MathML Presentation Markup for the Impatient

Hussein Shafie, XMLmind Software

September 9, 2019

Table of Contents

MathML comprises two sets of elements: *Presentation Markup*, the XML equivalent of TeX math, and *Content Markup*, which may be used to encode the mathematical structure of an expression, regardless of the way this expression is rendered visually. This short tutorial is exclusively about Presentation Markup. After reading it, you should be able to add equations to your DocBook, DITA or XHTML documents.

1. Basic elements

MathML most basic elements are `mrow`, `mi`, `mo` and `mn`. Example: $x + y = 2$ is encoded in MathML as:

```
<mrow>
  <mrow>
    <mi>x</mi>
    <mo>+</mo>
    <mi>y</mi>
  </mrow>
  <mo>=</mo>
  <mn>2</mn>
</mrow>
```

`mrow`

Use this element to group any number of subexpressions horizontally.

`mi`

Use this element to specify an identifier, that is, the name of a variable, a constant, a function, etc.

If this name is just one character long, the identifier is automatically rendered using an italic font, otherwise the name is rendered using a normal, upright, font.

`mo`

Use this element to specify an operator (e.g. '+'), a fence (e.g. '{') or a separator (e.g. ',').

The appropriate amount of space is added on the left and on the right of an `mo` depending on the textual contents of this element. Example: if in the above expression you replace `<mo>+</mo>` by `<mo>,</mo>`, this will suppress the space at the left of the `mo` element.

`mn`

Use this element to specify a numeric *literal*.

For example, PI should be specified as `<mi>PI</mi>` and not as `<mn>PI</mn>` while 3.14 should be specified as `<mn>3.14</mn>` and not as `<mi>3.14</mi>`.

It is really important to use `mi`, `mo` and `mn` appropriately because otherwise the MathML rendering engine will not be able to apply its built-in typographic rules.

2. More basic elements

2.1. The `math` top-level element

The above MathML expression cannot be inserted as is in a DocBook, DITA or XHTML document because the `mrow` element should be enclosed in a `math` element. The `math` element is the root of all MathML expressions.

```
<math xmlns="http://www.w3.org/1998/Math/MathML"❶
      display="inline">❷
  <mrow>
    <mrow>
      <mi>x</mi>
      <mo>+</mo>
      <mi>y</mi>
    </mrow>
    <mo>=</mo>xml
    <mn>2</mn>
  </mrow>
</math>
```

❶ The namespace of all MathML elements is "`http://www.w3.org/1998/Math/MathML`". Specifying such namespace is mandatory but it will be omitted in this tutorial for brevity.

❷ Note the `display="inline"` attribute which specifies that the `math` element is to be displayed inline (``like a word in a paragraph''). The other value is `display="block"` which specifies that the `math` element is to be displayed as a block (``like a paragraph''). This attribute has an influence on the typographic rules applied by the MathML rendering engine.

2.2. Plain text

We have already recommended to be very precise in the use of `mi`, `mo` and `mn` when tagging some text. But what if you just want to type plain text? Here enters the `mtext` element, which with `mi`, `mo`, `mn` and `ms` are the only MathML elements which may contain text. All the other MathML elements (`math`, `mrow`, `mfrac`, `msqrt`, etc) may only contain child elements.

Example:

if $x = y$ then $ax = ay$

is encoded in MathML as:

```
<mrow>
  <mtext>if</mtext>
  <mspace depth="0.5ex" height="0.5ex" width="1ex"/>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mi>y</mi>
  </mrow>
  <mspace depth="0.5ex" height="0.5ex" width="1ex"/>
  <mtext>then</mtext>
  <mspace depth="0.5ex" height="0.5ex" width="1ex"/>
  <mrow>
    <mrow>
      <mi>a</mi>
      <mi>x</mi>
    </mrow>
    <mo>=</mo>
    <mrow>
      <mi>a</mi>
      <mi>y</mi>
    </mrow>
  </mrow>
</mrow>
```

2.3. Explicit space between elements

If in the above example, you want to add some space after word "`if`", do not attempt to insert one or more whitespace characters in the corresponding `mtext` element (e.g. `<mtext>if </mtext>`). Doing so is useless because, leading and trailing whitespace characters are automatically removed from `mi`, `mo`, `mn`, and `mtext` by the MathML processor. Instead, you need to insert an `mspace` element in your MathML expression. Note that due to the built-in typographic rules, doing so is just occasionally needed.

`width`
: This optional attribute specifies the overall width of the `mspace` element.

`height`
: This optional attribute specifies the overall height above the baseline.

`depth`
: This optional attribute specifies the overall height below the baseline.

The value of these attributes is a number followed by one of the following units: `em`, `ex`, `px`, `in`, `cm`, `mm`, `pt`, `pc`.

3. Fractions

Fractions are specified using the `mfrac` element. Example: $\frac{x-1}{100}$

```
<mfrac>
  <mrow>
    <mi>x</mi>
    <mo>-</mo>
    <mn>1</mn>
  </mrow>
  <mn>100</mn>
</mfrac>
```

First child element is the numerator of the fraction. Second child element is its denominator.

Attribute `bevelled="true"` may be used to change the style of the fraction. Example: $x-1/100$.

4. Radicals

MathML has two elements allowing to specify radicals:

`msqrt`

Use this element to specify a square root. Example: $\sqrt{x+y}$

```
<msqrt>
  <mi>x</mi>
  <mo>+</mo>
  <mi>y</mi>
</msqrt>
```

Note that, like a number of other MathML elements (`mstyle`, `merror`, `menclose`, `mpadded`, `mphantom`, `mtd` and `math`), `msqrt` may have one or more child elements. Below the radical sign, `msqrt` behaves as if it had an implicit `mrow` element grouping all its child elements.

`mroot`

Use this element to specify a radical with an arbitrary index. Example: $\sqrt[3]{x}$

```
<mroot>
  <mi>x</mi>
  <mn>3</mn>
</mroot>
```

Unlike `msqrt`, `mroot` may only have two child elements. First child element is the base of the root. Second child element is its index. If you need more that one element below the radical sign, then use an explicit `mrow` element.

5. Subscripts and superscripts

Subscripts and superscripts elements are:

`msub`

Use this element to attach a subscript to a base. Example: x_i

```
<msub>
  <mi>x</mi>
```

```
  <mi>i</mi>
</msub>
```

`msup`

Use this element to attach a superscript to a base. Example: x^j

```
<msup>
  <mi>x</mi>
  <mi>j</mi>
</msup>
```

`msubsup`

Use this element to attach both a subscript and a superscript to a base. Example: x_i^j

```
<msubsup>
  <mi>x</mi>
  <mi>i</mi>
  <mi>j</mi>
</msubsup>
```

Note that for all the above elements, the base is the *first* child element.

6. Underscripts and overscripts

Underscripts and overscripts are similar to subscripts and superscripts, except that script elements are centered above and/or below the base element.

`munder`

Use this element to attach a underscript to a base. Example: $\underline{x}$

```
<munder>
  <mi>x</mi>
  <mo>&#9472;</mo>
</munder>
```

`mover`

Use this element to attach a overscript to a base. Example: $\vec{v}$

```
<mover>
  <mi>v</mi>
  <mo>&#8594;</mo>
</mover>
```

`munderover`

Use this element to attach both a underscript and a overscript to a base. Example: $\underset{a}{\overset{b}{x}}$

```
<munderover>
  <mi>x</mi>
  <mi>a</mi>
  <mi>b</mi>
</munderover>
```

7. The ubiquitous `mo` element

Even after all these explanations, it is probably still not obvious to figure out how to encode in MathML many common constructs such as integrals, limits, etc. The answer is simple: use an `mo` element containing the right character. This character typically belongs to the "Mathematical Operators" U+2200-U+22FF Unicode range or to the "Arrows" U+2190-U+21FF Unicode range.

Example 1:

$$\int_{-1}^{+1} \frac{dx}{x}$$

```
<mrow>
  <munderover>
    <mo>&#8747;</mo>
    <mn>-1</mn>
    <mn>+1</mn>
  </munderover>
  <mfrac>
    <mrow>
      <mi>d</mi>
      <mi>x</mi>
    </mrow>
    <mi>x</mi>
  </mfrac>
</mrow>
```

Example 2:

$$x \underset{\text{maps to}}{\longrightarrow} y$$

```
<mrow>
  <mi>x</mi>
  <munder>
    <mo>&#8594;</mo>
    <mtext>maps to</mtext>
  </munder>
  <mi>y</mi>
</mrow>
```

Note how the `mo` element stretches vertically or horizontally when needed to. The amount of stretching is normally automatically determined by the MathML renderer, but it is possible to influence it by specifying the `minsize` and/or `maxsize` attributes. Example: `<mo minsize="10">&#8594;</mo>` means: make the arrow at least 10 times its normal size.

8. Matrices

Matrices are specified using the `mtable` element which is similar to the simple — no `tbody` — XHTML table. An `mtable` table element contains `mtr` row elements and an `mtr` element contains `mtd` cell elements.

Example:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
<mrow>
  <mo>[</mo>
  <mtable>
    <mtr>
      <mtd>
        <mn>1</mn>
      </mtd>
      <mtd>
        <mn>0</mn>
      </mtd>
      <mtd>
        <mn>0</mn>
      </mtd>
    </mtr>
    <mtr>
      <mtd>
        <mn>0</mn>
      </mtd>
      <mtd>
        <mn>1</mn>
      </mtd>
      <mtd>
        <mn>0</mn>
      </mtd>
    </mtr>
    <mtr>
      <mtd>
        <mn>0</mn>
      </mtd>
      <mtd>
        <mn>0</mn>
      </mtd>
      <mtd>
        <mn>1</mn>
      </mtd>
    </mtr>
  </mtable>
  <mo>]</mo>
</mrow>
```

Note that by default, an `mtable` element has no borders at all. This is why you'll generally need to add an `mo` containing a fence character (e.g. '[', ']', '(', ')', '|') before and after the `mtable` when you specify a matrix or the determinant of a matrix.

9. Equations

The MathML `mtable` element is fairly generic. Use it whenever you need to layout elements in a rectangular grid. This feature is of course useful to specify matrices. It is also useful to specify a set of equations.

Example (how to properly align this set of equations is explained below):

$$\begin{cases} 2x + y = 8 \\ x + y = 6 \end{cases}$$

```
<mrow>
  <mo>{</mo>
  <mtable>
    <mtr>
      <mtd>
        <mrow>
          <mrow>
            <mrow>
              <mn>2</mn>
              <mo>&#8290;</mo>
              <mi>x</mi>
            </mrow>
            <mo>+</mo>
            <mi>y</mi>
          </mrow>
          <mo>=</mo>
          <mn>8</mn>
        </mrow>
      </mtd>
    </mtr>
    <mtr>
      <mtd>
        <mrow>
          <mrow>
            <mi>x</mi>
            <mo>+</mo>
            <mi>y</mi>
          </mrow>
          <mo>=</mo>
          <mn>6</mn>
        </mrow>
      </mtd>
    </mtr>
  </mtable>
</mrow>
```

Replacing an `mtr` row element by an `mlabeledtr` labeled row element allows to use the first `mtd` cell element of this row as the caption of the equation. Example:

Gauss' law $\nabla \cdot \mathrm{E} = \frac{\rho}{\varepsilon_0}$

Gauss's law for magnetism $\nabla \cdot \mathrm{B} = 0$

```
<mtable side="left">
  <mlabeledtr>
    <mtd>
      <mtext>Gauss' law</mtext>
    </mtd>
    <mtd>
```

```
      <mrow>
        <mrow>
          <mo>&#8711;</mo>
          <mo>&#8729;</mo>
          <mi mathvariant="normal">E</mi>
        </mrow>
        <mo>=</mo>
        <mfrac>
          <mi>&#961;</mi>
          <msub>
            <mi>&#949;</mi>
            <mn>0</mn>
          </msub>
        </mfrac>
      </mrow>
    </mtd>
  </mlabeledtr>
  <mlabeledtr>
    <mtd>
      <mtext>Gauss's law for magnetism</mtext>
    </mtd>
    <mtd>
      <mrow>
        <mrow>
          <mo>&#8711;</mo>
          <mo>&#8729;</mo>
          <mi mathvariant="normal">B</mi>
        </mrow>
        <mo>=</mo>
        <mn>0</mn>
      </mrow>
    </mtd>
  </mlabeledtr>
</mtable>
```

Note that without the `side="left"` attribute, captions are displayed at the right of equations and this, despite the fact that the caption is always specified by the contents of the first `mtd` child of a `mlabeledtr` element.

10. Other, less useful, elements

We'll not describe in this tutorial the following, rarely needed, elements: `mglyph`, `mmultiscripts`, `malignmark`, `merror`, `maction`. This being said, you may also skip this section if you are really impatient.

`ms`

Use this element to specify a quoted string literal. Example: "Hello word!"

```
<ms>Hello word!</ms>
```

`mfenced`

The `mfenced` element is a shorthand notation for common forms of `mrow`. Example: (x, y, z)

```
<mfenced>
```

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

is equivalent to: (x, y, z)

```
<mrow>
  <mo>(</mo>
  <mi>x</mi>
  <mo>,</mo>
  <mi>y</mi>
  <mo>,</mo>
  <mi>z</mi>
  <mo>)</mo>
</mrow>
```

The `open`, `separators` and `close` attributes of an `mfenced` element specify the opening fence added before its first child element, the separators added between child elements and the closing fence added after its last child element. By default, the values of these attributes are "`(`", "`,`" and "`)`".

`menclose`

The `menclose` element allows to draw lines, typically a box, around its child elements. Example: $n!$

```
<menclose notation="box">
  <mi>n</mi>
  <mo>!</mo>
</menclose>
```

The `notation` attribute of an `menclose` element specify which kind of lines are drawn around the child elements. The allowed values for this attribute are: `longdiv` (default value), `actuarial`, `radical`, `box`, `roundedbox`, `circle`, `left`, `right`, `top`, `bottom`, `updiagonalstrike`, `downdiagonalstrike`, `verticalstrike`, `horizontalstrike`.

`mpadded`

The `mpadded` element allows to add padding, that is extra space, around its child elements. It's an alternative to using `mspace`. Example:

if $x = y$ then $ax = ay$

```
<mrow>
  <mpadded width="+1ex">
    <mtext>if</mtext>
  </mpadded>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mi>y</mi>
  </mrow>
  <mpadded lspace="1ex" width="+2ex">
    <mtext>then</mtext>
  </mpadded>
  <mrow>
```

```
    <mrow>
      <mi>a</mi>
      <mi>x</mi>
    </mrow>
    <mo>=</mo>
    <mrow>
      <mi>a</mi>
      <mi>y</mi>
    </mrow>
  </mrow>
</mrow>
```

The attributes allowing to specify the padding are:

`width`
: This optional attribute specifies the overall width of the `mpadded` element.

  The value of this attribute, as well as the values of the `height` and `depth` attributes (but not the `lspace` attribute) described below, may start with a "+" sign which means: add specified amount to the intrinsic size.

`lspace`
: This optional attribute specifies the amount of space added before the first child of the `mpadded` element.

  There is no `rspace` attribute. The amount of space added after the last child of the `mpadded` element is: value of the above `width` attribute - intrinsic width of all the child elements - value of this `lspace` attribute.

`height`
: This optional attribute specifies the overall height above the baseline.

`depth`
: This optional attribute specifies the overall height below the baseline.

`mphantom`
: The `mphantom` element transforms its descendant elements into ‘‘phantoms": they are there, they occupy some space, but you cannot see them. The `mphantom` element is often the only way to properly align some elements. Example:

$$\frac{1}{x}+\frac{1}{x^{y^z}}$$

```
<mrow>
  <mfrac>
    <mn>1</mn>
    <msup>
      <mi>x</mi>
      <mphantom>
        <msup>
          <mi>y</mi>
          <mi>z</mi>
        </msup>
      </mphantom>
```

```
    </msup>
  </mfrac>
  <mo>+</mo>
  <mfrac>
    <mn>1</mn>
    <msup>
      <mi>x</mi>
      <msup>
        <mi>y</mi>
        <mi>z</mi>
      </msup>
    </msup>
  </mfrac>
</mrow>
```

`mstyle`

The `mstyle` element allows to specify attributes which are intended to be inherited by all its descendant elements. As such, the `mstyle` element supports *all* the attributes of *all* the other MathML elements.

The most commonly used attributes are those used to style the `mi`, `mo`, `mn` and `mtext` text container elements:

Attribute Name	Attribute Value	Default Value
`mathvariant`	`normal` | `bold` | `italic` | `bold-italic` | `double-struck` | `bold-fraktur` | `script` | `bold-script` | `fraktur` | `sans-serif` | `bold-sans-serif` | `sans-serif-italic` | `sans-serif-bold-italic` | `monospace`	`normal` (except on `mi`)
`mathsize`	`small` | `normal` | `big` | *`number v-unit`*	inherited
`mathcolor`	*`#rgb`* | *`#rrggbb`* | *`html-color-name`*	inherited
`mathback-ground`	*`#rgb`* | *`#rrggbb`* | *`html-color-name`*	inherited

Example: $x + y = 2$

```
<mstyle mathbackground="yellow" mathcolor="navy" mathsize="16pt"
        mathvariant="bold">
  <mrow>
    <mi>x</mi>
    <mo>+</mo>
    <mi>y</mi>
  </mrow>
  <mo>=</mo>
  <mn mathcolor="red">2</mn>
</mstyle>
```

`maligngroup`

Use this element to properly align a set of equations. Each inserted `maligngroup` specifies a ``sub-column'' within the column of an `mtable`. The `groupalign` attribute of the `mtable` element specifies the horizontal alignment within each ``sub-column''. Example:

$$\begin{cases} 2x + y = 8 \\ x + y = 6 \end{cases}$$

```
<mrow>
  <mo>{</mo>
  <mtable groupalign="{right center right center right}">
    <mtr>
      <mtd>
        <mrow>
          <mrow>
            <mrow>
              <maligngroup/>
              <mn>2</mn>
              <mo>&#8290;</mo>
              <mi>x</mi>
            </mrow>
            <maligngroup/>
            <mo>+</mo>
            <maligngroup/>
            <mi>y</mi>
          </mrow>
          <maligngroup/>
          <mo>=</mo>
          <maligngroup/>
          <mn>8</mn>
        </mrow>
      </mtd>
    </mtr>
    <mtr>
      <mtd>
        <mrow>
          <mrow>
            <maligngroup/>
            <mi>x</mi>
            <maligngroup/>
            <mo>+</mo>
            <maligngroup/>
            <mi>y</mi>
          </mrow>
          <maligngroup/>
          <mo>=</mo>
          <maligngroup/>
          <mn>6</mn>
        </mrow>
      </mtd>
    </mtr>
  </mtable>
</mrow>
```

The value of the `groupalign` attribute has the following syntax: one "`{...}`" group per column. A "`{...}`" group contains one alignment specification per sub-column (that is, per `maligngroup`). Alignment specifications are: `left`, `center` or `right`.