1 MathML 3 and implications on TeX4ht

MathML 3 is about to be formalized and going to be released as the new standard for encoding mathematics in web. It differs from the previous standard version 2. The main changes that have a bearing on the functionality of TEX4ht are discussed here.

2 Linking

MathML 1 and 2 didn't have any built in provisions to specify links for various elements. The recommendation was to use XLink since it was part of the original grand plan for XML in its formative years. However, XLink didn't gain much popularity nor was it supported widely. Therefore, MathML 3 now specifies that any MathML element can accept an href attribute that takes an URI specifying a hyperlink same as in the HTML element, namely, <a>. So this has an implication in the current scheme of TEX4ht. If readers provide a few examples of how they want to include links in a math fragment in LATEX, it will help largely to make a stable and suitable conversion scheme for TEX4ht.

3 Line breaking

Previous standards of MathML had the same limitations as TEX in the matter of breaking long equations. In TEX, it is easy for an author to break equations while authoring content taking into consideration of the margin constrains and paper sizes. However, for web content, margin and line-length are irrelevant when the window sizes are unknown or potentially liable to dynamic changes. Therefore, new MathML 3 specifies a line breaking model and introduces several new attributes to control properties of line breaks and alignments.

This has deeper implications on the rendering software than on the engine that creates MathML. However, since the specification allows newer properties for line breaks and alignments, we need to take care of those part in TEX4ht which at the moments obeys what the author has input in her LATEX document. Since TEX is incapable of determining the semantic meaning of equations, we have to provide hooks to make TEX4ht aware of the real line breaks and obligatory line breaks to adjust with page sizes. A better option will be to make use of breqn package while authoring and TEX4ht to provide necessary extra functionality to take care of breqn features in combination with MathML's newer goodies.

4 Image inclusion

The usage of <glyph> element in the previous MathML specification has been deprecated now. <glyph> element was used to specify characters from non-standard fonts that do not correspond to Unicode code points. Since the usage becomes difficult in a web context, as one needs to make sure that the font is available, <mglyph> has been extended with src attribute. <mglyph> now acts as a general image reference element in MathML much like HTML's <img> element.

5 Elementary math layout

Elementary school math layout is difficult to typeset even with TEX. Usually people adopt a tabular layout with a lot of complex macros to control space and rules around various number segments. Previous MathML standards provide elementary school math layout, much like in TEX, with a lot of tabular structure which obstructs rendering of math other than visual, particularly audio, because of the interference of tabular elements. MathML 3 provides better layout elements and scheme which help to specify even borrows and carries and many different kind of layout.

```
  435.3

   ______
  3)1306
    12

    __
     10
      9

     __
      16
      15

      __
       1.0
         9

         _
         1
```

5.1 MathML version of long division

Given below is the MathML 3 coding of the above long division.

```
<mlongdiv longdivstyle="lefttop">
  <mn> 3 </mn>
  <mn> 435.3</mn>
  <mn> 1306</mn>
  <msgroup position="2" shift="-1">
    <msgroup>
      <mn> 12</mn>
      <msline length="2"/>
    </msgroup>
    <msgroup>
      <mn> 10</mn>
      <mn> 9</mn>
      <msline length="2"/>
    </msgroup>
    <msgroup>
      <mn> 16</mn>
      <mn> 15</mn>
      <msline length="2"/>
      <mn> 1.0</mn>
    </msgroup>
    <msgroup position='-1'>
```

```
       <mn> 9</mn>
      <msline length="3"/>
      <mn> 1</mn>
    </msgroup>
  </msgroup>
</mlongdiv>
```

6 Directionality

MathML 3 provides support for right to left writing/reading direction whereas previous versions didn't have this functionality. The new specifications provide the user to control both writing directions of the identifiers within a formula and layout direction of the formula itself separately. I am not sure, if this has any influence on the behavior of current version of TEX4ht which is based on `dvi` files generated by TEX/LATEX while support for `xdv`, extended dvi generated by XeTEX, is still very basic which supports multi-directional scripts.