# Mathematical Content Conversions

David McKain

9th September 2009

# Outline

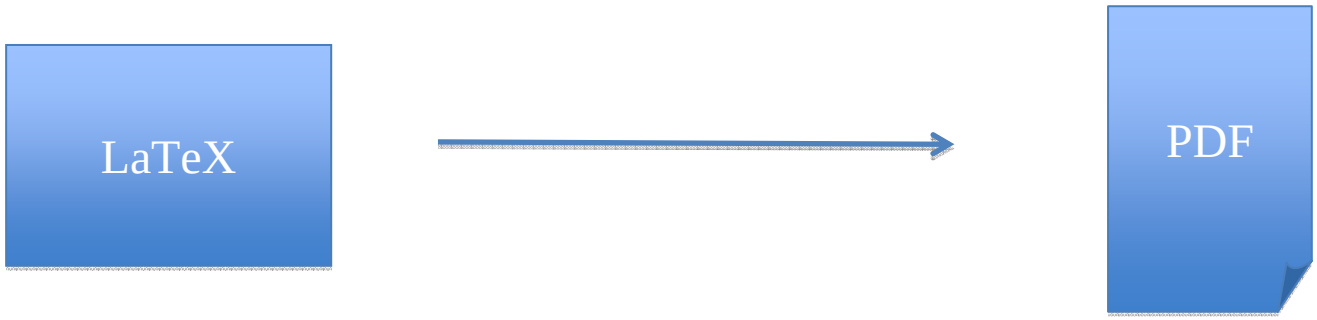
- Motivation
- Quick summary of mathematical formats
- LaTeX -> MathML conversion example
- Reality check
- Tools for LaTeX -> MathML
- Conversion techniques

# Motivation: Philosophy

- Lots of content already out there
  - in lots of different formats!
- One size never fits all
  - TeX/LaTeX: concise, easy to author, hard to manipulate
  - XML: verbose, hard(er) to author, easy to manipulate
- MathML is more complex than you think…
  - So needs converted between itself
- Prevent bit rot

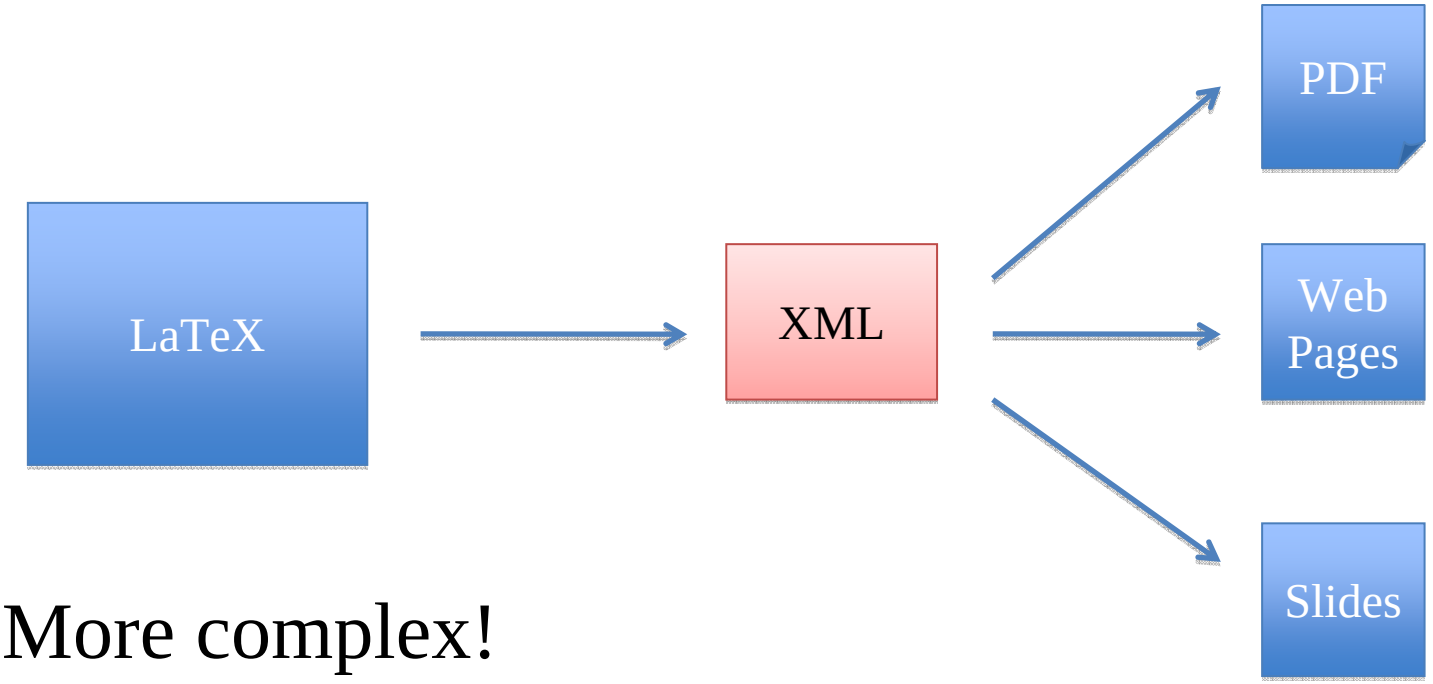
# Motivation: Practical examples

Bog standard LaTeX to PS/DVI/PDF



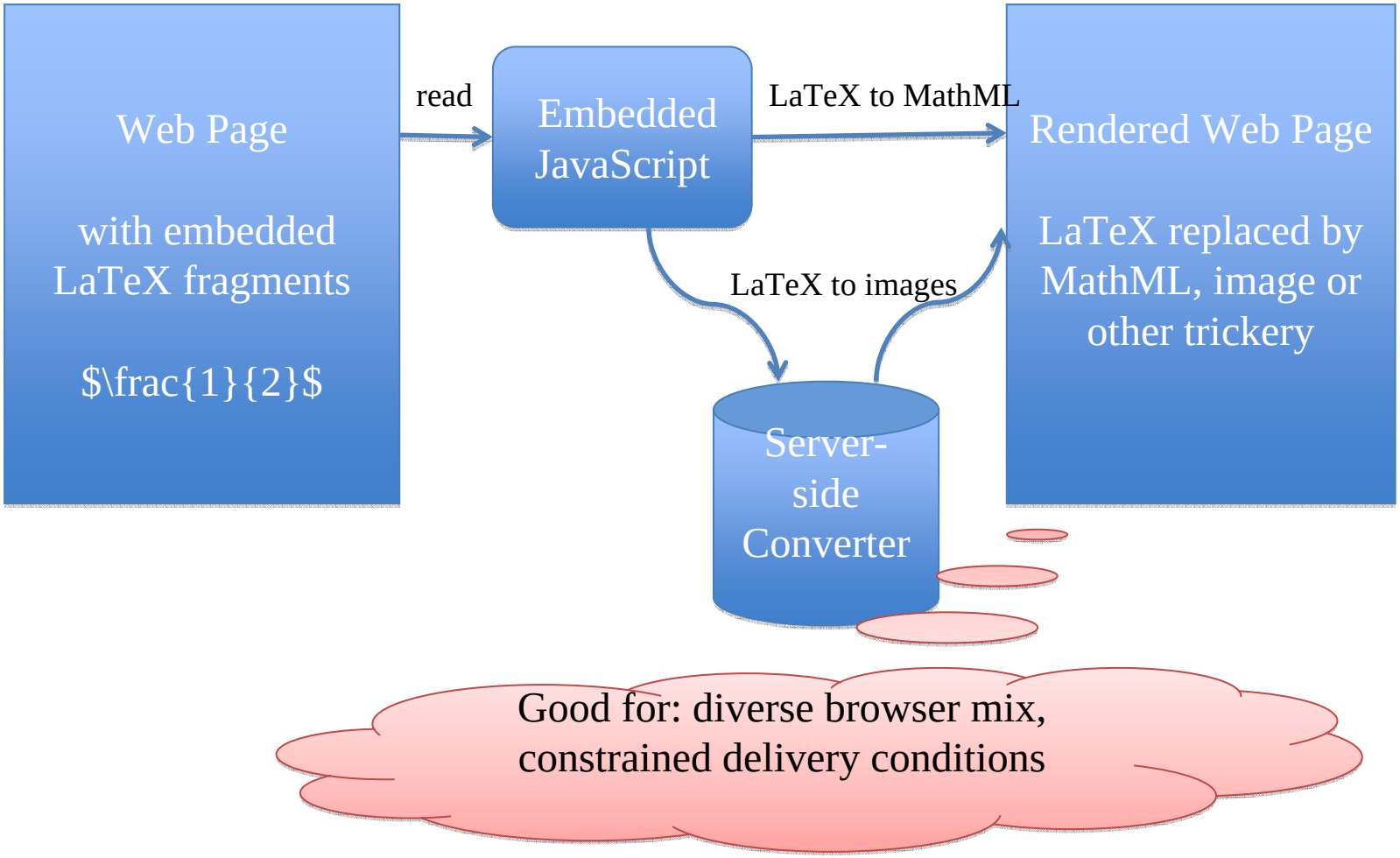

- This is really a conversion!
- LaTeX is both a **format** and a **process**
- Whole document conversion
- Easy turn-key solution

# Example: “Single Source, Multiple Output”

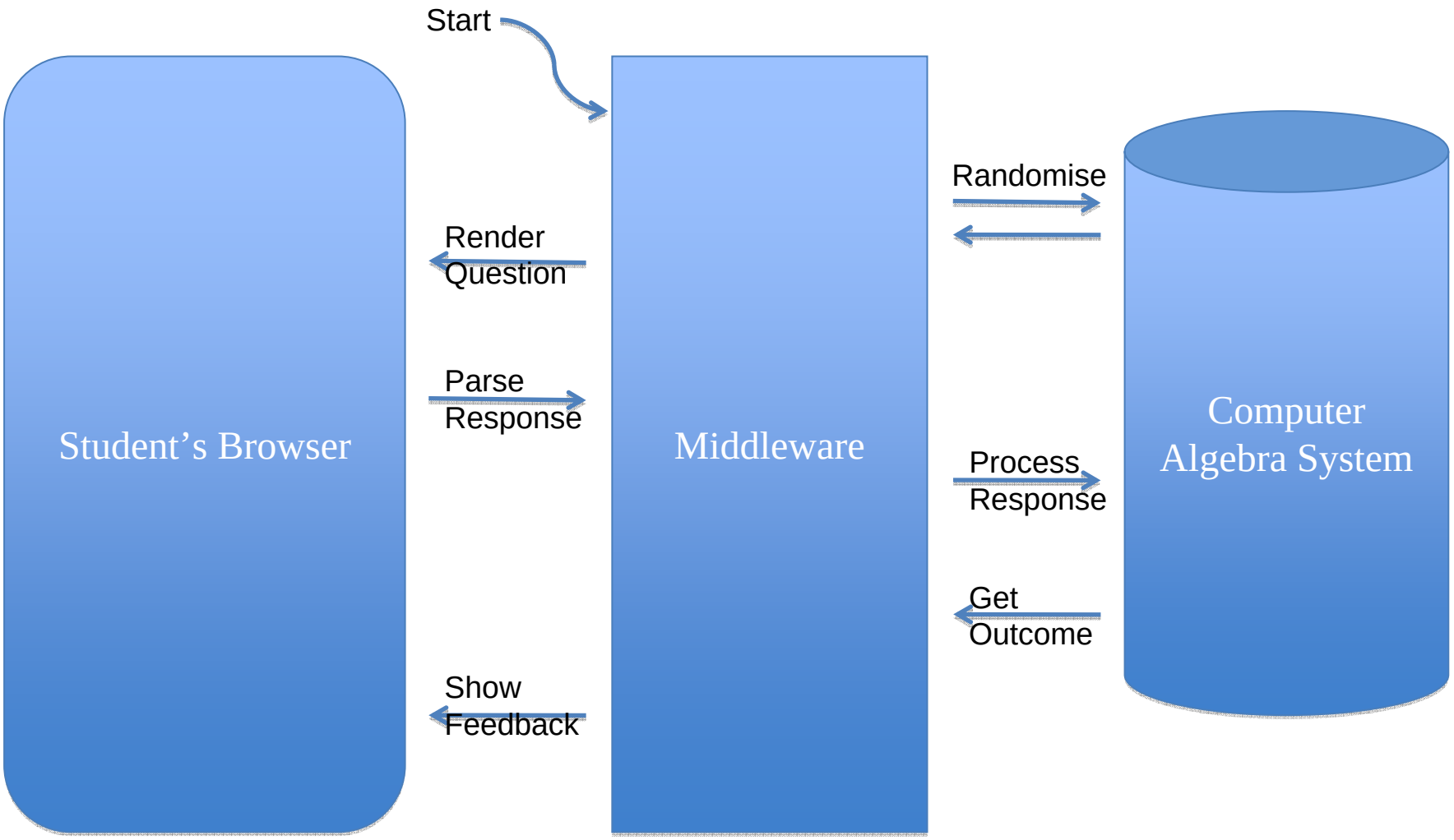



- More complex!
- Building block approach = useful
- Another “whole document” conversion
- Probably needs development effort

# Example: Dynamic maths in a browser

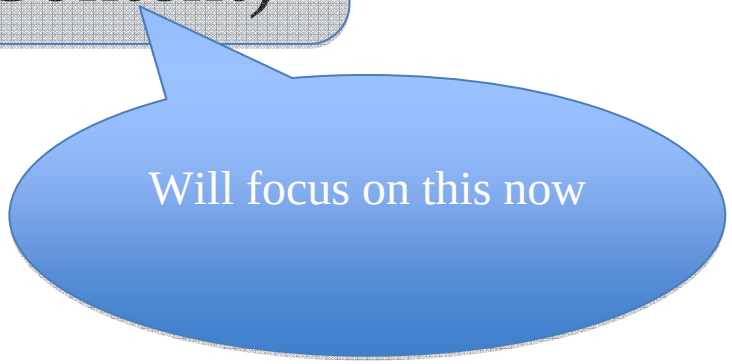


# Example: CAA question



# Typical formats

- TeX/LaTeX
- MathML (Presentation & Content)
- OpenMath
- HTML/XHTML
- Docbook
- Typesetting/display (PDF, PS, DVI)
- Other proprietary formats (e.g. Word)
- Computer Algebra System syntaxes



# Example: Conversion challenges

Consider the following function application:

$f(x+2\alpha)$

- We'll look at how this could be input using LaTeX and MathML
- This will demonstrate some of the challenges and limitations of any TeX <-> MathML conversion processes.

# LaTeX markup version

- This can be input in LaTeX (math mode) as:

```
f(x+2\alpha)
```

- LaTeX is most concerned about “display”
  - Its main function is to lay out each symbol in a rather beautiful way
  - It doesn’t care or know about “semantics”, e.g. whether brackets match or not
  - “Semantic” information can be specified though most people don’t bother

# Presentation MathML spectrum

```
<math>
  <mi>f</mi>
  <mo>(<mo>
  <mi>x</mi>
  <mo>+</mo>
  <mn>2</mn>
  <mi>α</mi>
  <mo>)</mo>
</math>
```

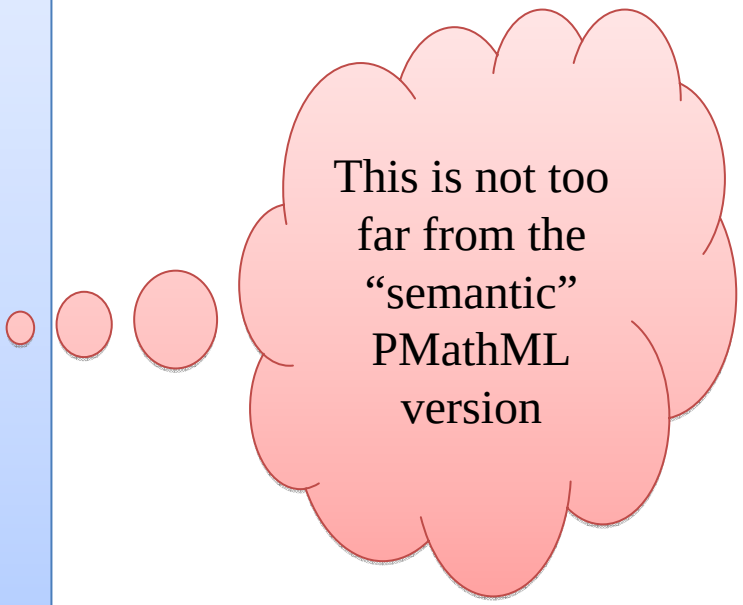
Similar to LaTeX!

```
<math>
  <mi>f</mi>
  <mo>&ApplyFunction;</mo>
  <mfenced open="(" close=")">
    <mi>x</mi>
    <mo>+</mo>
    <mrow>
      <mn>2</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>α</mi>
    </mrow>
  </mfenced>
</math>
```

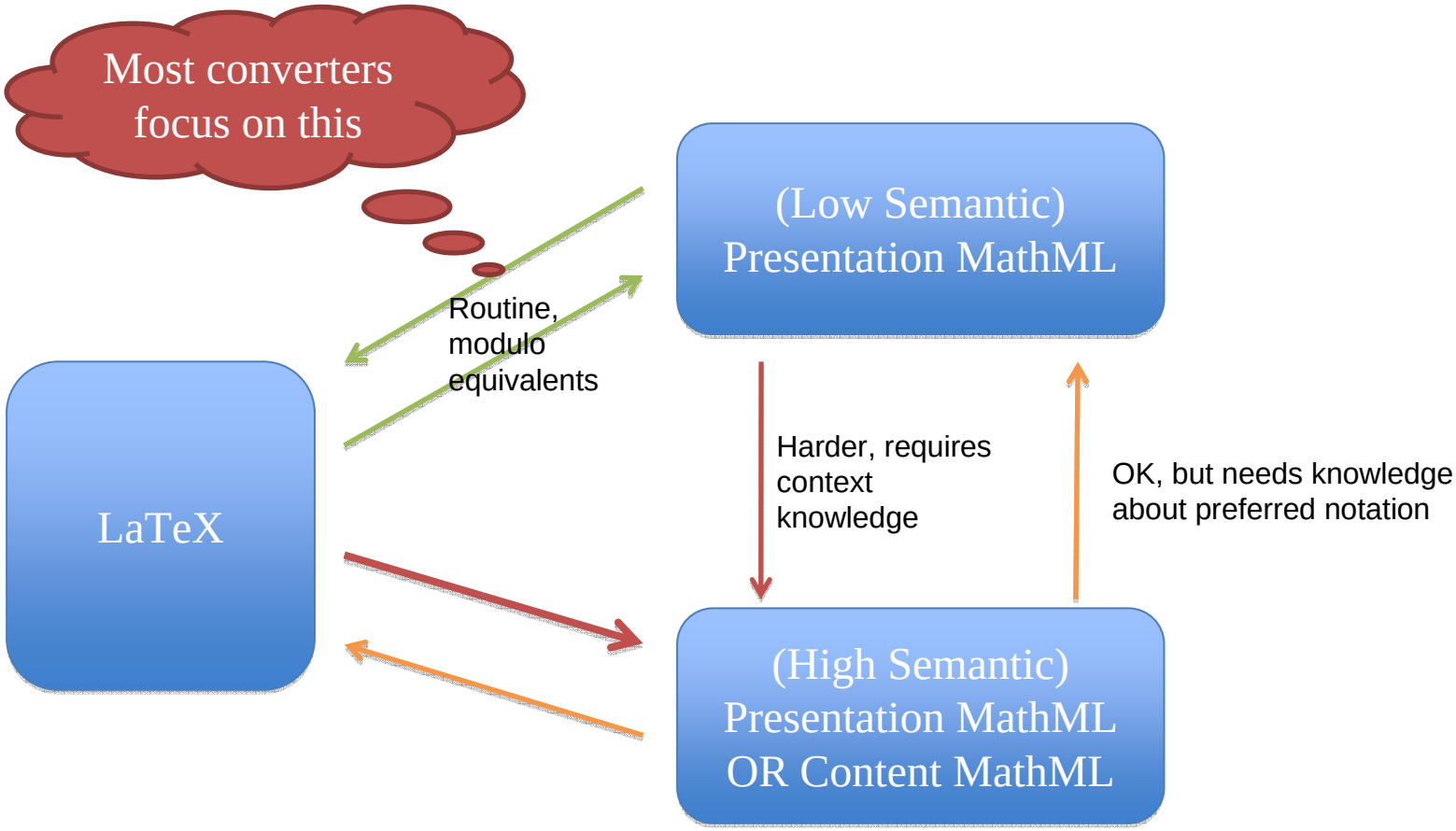
Encapsulates lots more meaning

# Content MathML version

```
<math>
  <apply>
    <ci type=“function”>f</ci>
    <apply>
      <plus/>
      <ci>x</ci>
      <apply>
        <times/>
        <cn>2</cn>
        <ci>α</ci>
      </apply>
    </apply>
  </apply>
</math>
```



# LaTeX <-> MathML possibilities



# Conversion challenges

- **Contextualisation:** $f(x+2\alpha)$ was a function earlier. What about $a(x+2\alpha)$? $e$?
- **Internationalisation:** $f'(x)$ vs $df/dx$, gcd vs ggT
- **Style: “**s.t.” vs “:” vs “|” vs bracketing
- **Notation Reuse:** 2|6 vs {n | n is prime}
- **Wiggle Room:** Presentation MathML representations are non-unique

# Conversion guidelines

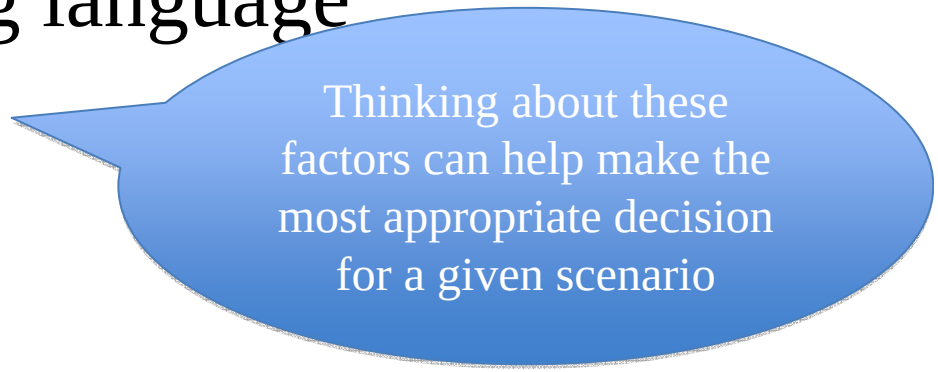
- Easiest to convert to a format with roughly same or lower semantic level
  - LaTeX <-> simple Presentation MathML
  - Semantic Presentation MathML <-> Content MathML
  - Content MathML -> OpenMath
- Conversion to higher semantic level possible if some of the “challenges” can be controlled:
  - Restrict to “UK school level algebra” and LaTeX to Content MathML is possible... kind of…
  - Has applications to e-Learning.

# Conversion tools

- There are lots of good tools out there, especially in the LaTeX -> simple PMathML direction.
- I’m not going to try to list them all. (Phew!)
- Some lists can be found at:
  - http://www.w3.org/Math/Software/mathml_software_cat_converters.html
  - http://www.cse.ohio-state.edu/~gurari/TeX4ht/
- Useful to consider some “discriminating factors” to help make an appropriate decision

# Discriminating factors

- Whole document vs. math fragments
- Level of TeX/LaTeX completeness
- Configurability/extensibility of outputs
- Software library vs. service
- Client (e.g. browser) vs. server
- Choice of programming language
- License/terms of use
- …



# Some (La)TeX-y to XML Converters (1)

| Name | Converts | Maths Output | Platform | Config? | License | Notes |
|---|---|---|---|---|---|---|
| TeX4ht | Whole doc | MathML, Images | TeX | Very | LPPL | |
| TtM | Doc | MathML, Images | C | | “Free” | |
| PlasTeX | Doc | Images | Python | “Renderers” | | Good for Python |
| LaTeXML | Doc | MathML, Images | Perl | Post-processor | Public Domain | |
| SnuggleTeX | Doc or Fragments | MathML, Images, XHTML+CSS | Java | Extensible, XSLT, DOM hooks | BSD | |
| GELLMU | Doc with special LaTeX markup | MathML, LaTeX | ELISP | Generates intermediate XML | GPL | Interesting approach |
| Tralics | Doc | MathML | C++ | | Custom | |

# Some (La)TeX-y to XML Converters (2)

| Name | Converts | Maths | Platform | Config? | License | Notes |
|---|---|---|---|---|---|---|
| MathTran | Fragments | Images | Web Service | | Not sure | Useful SOA idea |
| ORCCA | Fragments | XML | Java | Via Mapping file | ??? | |
| LaTeXMathML | Fragments | MathML | Browser JavaScript | As above | LGPL | Based on ASCIIMath |
| ASCIIMathML | Fragments | MathML | Browser JavaScript | Not really | LGPL | |

# Conversion techniques (geeky)

- Pipelining & reuse of components
- Build on what's already there… where possible
- Use XML early and often… if possible
  - Saves having to write custom parsers
  - Lots of XML tools already out there
  - XSLT & XPath are great. (Version 2.0 even better!)