Structured Math On The Web

First 10 Years Of MathML

T. V. Raman

Google Research

Outline

- MathML —Turning the clock back to 1995
- XML, MathML and Extensibility
- State of mathematics on the Web in 2006
- Future of online mathematics

The Web In 1995

Web In Early 90’s

- Invented by technologists for technologists
- Online math was our obvious next step
- Web made of static HTML
- Goal: display Math in a Web browser

Started as the Math ERB

MathML Design Goals

Design a markup language that could

- Capture semantics for computation
- Capture richness needed by publications
- Enable aural presentations and other views

MathML: First XML Web vocabulary

Web In 1995

Commercial Web came to the forefront

- Shopping carts overtook Math equations
- Mainstream browsers ignored online math
- Web browsers had a primitive layout model
- Math typesetting is hard

Impedance mismatch between Web and online Math

Consequences

- MathML relegated to browser plugins
- Retarded ubiquitous deployment
- Authoring (LaTeX)TeX still easier than XML

Fixed tagset and extensible math do not mix well

Consequences Of XML

- XML is extensible; XML dialects are not.
- Fixed tagsets *always* lose semantics.
- Author-level extensibility important for Math.

Compare with (LaTeX)TeX.

XML, MathML And Extensibility

XML Benefits

- Well-understood document semantics
- Mixed vocabulary documents
- Internationalizable
- Extensible framework

MathML rode the XML wave.

The S-Expression That Wasn't

Thoughts of a disillusioned XML hacker

- XML failed to deliver simple S-expressions
- XML dialects are not *extensible*
- Fixed tagsets make new notation difficult.

An (LA)TEX View Of Extensibility

Authors need to invent new notation

- Math notation cannot be frozen in time
- Authors invent new notation
- New notation requires new markup

Lack of extensibility leads to presentational markup.

An (LaTeX)TeX Example

```
\newcommand\infer[2]{%
  \frac{#1}{#2}}
```

$$\frac{a < b}{b >= a}$$

Tag set extended incrementally

Example From AsTeR

AsTeR —semantic markup ⟶ rich aural renderings

```
(def-object :macro-name "infer"
  :args 2 :object-name infer
  :processing-function infer-expand
  :precedence  arrow-operator
  :supers ( binary-operator)
  :children-are-called
  '("premise" "conclusion"))
```

Declarative markup ⟶ semantic representation.

ASTER Rendering

Rendering rule for semantic representation:

```
(def-rule speak  (infer)
  (afl:new-block
   (speak  (arg 1 infer ))
   (speak " implies ")
   (speak (arg 2 infer))))
```

(La)TeX Advantages

- Authors invent notation and markup
- Semantics captured at authoring time
- Declarative markup separates processing
- While allowing author control over processing

This is still *difficult to do in browsers*

Web Browser Equivalent

Firefox equivalent would require:

- Creating a custom markup element,
- Binding it to MathML via XBL,
- Style it for presentation using CSS,
- Add behavior via Javascript,
- Package pieces into a coherent component.

Overhead precludes extensions by authors

HTML Extreme

Just do it with two elements.

```
<div class="infer">
<span class="premise">a<b</span>
<span class="conclusion">b>=a</span>
</div>
```

- Style with CSS
- Use CSS selectors to attach behavior

This solution cannot scale over time.

Publishing And The Web

- Print publishing provides reusable styles
- Contrast with Web publishing
- Interaction behaviors alien to publishing

Online Math victim of this impedance mismatch.

Math On The Web In 2006

Available Technologies

- Mixed namespace documents
- Plugin development is easier
- Augment declarative markup with behaviors
  - CSS for styling,
  - XPath, XSLT for rearranging content,
  - JavaScript for interactivity,
  - XBL for creating components

What Can We Build?

- Math Wikis?
- Wikipedia uses (LA)TEX for alt text
- Enable authors write simple math as (LA)TEX
- Map (LA)TEX to MathML DOM
- Enable blogging community author Math

Bring Math on the Web to the mainstream

Incremental Deployment

High-end math rendering will remain a hard *problem.*

- Package MathML as script libraries,
- Leverage SVG for final-form rendering,
- Enable online interaction via scripting,

Online Math need no longer be static.

On-ramp For Ubiquitous Math

- Use incremental deployment to make MathML essential
- Browser integration needs user demand
- Retain advantage of being an extension

What Can We Add To The Web

Evolving The Web

Demands of Math typesetting gave us (LaTeX)TeX

- Highly interactive online math
- Exploratory UI for interactive proofs
- Computing on online Math

Where do these requirements lead the Web?

Consequences For Mainstream

- Math more complex than plain text
- Fulfilling these impacts online education
- Potential for pioneering work

Create next hypertext revolution

Watch The Math Web Take-Off!