Digital Library of Mathematical Functions: \LaTeX, MathML and … OpenMath?

Bruce R. Miller
NIST
Needing no introduction...
Old, but still relevant

Citations of AMS55 relative to All Scientific.

AMS55 is apparently used more than ever.
Time for a Rewrite

- New functions;
- New properties of old functions;
- New applications.
- …and many opportunities.
  - The Internet;
  - Computer Algebra, Theorem Proving systems;
  - The Semantic Web.
DLMF Project

- Started looking at feasibility in 1997.
- NSF funding for authorship in 1999.
- 4 editors, ≈ 12 associate editors, ≈ 40 authors.
- Goals:
  - New mathematical content updating AMS55,
  - in form of Digital Library,
  - and in print form,
  - by 2005.
Choices: \LaTeX{}, XML, MathML, OpenMath

- \LaTeX{} is obviously good choice for document source.
- …and obviously bad.
- Target: XML, MathML, and (eventually) OpenMath.

I don’t need to tell you why…
Overview of talk

- \texttt{LATEXML} tool.
- Metadata: markup, annotations and connections,
- Data model of the Library
- Math: Parsing, synthesizing meaning.
\textbf{\LaTeX}\textsc{xml}: Goals

- $\LaTeX \Rightarrow \text{XML}$ Transformer
  - General purpose.
  - $\LaTeX$-like DTD (or other?)
  - Math to MathML, OpenMath
- Closely mimic $\TeX$ behaviour (& Quirks).
- Lossless.
- Extensible, Adaptable.
- Encourage higher-level markup, declarations.
- …and finish DLMF project!
To make more feasible adopt

- Modestly Content-oriented \LaTeX.
- Discourage Presentation Markup but don’t forbid.
- Encourage Content Markup, but keep typeable.
- Use document-specific information (internal/external) to resolve ambiguities.
Metadata: Making Connections

- Traditional \LaTeX: \ref, \cite, \index.
- Leverage our mathematics markup.
- Additional markup:
  - Annotations \note.
  - Special metadata: Original handbook reference.
  - Additional declarations.
Metadata: Using Connections

- Postprocessing XML documents.
- Disassemble XML into ‘database’.
- Note all connections.

Not really that hard.
DLMF Data Model

- Simple model (maybe too simple)
  - ID ⇒ Object(XML)
    (Chapter, Section, Table, Equation, ...)
  - linkages embedded within each object
    (insertion, reference, ...)

- Can (re)construct as necessary
  - Sectional units,
  - Search ‘hit-lists’

- Developing an ‘Indexing’ API by which
  search, refnum lookup, ...
  ⇒ ID’s
TEX source $\xrightarrow{\text{\LaTeX XML}}$ XML

- Let $\text{\LaTeX XML}$ deal with TEX quirks.
- Acts as structure-preserving Lexer.
  - Possibly augmented (math) Tokens:
    - Name,
    - Unicode, Font, ...
    - PartOfSpeech (ID, Function, Operator, ...
    - Type (eventually).
  - preserve any given structure (eg. \frac, ...)
Math: The Easy Stuff

\[ a = b + c \]

**\LaTeX XML** produces the tokens

\[
\begin{align*}
  &\text{<XMTok} a \text{</XMTok>} \\
  &\text{<XMTok} = \text{</XMTok>} \\
  &\text{<XMTok} b \text{</XMTok>} \\
  &\text{<XMTok} + \text{</XMTok>} \\
  &\text{<XMTok} c \text{</XMTok>}
\end{align*}
\]
Grammar-based parser.

Undeclared tokens get PartOfSpeech from

- Document-specific dictionary
  (possibly sectionally scoped)
- Default dictionary

Resulting Expression tree

- inspired by OpenMath.
- $\approx$ Content MathML;
  (although we haven’t done this yet).
- Easily converted to Presentation MathML.
Math: The Easy Stuff continued

\[ a = b + c \]

\LaTeX\_XMLpost parses this into

\[
<\text{XMApp}><\text{XMTok}>=</\text{XMTok}>

<\text{XMTok}>a</\text{XMTok}>

<\text{XMApp}><\text{XMTok}>+</\text{XMTok}>

<\text{XMTok}>b</\text{XMTok}>

<\text{XMTok}>c</\text{XMTok}>

</\text{XMApp}>

</\text{XMApp}>

10 Years of OpenMath,
Math: The Easy Stuff continued

\[a = b + c\]

Conversion to MathML yields

```xml
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>a</mi>
    <mo>=</mo>
    <mrow>
      <mi>b</mi>
      <mo>+</mo>
      <mi>c</mi>
    </mrow>
  </mrow>
</math>
```
Extension of Dictionary to support some Type system.

Type Analysis to further resolve ‘meaning’

→ OpenMath.

Any advice?
Reduce ambiguities by introducing higher-level markup:

\[ \text{deriv}[n]{f}{x} \Rightarrow \frac{d^n f(x + y)}{dx^n} \]

\LaTeX \text{ code:}

\textit{omitted}

\LaTeXXML \text{ declaration:}

\begin{verbatim}
DefConstructor('deriv[]{}{}',
    "<XMApp !#2(POS='BIGOP')>
    . "<XMTok name='deriv'/>
    . "?#2(<XMAvg>#2</XMAvg>)!#2(<XMTok name='Empty'/>)
    . "<XMAvg>#3</XMAvg>
    . "?#1(<XMAvg>#1</XMAvg>)</XMApp>";
\end{verbatim}
\textsc{LaTeXML} constructs the tree:

\begin{verbatim}
<XMApp><XMTok name='deriv'/>
   <XMAArg><XMTok>f</XMTok>
      <XMTok>(</XMTok><XMTok>x</XMTok><XMTok>+</XMTok><XMTok>y</XMTok><XMTok>)</XMTok>
      <XMTok>x</XMTok><XMTok></XMTok>
      <XMTok>+</XMTok><XMTok>y</XMTok><XMTok></XMTok>
   </XMAArg>
</XMApp>
\end{verbatim}

Parser can treat args individually,

\ldots avoiding much guesswork.
Math: Special Functions

With appropriate TeX macrology:

\[ \HyperpFq{p}{q} \Rightarrow p F_q \]

Introduce notion of evaluating a function at:

\[ \HyperpFq{p}{q}@{a}{b}{z} \Rightarrow p F_q (a; b; z) \]

or (alternative notation)

\[ \HyperpFq{p}{q}@@{a}{b}{z} \Rightarrow p F_q \left( \begin{array}{c} a \\ b \end{array}; z \right) \]

Palatable notation? Easier to type than

\[ \sideset{-p}{-q}{\mathop{F}}\left(\{a \ \atop b\};z\right) \]
With the end result:

\[
\begin{align*}
\langle \text{XMApp} \\ \\
\langle \text{XMTok name='HyperpFq'>F</XMTok> \\
\langle \text{XMTok}>p</XMTok> \\
\langle \text{XMTok}>q</XMTok> \\
\langle \text{XMTok}>a</XMTok> \\
\langle \text{XMTok}>b</XMTok> \\
\langle \text{XMTok}>z</XMTok> \\
\rangle \text{XMApp}
\end{align*}
\]

and we know which ‘\( F \)’ is intended.
Math: Issues

- Role of text and spacing in math.
- Overloading of *symbols* (scoping?)
  - $f$ is a function here, but a variable there.
- Palatable content math markup for \LaTeX.
- For *really* meaningful math (e.g., OpenMath)
  - need type analysis
  - need more info from authors
- Open ended...
Trends? (Or Wishes)

- Continued development and support for MathML
- Ditto for OpenMath
- Convergence of Markup styles and DocTypes for
  - Various \LaTeX\Rightarrow XML converters
  - Richer \LaTeX content markup in general (\LaTeX3?)
  - Project Authors able use different tools \LaTeX, CAS, Thm.Provers, Word Processors.