EuroTEX2001 Kerkrade

Usage of MathML for paper and web publishing

Tobias Burnus

The pre-MathML situation

TEX

This is the method used by the most (all) of those sitting here.

Good:

- Easy to type in
- Wide spread
- Optimal quality
- Can be easily shared as PDF

Bad:

- The TEX code is not very standardised due to different packages and LATEX2E vs. ConTEXt
- Can't be validated

Ugly:

- TEX to HTML output using Images creates big, unreadable and not postprocessable files

Word processors and DTP programs

Good:

- WYSIWG (well, nearly)
- If you like clicking it is easy to create (or you have to learn yet another math language as in OpenOffice)

Bad:

- You cannot pre- or postprocess it due to changing, binary and proprrietorial format

Ugly:

- Frequently sub-optimal quality
- Web output has the same problems as TEX (if there exists an converter)

The brave new world

Promises

The World Wide Web Consortium says that those goals are met by MathML (excerpt):

- Encode mathematical material suitable for teaching and scientific communication at all levels.
- Encode both mathematical notation and mathematical meaning.
- Facilitate conversion to and from other mathematical formats, both presentational and semantic.
- Be well suited to template and other mathematics editing techniques.
- Be human legible, and simple for software to generate and process.

But how does a language defined in such a way look like ...

MathML tiny sample

Let's show how a simple formula in written in MathML. *We* TEXies write a mathematical formula as:

```
\int_a^b f(x)\,{\rm d}x = F(b)-F(a)
```

The mathematicians write it in these strange hieroglyphs:

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

And in MathML ...

The ...
The brave ...
Why ...
Usage
Conclusions
Close

Presentational MathML

In presentation markup of MathMLit looks like this:

```
<math>
  <mrow>
    <mrow>
       <msubsup><mo>&int;</mo><mi>a</mi><mi>b</mi></msubsup>
       <mrow><mi>f</mi><mo>&ApplyFunction;</mo>
          <mo>(</mo><mi>x</mi><mo>)</mo>
       </mrow>  <mo> &InvisibleTimes; </mo>
       <mrow>
         <mo>d</mo>
         <mi>x</mi>
       </mrow>
    </mrow>
    <mo>=</mo>
    <mrow>
      <mi>F</mi><mo>&ApplyFunction;</mo>
        <mrow><mo>(</mo><mi>b</mi><mo>)</mo></mrow>
      <mo>-</mo>
      <mi>F</mi><mo>&ApplyFunction;</mo>
        <mrow><mo>(</mo><mi>a</mi><mo>)</mo></mrow>
```

```
    </mrow>
  </mrow>
</math>
```

$$\int_a^b f(x)\mathrm{d}x = F(b) - F(a)$$

In a more structured way ...

Content MathML

More beautiful the content markup:

```
<math>
  <apply><eq/>
    <apply><int/>
      <bvar><ci>x</ci></bvar>
      <lowlimit><ci>a</ci></lowlimit>
      <uplimit><ci>b</ci></uplimit>
      <apply><fn><ci>f</ci></fn><ci>x</ci></apply>
    </apply>

    <apply><minus/>
      <apply><fn><ci>F</ci></fn><ci>b</ci></apply>
      <apply><fn><ci>F</ci></fn><ci>a</ci></apply>
    </apply>
  </apply>
</math>
```

$$\int_a^b f(x)\,\mathrm{d}x = F(b) - F(a)$$

Why content markup is that cool

Close

Directives to change the layout

The nice part of the encoding of meaning is that one can change the layout easily. Let's take this long fraction:

```
<math>
<apply><approx/>
   <apply><sin/><ci>x</ci></apply>
   <apply><divide/>
       <ci>x</ci>
       <apply><divide/><cn>1</cn>
           <apply><divide/><cn>1</cn><cn>1</cn></apply>
       </apply>
   </apply>
</apply>
</math>
```

Which looks as reasonable default:

$$\sin x \approx \frac{x}{\frac{1}{\frac{1}{1}}}$$

You can also display it as:

$$\sin\left(x\right) \approx x/1/1/1$$

This is done by

```
<?context-mathml-directive divide level 0?>
<?context-mathml-directive function reduction no?>
```

Usage

Why is presentational MathML used?

Almost all programs which can write MathML files use presentational MathML.

This brings us to the question who produces MathML:

- Mathematica. I tried 4.0 and it produces a wild HTML+MathML mixture (4.01 should be better)
- Maple 6 / Maple 7 (untried, Maple V R5.5 doesn't)
- OpenOffice/StarOffice 6: Not very well but you guess that it will work in the final release
- Ωmega. This is the natural way to produce MathML files which then will be processed by TEX

The . . .
The brave . . .
Why . . .
Usage
Conclusions
Close

Other MathML renderers: Mozilla

From the list on the MathML page at the World Wide Web Consortium you can see that there are not many programs which can render MathML, the number of those creating it is much higher.

MathML impressions:

- For HTML+MathML you need a special header (DOCTYPE) which is not standard conform and crashes some other MathML renderers
- It is not enabled by default in Mozilla
- Is has font problems: You need to have certain fonts installed
- It has still some problems with Content MathML

Conclusions

The Good

- It is becoming a standard and TEX can use it
- Content markup allows you to setup the rendering in a consistent way
- Presentational markup is very easy to create by a software
- You can include literal TEX using annotations

The Bad

- The documentation: It is not very clear and has some bugs in it
- The presentation markup is frequently rather complicated and long
- You need (and can ⇒ good) intermix the content and presentation module
- Viewers and editors are not widely spread

The Ugly

- The test cases on W3C are not only frequently contradictionual to the specification but also to other test cases in the same group – That is not only ugly but also really BAD.