Why (and how) I am using XML and MathML

Richard Kaye, School of Mathematics, University of Birmingham

2006-01-17

This article is a follow-up to that by Peter Rowlett (MSOR Connections Nov 2005, vol 5 no 4, 25–6). I give some personal reasons to add to Rowlett's comments on why mathematicians should be using XML and MathML much more, and outline my own attempts to make XML/MathML publishing accessible to the mathematics community.

1 Why

Mathematicians have been served well by TeX and LaTeX for their mathematical typesetting. Too well, perhaps. At least, if an dedicated TeXnician of the last ten years has a chance to \relax and look about himself he will see that the rest of the world has moved on in several incompatible ways to the cosy world of TeX.

We probably were drawn to TeX in the first place because for the first time it provided all the mathematical characters we needed in a series of 7-bit fonts. But nowadays there seems to be a consensus on how characters should be encoded in computer documents. The unicode standard is out there and increasingly being used to provide standardised encodings of characters beyond the basic ASCII character set, and sadly unicode is incompatible with TeX's 7-bit kludge. Search engines, like Google, do remarkably well in indexing PDF and PS files, but they would do even better on an en-unicoded (or is that uni-encoded?) HTML web page, especially if it involves special mathematical characters or accented characters.

And then there is future-proofing. XML is just a data format for whatever data you happen to have, but comes with many standard tools for its display and transformation into other formats, such as CSS and XSLT. CSS enables web browsers to display XML, and XSLT is a sort of 'macro processor' for transformation. The other good news is that XML has a system of 'namespaces' that ensures that mark-up (roughly equivalent to TeX's 'macros') from different sources with the same name do not clash with each other, a problem I for one have encountered with TeX as I loaded up yet another macro-package only to find it breaks an existing one because of a name-conflict. So even if MathML turns out *not* to be the flavour of, say, the 2030s, then there will be ways to convert our documents into the new format and preserve the intended meaning. And it is 'intended meaning' that prevents any good way of converting (La)TeX to XHTML: (La)TeX is simply not rich enough to present the meaning of subexpressions in a mathematical formula to allow translation to MathML, let alone to a computer algebra system or other such program.

I could go on. For example, I believe strongly that, in the academic community, all of us have a duty to make our documents as widely accessible as possible to all in the world, irrespective of language or disability. Paper-based PDF documents are not the way to go here, though they may still be the final medium chosen for printing the documents out for the majority of us. (I recently had an email from a prospective student who is blind. He wanted to know how many lecturers here were using MathML, as he had software that could read such documents out loud. Failing that, he was just able to 'read' TeX source files, but PDF documents were quite impossible.)

Academics must stick to standards where they exist, to enable global searches, automatic translations, or other automatic transformations to aural or other formats wherever possible. The nature of our subject is not to pre-judge the future. And if, by sensible use of technology, our documents reach a wider readership, then that is good for us too.

I first heard about XML and MathML over 10 years ago. One puzzle is, given that it is so much superior to TeX, why are so few people in the mathematics community using it? One reason is that only recently have 'mainstream' browsers such as Firefox (`http://www.mozilla.com/firefox/`) been able to display MathML. And why has that taken so long? Well, in part it has to do with Microsoft's battles with Netscape over the browser market but mostly it is because we, the mathematical community have not seen the need for it. So to help promote MathML, as well as for the other reasons given above, I have started to write my lecture notes for the new module I will be teaching in the new year in XHTML+MathML, and direct my students to those places on the internet they can find a suitable browser and the fonts required, rather than providing PDF 'translations' for them.

You can see the results of this work at `http://web.mat.bham.ac.uk/R.W.Kaye/seqser/` for yourself. This address gives the home page for my module on Introductory Real Analysis (Sequences and Series). The home page itself hasn't got any mathematics itself, but towards the bottom it links to exercise sheets and lecture notes towards. As I write, there are five such pages linked in. By the time you read this there should be many more.

If you do decide you want to look at these pages and you haven't looked at MathML before with a web browser, you will need to ensure you have the correct software on your computer. In my view, the best browser for mathematics is Mozilla Firefox (`http://www.mozilla.com/firefox/`). The main reason is that it can display web pages with mathematics directly without having to make last-minute behind-the-scenes translations. Unfortunately, mathematics will not display properly unless additional fonts are installed, and the web page at `http://www.mozilla.org/projects/mathml/fonts/` details what is needed here. (A future version of firefox with all the required fonts included is promised some time in 2006.) On my MS-Windows machine, font installation went smoothly. It was a little more awkward on my Linux machines. For people using MS Internet Explorer, mathematics support has improved considerably since I last looked at it a year and a half ago with MathPlayer 2.0 (`http://www.dessci.com/en/products/mathplayer/`). My main gripes with this set up are: (1) the way MathPlayer has to make a transformation of the source document before it is displayed (so 'view > page source'*doesn't* show the source but in fact shows an intermediate); and (2) the fact that Internet Explorer does not seem to be fully XHTML-compliant yet, in particular cannot display the XML-standard combination '&apos;'.

2 How

The main disadvantage with XML and MathML in particular is how verbose it is. It was never designed for direct entry from a text editor, in the way LaTeX is commonly typed, or HTML can be. It seems that the W3C (an independent consortium who publish the web standards, including XML and MathML) never expected XML or MathML to be typed in directly. Instead, they rather expect XML authors to use specialised XML editors with drop-down menus using the mouse presenting a 'palette' of options available in that context, rather similar to the equation editor in word. There are such editors available, and one or two free ones that work across several platforms, including the equation editor in Open Office (which can export MathML) and W3C's own Amaya (`http://www.w3.org/Amaya/`) browser/editor (which can author XHTML with embedded MathML directly).

I for one find mouse-based editing tedious in the extreme as it can be very slow to use and rather limiting in that only the combinations available in the palettes can be used. In principle, it would be possible to input a special vocabulary of TeX and allow TeX itself to convert this text to XML, and this approach has certainly been advocated. I decided to experiment with a more flexible approach and wrote a Java program called *gloss* to convert a text file into XML: the conversion process itself is controlled by an XML file called a 'modular vocabulary' ('MV') and by writing different MVs it is in principle possible to convert other types text files to XML. The main application at present is authoring XHTML and MathML by writing plain text in a text editor and converting to XML with *gloss*. *Gloss* is still in early days—any information that is available can be accessed via my home page at `http://web.mat.bham.ac.uk/R.W.Kaye/`. The subject of text-input with *gloss* is too large and still too experimental for this article, but my experience is that it really can provide a format in which mathematical text can be typed in a text editor as quickly as LaTeX can be, and the source file is at least as legible as LaTeX is. The processing stage is slightly slower than LaTeX, but still only a matter of a couple of seconds or so for a typical document. This document has been typed using emacs and converted using *gloss*, and will be made available on the web via my Home Page.

As I have mentioned, the notes I have been writing have been on Sequences and Series for First Years. So I have had to be able to write statements like

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \cdots < \frac{1}{1} + \frac{1}{2} + \frac{1}{6} + \frac{1}{12} + \cdots = 1 + \sum_{n=2}^{\infty} \frac{1}{n\,(n-1)}.$$

In MathML this is stored as

```
<m:math xmlns:m="http://www.w3.org/1998/Math/MathML" display="block">
<m:munderover>
  <m:mo>&#x02211;</m:mo>
  <m:mrow><m:mi>n</m:mi><m:mo>=</m:mo><m:mn>1</m:mn></m:mrow>
  <m:mi>&#x0221E;</m:mi>
</m:munderover>
<m:mfrac>
  <m:mn>1</m:mn>
  <m:mrow><m:msup><m:mi>n</m:mi><m:mn>2</m:mn></m:msup></m:mrow>
</m:mfrac>
<m:mo>=</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>1</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>4</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>9</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>16</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mi>&#x02026;</m:mi>
<m:mo>&lt;</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>1</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>2</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>6</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mfrac><m:mn>1</m:mn><m:mn>12</m:mn></m:mfrac>
<m:mo>+</m:mo>
<m:mi>&#x02026;</m:mi>
<m:mo>=</m:mo>
<m:mn>1</m:mn>
<m:mo>+</m:mo>
<m:munderover>
  <m:mo>&#x02211;</m:mo>
  <m:mrow><m:mi>n</m:mi><m:mo>=</m:mo><m:mn>2</m:mn></m:mrow>
  <m:mi>&#x0221E;</m:mi>
</m:munderover>
<m:mfrac>
  <m:mn>1</m:mn>
  <m:mrow>
    <m:mi>n</m:mi>
    <m:mo>(</m:mo>
    <m:mi>n</m:mi><m:mo>-</m:mo><m:mn>1</m:mn>
    <m:mo>)</m:mo>
  </m:mrow>
</m:mfrac>.
</m:math>
```

whereas in the *gloss* system I have been using this is entered as

```
math
  series {mrow n = 1} infin {mfrac 1 {mrow n sup 2}}
  = {mfrac 1 1} + {mfrac 1 4} + {mfrac 1 9} + {mfrac 1 16} + hellip
  < {mfrac 1 1} + {mfrac 1 2} + {mfrac 1 6} + {mfrac 1 12} + hellip
  = 1 + series {mrow n = 2} infin {mfrac 1 {mrow n (n - 1)}}
```

You can probably make reasonable guesses as to how the first is a translation of the second. Suffice it to say that the typing can be done in a reasonable amount of time.

One of the joys of being able to write student's lecture notes as web pages is the extra facility that hyperlinks provide. For example, in the very first lecture I discussed how difficult it is to tell from numerical experiments whether $\sum_{n=1}^{\infty} \frac{1}{n^2}$ and $\sum_{n=1}^{\infty} \frac{1}{n}$ converge. In a web page I could just link the computer program and its output as hyperlinks for only those students who are curious enough to see them, thus having just the essential material on the main page. Similarly, rather than just quoting 'Theorem 4.23' and expecting the students to find it in their notes I can use a hyperlink.

So far, all my MathML experiments have been with presentation-MathML, which focuses on the presentational aspects of mathematics. There is a parallel form of MathML, Content-MathML that focuses on meanings rather than presentation. In time I do want to look at such semantic aspects. However, I am rather taken with OpenMath (`http://www.openmath.org/`), a rather elegant and much more flexible XML mathematics format that concentrates on semantic aspects only and which can be used in conjunction with presentation-MathML, and expect to be looking into this rather more in the future. However, whatever the future of my own experiments with *gloss*, MathML and OpenMath, the documents I will be writing in the near future will conform to existing standards and can be viewed, transformed, saved and edited in many readily-available editors.

3 Afterword: publication details

The above article appeared in MSOR Connections Vol 6 No 1 Feb 2006, 20–22. The only changes I have made are to update some of the web links from my experimental server at mat140.bham.ac.uk to the main School of Mathematics server at web.mat.bham.ac.uk. The article was written using an early version of my 'gloss' system and the source code, as well as translations into other formats, and any other information on the ariticle, should be available in this directory.

Richard Kaye
School of Mathematics
University of Birmingham
`http://web.mat.bham.ac.uk/R.W.Kaye/`