="arith1" name="divide"/>
    <OMI>123</OMI>
    <OMI>456</OMI>
    </OMA>
    </annotation-xml>
  </semantics>
```

This allows interoperable extension of Content MathML to potentially infinite areas of mathematics covered by OpenMath.

Implementations

At the time of writing, about 25 implementations of MathML exist. They are listed on the W3C MathML implementations page and fall into 5 categories:

- Renderers take a piece of MathML markup and transform it into something a user can comprehend (visually, vocally, etc.). Most major Web browsers in particular can display MathML, either through native rendering (e.g. Netscape 7) or via plug–ins (e.g. Internet Explorer or Opera). Other renderers exist that produce screen displays or document formats such as PDF, DVI, RTF (for Microsoft Word)
- Editors also exist, such as MathType, Publicon, or Amaya. They all provide a WYSIWYG interface for editing equations, and produce MathML content automatically, avoiding the troubole of editing MathML's somewhat verbose syntax. Amaya, from W3C, is detailed below.
- Most of the well–known mathematics software vendors have participated in the creation of the MathML standard and provide MathML input and output in their products. Among others: Mathematica, Scientific Workplace, Maple, Mathcad, MathType
- Converters. A lot of software can be made interoperable with MathML through conversion of native mathematical notation to MathML. Of particular interest is

TeX, since it is of widespread use for typesetting a lot of today's mathematics, with the consequence that a lot of scientific documentation exists only in that format. Conversion from TeX to Presentation MathML is not trivial though, as the latter has additional structure, such as delimitation of numbers, operators and identifiers. These three types of elements are usually rendered differently (e.g. identifiers in italic type), something that TeX usually infers (e.g. a letter is an identifier). Euristics have to be written to automatically do these inferences, as well as others. Many tools doing so are under development today (most linked to on the MathML implementations page).

Amaya

Amaya is W3C's open source browser/editor which can be used to interactively edit HTML/XHTML pages with embedded MathML (see screenshot below), or SVG (Standard Vector Graphics)

This paper was written using Amaya, and was both published on the Web and printed without requiring conversions. In the appendix is a screenshot of Amaya showing how equations are interactively entered. Two additional helper views, below and to the right of the equation show the XML source generated as well as a structure view of the equation.

Conclusion

MathML is on its way to success. A year after version 2 of the language was published, many implementations support it, Web browsers and editors in particular, and show that it is the de–facto standard for on–line mathematics. Although the specification itself is finished, more work needs to be done and the W3C math working group is not at rest. Conversion of other mathematical formats to MathML is an important task to enable publishing documents on the Web. Tools need to be developed in order to convert from other popular standards. Large amounts of legacy documents have to be made available on–line in order to make the success of MathML complete.

Links

- The W3C Math home page: http://www.w3.org/Math
- MathML implementations page: http://www.w3.org/Math/implementations.html
- The MathML 2.0 Recommendation: http://www.w3.org/TR/MathML2
- O. Caprotti, D. P. Carlisle, A. M. Cohen (editors); The OpenMath Standard, February 2000: http://www.openmath.org/standard
- Amaya. Open source software from W3C. Available on Linux, Windows and Macintosh: http://www.w3.org/Amaya

Appendix: Snaphot of Amaya's MathML editor