2.9.17 MathML

MathML [1] is an XML representation for mathematical objects, allowing expressions to be stored in databases, transmitted between applications and operated upon by programs. MathML can be used to express mathematical content in web pages and digital libraries, and has become an accepted form for input and output of computer algebra systems.

With MathML available in a broad range of software tools, we can now share mathematical data in new ways. For example, with a judicious choice of software, one can now cut an expression from a web page, paste it into an E-mail message to a colleague, who can then use it in a computer algebra system, and paste the result into a patent application.

MathML provides a vocabulary for such things as identifiers, numbers, operators, grouping *etc.* There are two broad classes of constructions: The set of elements that describe the appearance, or notation, of an expression form what is called *presentation MathML.* The elements that describe the meaning, or semantics, of an expression are known as *content MathML.*

Presentation MathML

MathML has a quite complete set of elements to describe mathematical notation. There are primitives for various kinds of tokens, and others to describe relative position and grouping of subexpressions. Together, these notational primitives form a subset known as "presentation" MathML. A few examples of presentation MathML are given in Figure 1.

To illustrate the basic concepts, we examine the presentation MathML for the expression $x^2 + y^2$:

```
<mrow>
    <msup> <mi>x</mi> <mn>2</mn> </msup>
    <mo>&times;</mo>
    <msup> <mi>y</mi> <mn>2</mn> </msup>
</mrow>
```

Here the `<mi>` elements give math identifiers (variables or parameters), the `<mn>` denote numbers and the `<mo>` denotes an operator. The `<msup>` elements express superscripts and the `<mrow>` is used for a horizontal sequence. The form `×` is a named entity that expands to the Unicode character "×" (`U+00D7`). Note that there is no meaning ascribed to this expression. For example, x^2 is merely a superscripted quantity. There is no implication that it is a power—it could equally well denote the second component of a contravariant vector, for example.

When writing, or generating, presentation MathML it is important to use sufficient markup used to capture the syntactic structure of the expression. In effect, all subexpressions should be explicitly grouped with `<mrow>`s. For example, to express $a = b + c$ one would use

```
<mrow>
   <mi>a</mi>
   <mo>=</mo>
   <mrow><mi>c</mi><mo>+</mo><mi>d</mi></mrow>
</mrow>
```

and not

```
<mrow><mi>a</mi> <mo>=</mo> <mi>b</mi> <mo>+</mo> <mi>c</mi></mrow>.
```

This way, line breaking and sub-expression selection can be handled correctly.

Content MathML

MathML provides facilities to describe the meaning of mathematical expressions. The subset designed for this purpose is known as "content" MathML. Content MathML has a set of built-in tags to express the concepts which occur in elementary mathematics, up to the level corresponding approximately to the last year of secondary school or first year of university. More advanced concepts are expressed using markup with external references.

Content MathML expressions consist typically of expressions with operators applied to arguments, and are thus reminiscent of Lisp S-expressions. Supposing the presentation example above were intended as a sum of two squares, it could be represented in content MathML as

```
<apply>
    <plus/>
    <apply> <power/> <ci>x</ci> <cn>2</cn> </apply>
    <apply> <power/> <ci>y</ci> <cn>2</cn> </apply>
</apply>
```

Here the `<ci>` and `<cn>` give content markup for identifiers and numbers respectively. The `<plus/>` and `<power/>` elements denote operators. Expressions are formed from these leaves by giving function application elements with `<apply>`.

A few examples illustrating content MathML are given in Figure 2.

Characters and Symbols

Mathematical expressions can make use of a broad range of special symbols. MathML uses the full Unicode character set [6] to provide the numerous symbols and technical characters required.

Sometimes it is not convenient to work directly with the full range of Unicode characters. This is the case when using tools which do not support Unicode or when one must embed MathML data in a non-Unicode format. For this purpose MathML provides more than 2000 named entities for special characters. These allow Unicode characters to be referred to by mnemonic names, *e.g.* `α` for α, and for any MathML expression to be written in ASCII. The MathML entities include alphabets used in mathematics (*e.g.* Greek, Cyrillic, Hebrew,

$$\frac{a \pm \sqrt{b}}{c}$$

```
<mfrac>
  <mrow><mi>a</mi><mo>&PlusMinus;</mo><msqrt><mi>b</mi></msqrt></mrow>
  <mi>c</mi>
</mfrac>
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
<mfenced open="[" close="]"> <mtable>
    <mtr><mtd><mi>a</mi></mtd> <mtd><mi>b</mi></mtd></mtr>
    <mtr><mtd><mi>c</mi></mtd> <mtd><mi>d</mi></mtd></mtr>
</mtable> </mfenced>
```

$$\sum_{i=1}^{n} e^{i} \omega_i$$

```
<mrow>
    <msubsup>
        <mo>&sum;</mo>
        <mrow> <mi>i</mi> <mo>=</mo> <mn>1</mn> </mrow>
        <mi>n</mi>
    </msubsup>
    <mrow>
        <msup><mi>e</mi><mi>i</mi></msup>
        <mo>&InvisibleTimes;</mo>
        <msub><mi>&omega;</mi> <mi>i</mi></msub>
    </mrow>
</mrow>
```

$$\lim_{h \to 0} \frac{f\left(t+h\right)}{h}$$

```
<mrow>
   <msub>
      <mo>lim</mo>
      <mrow><mi>h</mi> <mo>&rarr;</mo> <mn>0</mn></mrow>
   </msub>
   <mfrac>
      <mrow>
          <mi>f</mi>
          <mo>&ApplyFunction;</mo>
          <mfenced><mi>t</mi><mo>+</mo><mi>h</mi></mfenced>
      </mrow>
      <mi>h</mi>
   </mfrac>
</mrow>
```

Fig. 1. Presentation MathML examples

$$\sin^2 \theta$$

```
<apply>
   <power/>
   <apply> <sin/> <ci>&theta;</ci></apply>
   <cn>2</cn>
</apply>
```

$$\log^{-1} = \exp$$

```
<apply>
   <eq/>
   <apply><inverse/><log/></apply>
   <exp/>
</apply>
```

$$\int_a^b f(t)\,dt$$

```
<apply>
   <int/>
   <bvar><ci>t</ci></bvar>
   <lowlimit><ci>a</ci></lowlimit>
   <uplimit><ci>b</ci></uplimit>
   <apply> <ci>f</ci> <ci>t</ci> </apply>
</apply>
```

$$\forall x, x \in \mathbb{R}, x > 1 : x^2 > x$$

```
<apply>
   <forall/>
   <bvar><ci>x</ci></bvar>
   <condition>
      <apply>
         <and/>
         <apply><in/> <ci>x</ci> <reals/></apply>
         <apply><gt/> <ci>x</ci> <cn>1</cn></apply>
      </apply>
   </condition>
   <apply>
      <gt/>
      <apply><power/><ci>x</ci><cn>2</cn></apply>
      <ci>x</ci>
   </apply>
</apply>
```

Fig. 2. Content MathML examples

fraktur, script and open-face), as well as numerous brackets, operators and other symbols.

The Unicode character set is organized in a set of *planes* of 2^{16} points. The *Basic Multilinugal Plane* (BMP) consists of the characters with values from 0 to $2^{16}-1$. At the time of writing, many programs do not properly handle Unicode characters outside the BMP. While most characters for symbols used by MathML lie in the BMP, certain mathematical alphabets (*e.g.* open face, script and fraktur) lie in other planes. To give access to these alphabets in applications that handle only BMP values, the `mathvariant` attribute may be used. For example, one can give a script A identifier as `<mi mathvariant="script">A</mi>`. Unlike using a numeric or named entity (`<mi>𝥌</mi>` or `<mi>𝒜</mi>`), the `<mi>` with a `mathvariant` attribute is not automatically expanded to use `U+1D94C`.

Annotations

It is often the case that MathML objects can have additional associated information. For example, if a computer algebra system generates MathML output, it may be desired to associate the system's original expression with the MathML. The `<semantics>` element is used for this purpose. The first child of this element is an expression to be annotated, and the second and any subsequent children are annotations either in textual or XML form. Figure 3 gives an expression with annotations providing meanings in Maple, TEX and OpenMath [2][3][4].

```
<semantics>
   <mrow> <mi>x</mi> <mo>&times;</mo> <mi>y</mi> </mrow>

   <annotation encoding="Maple"> x * y </annotation>
   <annotation encoding="TeX"> x \times y </annotation>
   <annotation-xml encoding="OpenMath">
      <OMOBJ xmlns="http://www.openmath.org/OpenMath">
         <OMA>
            <OMS cd="arith1" name="times"/>
            <OMV name="x"/>
            <OMV name="y"/>
         </OMA>
      </OMOBJ>
   </annotation-xml>
</semantics>
```

Fig. 3. Example of annotations for alternative encodings

Combining Presentation and Content

It is not uncommon to work with both presentation and content for the same mathematical expression. This can be done with a `<semantics>` element, giving either the presentation or the content as the first child and the other as the annotation, as shown in Figure 4. Joining a presentation expression and content expression in this way gives what is known as *top-level parallel markup*.

```
<semantics>
    <mrow> <mi>a</mi> <mo>+</mo> <mi>b</mi> </mrow>
    <annotation-xml encoding="MathML-Content">
        <apply> <plus/> <ci>a</ci> <ci>b</ci> </apply>
    </annotation-xml>
</semantics>
```

Fig. 4. Example of top-level parallel markup

In many applications, it is desirable to be able to select subexpressions and to be able to find both their content and presentation markup. Top-level parallel markup is insufficient for this purpose. MathML provides `id` and `ref` attributes which may be used to cross-reference the subexpressions of content and presentation trees. This gives what is known a *fine-grained parallel markup*, as shown in Figure 5.

```
<semantics>
    <mrow id="G1">
        <mi id="G2">a</mi>
        <mo id="G3">+</mo>
        <mi id="G4">b</mi>
    </mrow>
    <annotation-xml encoding="MathML-Content">
        <apply xref="G1">
            <plus xref="G3"/>
            <ci xref="G2">a</ci>
            <ci xref="G4">b</ci>
        </apply>
    </annotation-xml>
</semantics>
```

Fig. 5. Example of fine-grained parallel markup

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
                 href="http://www.w3.org/Math/XSL/mathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>My Page</head>
  <body>
    <h1>Example</h1>
    <math xmlns="http://www.w3.org/1998/Math/MathML">
        <mrow> <mi>a</mi> <mo>+</mo> <mn>b</mn> </mrow>
    </math>
  </body>
</html>
```

Fig. 6. Use of W3C universal MathML style sheet

Current Status

MathML has been an official Recommendation of the World Wide Web Consortium (W3C) since March 1998 as Version 1.0, and since February 2001 as Version 2.0.

At the time of writing, MathML can be imported and exported from major computer algebra systems, is supported natively or via plug-ins in the most popular web browsers, and handled by certain editors. Of the computer algebra systems, MathML may be imported or exported by both Maple (version 8 and higher) and Mathematica (version 4 and higher). Maple places greater emphasis on Content MathML while Mathematica emphasizes presentation MathML. The browsers Amaya, Mozilla 1.0 and Netscape 7.0 PR1 support MathML natively (though Amaya supports only Presentation MathML). IBM's techExplorer and Design Science's MathPlayer can both be used to display MathML in InternetExplorer 5.5 and later. Used naïvely, some of the browsers/extension combinations require browser-specific markup (`<object>`, `<applet>`, or `<embed>` tags) to view pages containing MathML. It is however possible to write *browser-independent pages* by making use of the MathML universal style sheet [7] from the W3C Math Working Group. With this, an XHTML [5] page containing MathML would appear as shown in Figure 6. The two key items that must be included are the `<?xml-stylesheet...?>` processing instruction and the `xmlns` attribute on the `<math>` tag.

MathML is now supported by a wide range of software, and has found many applications outside of the original context of math for web pages. For example, it is used now in web-services for computer algebra systems and as the archival form for mathematics in all United States patents.

Stephen Watt (London Ontario)

References

1. R. Ausbrooks, S. Buswell, S. Dalmas, S. Devitt, A. Diaz, R. Hunter, B. Smith, N. Soiffer, R. Sutor, and S. Watt. *Mathematical Markup Language (MathML) Version 2.0*. W3C Recommendation 21-February-2001. World Wide Web Consortium, 2001. D. Carlisle, P. Ion, N. Poppelier and R. Miner, editors, `http://www.w3.org/TR/2001/REC-MathML2-20010221`.
2. Olga Caprotti, David P. Carlisle, and Arjeh M. Cohen, editors. *The OpenMath Standard Version 1.0*. The OpenMath Esprit Consortium, February 2000. `http://www.openmath.org/standard/omstd.pdf`.
3. S. Dalmas, M. Gaëtano, and S.M. Watt. An OpenMath v1.0 implementation. In W. W. Küchlin, editor, *ISSAC '97, International Symposium on Symbolic and Algebraic Computation, Maui, Hawaii*, pages 221–248, New York, 1997. ACM.
4. W. N. Naylor and S. M. Watt. On the relationship between OpenMath and MathML. In *Workshop on Internet Accessible Mathematical Communication*. ICM, 2001. `http://icm.mcs.kent.edu/research/iamc01/proceedings.html`.
5. Steve Pemberton, Murray Altheim, and et al. *XHTML[tm] 1.0: The Extensible HyperText Markup Language*. W3C Recommendation 26 January 2000. World Wide Web Consortium, 2000. `http://www.w3.org/TR/2000/ REC-xhtml1-20000126`.
6. The Unicode Consortium. *The Unicode Standard, Version 3.0*. Addison-Wesley, 2000. as ammended by Unicode Standard Annex 27: Unicode 3.1 `http://www.unicode.org/reports/tr27/` and by Unicode Standard Annex 28: Unicode 3.2 `http://www.unicode.org/reports/tr28`.
7. W3C Math Working Group. *Putting Mathematics on the Web with MathML*. World Wide Web Consortium, March 2002. `http://www.w3.org/Math/XSL/`.

This article was processed using the LaTeX macro package with LMAMULT style