Math on World Wide Web

Marat KALIBEKOV [*]

Niyazi ARI [**]

Abstract

Introducing mathematical content in World Wide Web is a problem for many people. In this paper we will explain and show you a W3C recommendation – MathML. MathML is a one way to express a mathematical content in a structured plain text form, which means, that it can be easily recognized, indexed, edited, parsed, rendered and transformed. Based on XML vocabulary and supported by (natively and via plugins and addons) Internet Explorer and Mozilla Firefox, MathML has a prospective future.

Key Words: Web, Math, MathML, XML

Introduction

Mathematical equations and expressions often have a very difficult structure, not only in way of writing, but also in way of displaying them. Mostly rendering problems stay as a sticking point for mass integration of mathematical content in WWW. Users of WWW are presented with Web browsers. Lack of compability among browsers also one problem of math on Web. There were many attempts to standardize math representation or give it format that will be accepted widely. But one of the most successful was MathML from WWW Consortium. Rendering mathematical expressions has a specific nature, because it means to use specific fonts with many specific options (underlines, spacings, overlines, sub and super scripts, roots, scaling of symbols, and so on). Currently available font solutions (packages) either are not free, neither are constrained with some specific license, which is complicates and slows of mass integration of MathML in browsers. But many scientific organizations joined to develop a set of fonts with free license to represent scientific data, which means mathematical content too. This project was named as STiX fonts (homepage is http://www.stixfonts.org). At the moment of writing this document, they planned to release a first stable version in the April, 2007. Many companies are interested in this project, because it can be a good basis for ideas that couldn't be implemented due to realization difficulties related with fonts.

About MathML

As we said earlier, MathML is the most successful format for representing mathematical content on WWW from W3C. At least W3C is an authoritative organization that specializes in standardizing WWW and bring some standards and recommendations that will be accepted by programmers (or organizations that stay behind them), who write their applications for browsers and who develop browsers. At the moment of writing this document, the last version was 2.0, but works toward version 3.0 was in progress.

MathML is a XML vocabulary that is aimed to encode (and explain – in next articles) mathematical expressions in plain text. Encoded in such way (in XML), content can be easily indexed, transferred, converted, recognized, generated or at least reprinted from the paper. It is not a binary format, it is a feature than disadvantage. As a basis for all markup languages, it is supposed, that client that processes it (in our and in most cases it is a Web browser on computer, your cell phone or PDA) will understand it (support the MathML recommendation) and render it in right way. But how expressions must be exactly rendered is out of scope of this article, we will concentrate only on encoding mathematical expressions.

Preparing environment for working with MathML

Mozilla Firefox (and Mozilla Suite)

Mozilla Firefox beginning with the version 1.5 has a native support to MathML. But as the said above, due to license limitations fonts must be installed separately. All information can be found on Mozilla MathML fonts

[*] MSc, Kazakhstan Suleyman Demirel University

[**] Prof.Dr., University of Technology, Switzerland

project – http://www.mozilla.org/projects/mathml/fonts. Simply you must download and install new fonts for this version.

Internet Explorer

IE doesn't have native support for MathML, but there is a lot of plug-ins, that add MathML rendering to IE. But we will discuss only two of them, because they are easier to install and to use.

1. DesignScience MathPlayer - http://www.dessci.com

Tools are so easy to use these days; just download and install it. No extra efforts are needed. After installing this plugin, Microsoft IE will correctly process MathML data.

2. Integre techexplorer Hypermedia Browser - http://www.integretechpub.com

This product also easy to install and use, but more steps are needed to download it – you must provide purpose of need and register. Because of that all these products are not 'free'.

So if we are taking into an account ease of installation, chose one of mentioned plug-ins. Hopefully things will be easier after IE will support natively MathML.

Conclusion on browsers

There are also other browsers that can work with MathML. We stopped on these two browsers, because they are most used browsers and ones in which some work on integrating MathML is done. You can try a W3C browser – Amaya (http://w3.org), but it is not so good as Mozilla's products or IE, and far from daily normal usage.

Starting with MathML

This part of MathML is directed to displaying math visually, not interested in the meaning of expressions.

Presentational MathML consists about 30 elements which accept over 50 attributes. These elements can be divided into two parts, token elements and layout schemata, which are explained later. But before continuing, let consider few examples to have an idea of MathML.

Inserting MathML in XHTML

Having this knowledge, you now can easily start to writing your scientific documents, that can be viewed on WWW. Put only your MathML in math tag and in XHTML file of such format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
"http://www.w3.org/Math/DTD/mathml2/xhtml-math11-f.dtd">
<?xml-stylesheet type="text/xsl" href="http://www.w3.org/Math/XSL/pmathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title> ... </title></head>
<body>
<math xmlns="http://www.w3.org/1998/Math/MathML">
<msqrt><mi>x</mi></msqrt>
</math>
</body>
</html>
```

Save it with the extension xml and open in your browser. This test example should show you a variable x under square root.

Example 1: [1]

```
<msup>
<mi>x</mi><mn>2</mn>
</msup>
```

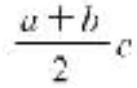
These examples show us how to display power of some number. Here tag mi represents mathematical identifier and tag mn represents number. Tag msup takes it's first element as a base, and a second element as a power. So here variable x to the power of 2 must be displayed.

[1] Now in examples we will give only a parts of markup, to see how it will be viewed on browsers, do steps mentioned in "Inserting MathML in Web-pages"

Example 2:

```
<mfrac>
<mrow><mi>a</mi><mo>+</mo><mi>b</mi></mrow>
<mn>2</mn>
</mfrac>
<mo>&InvisibleTimes;</mo>
<mi>c</mi>
```

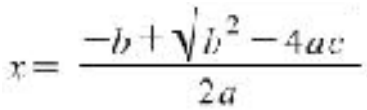
Here three new constructions are used, mfrac is used to present fractions, where two parameters as children are needed. First one is numerator, and second one is denominator. mrow is a special tag that is used to group logically complex expressions into one. ⁢ MathML character entity is used here to indicate that a special spacing rules between fraction and c are needed and fraction and variable c should not be broken onto separate lines.



Example 3.

```
<mi>x</mi>
<mo>=</mo>
<mfrac>
<mrow>
<mrow><mo>-</mo><mi>b</mi></mrow>
<mo>&PlusMinus;</mo>
<msqrt>
<mrow>
<msup><mi>b</mi><mn>2</mn></msup>
<mo>-</mo>
<mrow>
<mn>4</mn><mo>&InvisibleTimes;</mo><mi>a</mi><mo>&InvisibleTimes;</mo><mi>c</mi>
</mrow>
</mrow>
</msqrt>
</mrow>
<mrow>
<mn>2</mn>
<mo>&InvisibleTimes;</mo>
<mi>a</mi>
</mrow>
</mfrac>
```

Try to guess, what is encoded here? I think most of the code is the self explanatory. Here comes a problem with XML, which I mentioned above. This is simple expression, but takes a relatively big amount of coding.



MathML explained

Now let go inside presentational MathML:

As mentioned above, presentational MathML consist of two parts: Token elements and Layout schemata.

1. Token elements:
2. Layout schemata:

Token elements:

1. **Identifiers**
   Text inside mi, must be rendered as an identifier. Identifiers can include variables, function names, and symbolic constants.
2. **Numbers**
   Text inside mn represents “numerical literal” or other data that should be rendered as an numerical literal.
3. **Operators**

An mo element represents an operator or anything that should be rendered as an operator.

Example 4:

```
<mrow>
<mo> ( </mo>
<mrow>
<mi> a </mi>
<mo> + </mo>
<mi> b </mi>
        </mrow>
        <mo> ) </mo>
        </mrow>
```

$(a+b)$

Example 5:

```
<mrow>
<mo> [ </mo>
<mrow>
<mi> x </mi>
<mo> , </mo>
<mi> y </mi>
</mrow>
<mo> ) </mo>
</mrow>
```

$[x, y)$

Example 6:

```
<mrow>
<mi> f <mi>
<mo> &ApplyFunction; </mo>
<mrow>
<mo> ( </mo>
<mrow>
<mi> x </mi>
<mo> , </mo>
<mi> y </mi>
</mrow>
<mo> ) </mo>
</mrow>
</mrow>
```

$f(x, y)$

4. **Invisible operators**

Certain operators that are invisible in traditional mathematical notation should be represented using specific entity references within mo elements, rather than by simply nothing. Examples are ⁢ , ⁡ , ⁣ . For full list you can always refer to specification of MathML.

Layout schemata:

1. Horizontally Group Sub-Expressions

An mrow element used to group together any number of sub-expressions, usually consisting of one or more mo elements acting as “operators” on one or more other expressions that are their “operands”. There was enough examples above, I think.

2. Fractions

The mfrac element is used for fractions. It can also be used to mark up fraction-like objects such as binomial coefficients. mfrac element takes as parameter two elements, where first is numerator and second is denominator. There was two examples above about fractions.

3. Radicals

There are two elements for radicals – msqrt and mroot. Elements inside sqrt rendered as it is under square root. mroot takes as parameters exact two elements, where first one is expression under root, and the second is the base of this radical.

Example 7:

```
<mfrac>
<mi>x</mi>
</mfrac>
<mo>=</mo>
<mfrac>
<mrow><mi>x</mi><mo>+</mo><mi>4</mi></mrow>
<mn>3</mn>
</mfrac>
<mo>=</mo>
<mfrac>
<mrow><mi>x</mi><mo>+</mo><mi>12</mi></mrow>
<mi>n</mi>
</mfrac>
```

$$\sqrt{x} = \sqrt[3]{x+4} = \sqrt[n]{x+12}$$

4. Expressions inside pair of fences

The mfenced element provides a convenient form in which to express common constructs involving fences, possibly including separators (such as comma) between the arguments.

Attributes of mfenced are open, close and separators. They are used for showing opening part, closing part and type of separator respectively. Defaults are opening brace - '(', closing brace - ')' and comma - ','.

Example 8:

```
<mfenced>
<mi>x</mi>
</mfenced>
```

$$(x)$$

Example 9:

```
<mfenced>
<mi> x </mi>
<mi> y </mi>
</mfenced>
```

$$(x, y)$$

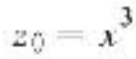
5. Sub and superscripts

msup and msub presents a super and a sub scripts respectively. These elements take as an argument two operands, base and a super or sub script respectively.

Example 10:

```
<msub><mi>z</mi><mn>0</mn></msub>
<mo>=</mo>
```

<msup><mi>x</mi><mn>3</mn></msup>

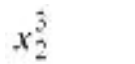


Sub and subscript simultaneously

If you need sub and super script simultaneously, you can use element msubsup. This element takes as operands three elements, where first is base, second subscript and third, last one is superscript.

Example 11:

<msubsup>
<mi>x</mi>
<mn>2</mn>
<mn>3</mn>
</msubsup>

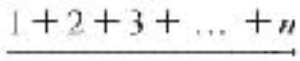


7. Underscripts

With the help of element munder, you can put some operator under needed expression. It is done in such a way, in munder tag you put your expression, then type of operator that is must be under needed expression.

Example 12:

<munder>
<mrow>
<mn>1</mn><mo>+</mo><mn>2</mn><mo>+</mo>
<mn>3</mn><mo>+</mo><mo>…</mo><mo>+</mo><mi>n</mi>
</mrow>
<mrow><mo>→</mo>
</mrow>
</munder>

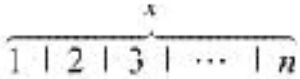


8. Overscripts

With the help of element mover, you can put some operator over needed expression. It is done in such way, in mover element you put your expression, then the type of operator that is must be an over needed expression.

Example 13:

<mover>
<mrow>
<mover>
<mrow>
<mn>1</mn><mo>+</mo><mn>2</mn><mo>+</mo>
<mn>3</mn><mo>+</mo><mo>⋯</mo><mo>+</mo><mi>n</mi>
</mrow>
<mo>⏞</mo>
</mover>
</mrow>
<mrow><mi>x</mi></mrow>
</mover>



9. Tables and matrices

Tables and matrices represented in MathML with the help of mtable element, where this element can contain only mtr and mlabeledtr elements, which are refer to rows (usual and labeled respectively). And rows can contain only elements mtd, which refers to table entries. This is very similar to HTML tables.

Example 14:

<mtable>

```
<mtr>
<mtd><mn>1</mn></mtd><mtd><mn>2</mn></mtd><mtd><mi>y</mi></mtd>
</mtr>
<mtr>
<mtd><mn>2</mn></mtd><mtd><mn>1</mn></mtd><mtd><mn>2</mn></mtd>
</mtr>
<mtr>
<mtd><mi>x</mi></mtd><mtd><mn>2</mn></mtd><mtd><mn>1</mn></mtd>
```

$$\begin{matrix} 1 & 2 & y \\ 2 & 1 & 2 \\ x & 2 & 1 \end{matrix}$$

```
</mtr>
</mtable>
```

Limitations of MathML

As a MathML is a XML vocabulary, it brings the same problems as XML. The main problem is the markup code complexity and its size. It is possible to write and edit a document with few equations by hand in your favorite text editor, but it will be a problem if your document contains a lot mathematical expressions. It can be very hard to be not confused in this code. But if you are using a good WYSIWYG editor, which supports MathML, it is a joy to write mathematical texts. But knowledge of MathML syntax is needed to have a understanding what is going on and maybe to solve some problems which can occur authoring scientific documents. Hopefully after the availability of free versions of widely usable MathML products, viewing and writing mathematical and scientific documents will be standardized.