

Why \TeX and \LaTeX ML?

Bruce R. Miller

NIST

Why T_EX?

Used to be an easy question!

- Top-quality typesetting
— including mathematics;
- non-proprietary;
- flexible;
- yields to a variety of automation tasks.
- and *Everybody* uses T_EX!

But change is underfoot... underway, afoot, whatever

- Word processors improving.
- Better formatting.
- Even integrate math.

So then, *Why* T_EX?

Maybe you're a

- traditionalist,
- or a control freak,
- or maybe your colleagues (still) use it,
- or maybe you just like doing things that way!

In other words...

- I dunno, I like it.
- You'll have to decide for yourself.

Why T_EX in DLMF, eg.?

- Biggest bottleneck:
getting knowledge out of ~ 40 mathematician's heads
- ie. Authoring, Editing, Validating.
- Most are fluent in T_EX.
- Most expect it to look like T_EX.
- Only a few familiar with each of other options.

Well then, Why L^AT_EXML?

- Because you decided to use T_EX, of course,
- and you want to put it on the web,
- or do other cool electronical things.

What is \LaTeX XML?

- \LaTeX (or \TeX) \Rightarrow XML Transformer;
- Closely mimic \TeX behaviour (& Quirks);
- \LaTeX -like Document model
something you can *do things* with;
- \sim Lossless, Extensible, Adaptable;
- Encourage higher-level markup, declarations;
- Math to Presentation-MathML; Content & OpenMath(someday)
- (... and finish DLMF project!)

How do you use it?

Simplest case, reasonably clean \LaTeX \rightarrow XHTML.

- `latexml -dest=mydoc.xml mydoc`
- `latexmlpost -dest=mydoc.xhtml mydoc`

First task: Conversion

- `latexml -dest=mydoc.xml mydoc`
- Deals with \TeX and its complications;
- Expands macros, interprets primitives;
- but constructs XML instead of DVI.
- Finally, does some basic rewriting and parses the mathematics.

...example

Next task: Postprocessing

- Now you've got XML that nobody recognizes,
- but you can play with it.

Tasks:

- crossreferencing, indices, bibliography;
- conversion of math (MathML, images, ...);
- conversion of graphics;
- scanning to a database;
- XSLT to HTML, XHTML,

...do it.

Document ‘management’

- 1 $\text{T}_{\text{E}}\text{X}$ \rightarrow 1 HTML
- 1 $\text{T}_{\text{E}}\text{X}$ \rightarrow many HTML
- many $\text{T}_{\text{E}}\text{X}$ \rightarrow many HTML

...do that.

Customization

- Simple: customize the CSS;
- Less simple: customize the XSLT.

Hard Core Customization

For full enjoyment, Customize the conversion:

- redefine the handling of control sequences.
- Adopt a more semantic \LaTeX markup;
 - Embedded declarations;
 - high-level macros.

L^AT_EX_ML Math Processing; Lexing

Acts as structure-preserving Lexer.

- Math ‘Tokens’, possibly augmented:
 - Name,
 - Content: Unicode, Font, ...
 - Role (Grammatical): ID, Function, Operator, ...
 - Meaning (if known).
- Structures:
 - preserve any given structure (eg. `\frac, ...`)

L^AT_EX_ML Math Processing; Parsing

- Grammar-based parser,
- uses Role as the indicator.
- Undeclared tokens get grammatical role from
 - Document-specific dictionary (possibly sectionally scoped)
 - Default dictionary
- Resulting Expression tree:
 - inspired by OpenMath;
 - Dual/Parallel markup for presentation and content;
 - \approx Content MathML; (although we haven't done this yet).
 - Easily converted to Presentation MathML.

Simple example

$$F x = \frac{a}{1+x}$$

L^AT_EX_ML produces the tokens

```
<XMath>
```

```
  <XMApp>
```

```
    <XMTok meaning="eq" role="RELOP">=</XMTok>
```

```
    <XMApp>
```

```
      <XMTok role="MULOP">&InvisibleTimes;</XMTok>
```

```
      <XMTok role="UNKNOWN" font="italic">F</XMTok>
```

```
      <XMTok role="UNKNOWN" font="italic">x</XMTok>
```

```
    </XMApp>
```

```
    <XMApp>
```

```
      <XMTok meaning="div" role="MULOP" style="display"/>
```

```
      <XMTok role="UNKNOWN" font="italic">a</XMTok>
```

```
      <XMApp>
```

```
        <XMTok meaning="plus" role="ADDOP">+</XMTok>
```

```
        <XMTok meaning="1" role="NUMBER">1</XMTok>
```

```
        <XMTok role="UNKNOWN" font="italic">x</XMTok>
```

```
      </XMApp>
```

```
    </XMApp>
```

```
  </XMath>
```

Math: The Easy Stuff *continued*

Add

```
\usepackage{latexml}  
\lxDeclare[role=OPFUNCTION] {$F$}  
  
<XMApp>  
  <XMTok role="OPFUNCTION" font="italic">F</XMTok>  
  <XMTok role="UNKNOWN" font="italic">x</XMTok>  
</XMApp>
```

Math: Higher Level Markup

Define

```
\newcommand{\floor}[1]{\left\lfloor#1\right\rfloor}
```

and

```
DefMath('\floor{','\left\lfloor#1\right\rfloor', meaning=>'floor');
```

Now, $\$ \backslash \text{floor}\{x\} \$ \Rightarrow \lfloor x \rfloor$ produces

```
<XMath>
```

```
  <XMDual>
```

```
    <XMApp>
```

```
      <XMTok meaning="floor" role="UNKNOWN"/>
```

```
      <XMTok role="UNKNOWN" id="XX.S0.XM1" font="italic">x</XMTok>
```

```
    </XMApp>
```

```
    <XMApp>
```

```
      <XMTok name="floor" role="FENCED" argclose="[" argopen="]"/>
```

```
      <XMRef idref="XX.S0.XM1"/>
```

```
    </XMApp>
```

```
  </XMDual>
```

```
</XMath>
```

Example from DLMF: Special Functions

With appropriate T_EX macrology:

$$\backslash\text{HyperpFq}\{p\}\{q\} \Rightarrow {}_pF_q$$

Introduce notion of *evaluating a function at*:

$$\backslash\text{HyperpFq}\{p\}\{q\}@{\{a\}\{b\}\{z\}} \Rightarrow {}_pF_q(a; b; z)$$

or (alternative notation)

$$\backslash\text{HyperpFq}\{p\}\{q\}@@{\{a\}\{b\}\{z\}} \Rightarrow {}_pF_q\left(\begin{matrix} a \\ b \end{matrix}; z\right)$$

Palatable notation? Easier to type than

$$\backslash\text{sideset}\{_{\{p\}}\}_{\{q\}}\{\mathop{\text{F}}\}\backslash\text{left}(\{a \ \text{atop} \ b\}; z\backslash\text{right})$$

Math: Special Functions *continued*

With the end result:

```
<XMath>
  <XMDual role="ATOM">
    <XMApp>
      <XMApp role="FUNCTION">
        <XMTok meaning="HyperpFq" omcd="dlmf" role="FUNCTION"/>
        <XMTok role="UNKNOWN" id="CTEST.S1.XM21" font="italic">
        <XMTok role="UNKNOWN" id="CTEST.S1.XM22" font="italic">
      </XMApp>
      <XMTok role="UNKNOWN" id="CTEST.S1.XM23" font="italic">a<
      <XMTok role="UNKNOWN" id="CTEST.S1.XM24" font="italic">b<
      <XMTok role="UNKNOWN" id="CTEST.S1.XM25" font="italic">z<
    </XMApp>
    <XMApp>
      <XMApp argclose=")" argopen="(" separators=";;">
        <XMTok role="SUBSCRIPTOP" scriptpos="pre3"/>
      </XMApp>
    </XMApp>
  </XMDual>
</XMath>
```

Negatives?

- Like $\text{T}_{\text{E}}\text{X}$, but *not* $\text{T}_{\text{E}}\text{X}$.
- Packages need to be implemented for $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}\text{M}\text{L}$, or ignored.
- You may not like my style.
- Unfinished, but under active development.

So

- If it interests you, try it out:
<http://dlmf.nist.gov/LaTeXML/>