MathML-aware article conversion from LaTeX. A comparison study.

Heinrich Stamerjohanns, Deyan Ginev, Catalin David, Dimitar Misev, Vladimir Zamdzhiev, Michael Kohlhase

Computer Science, Jacobs University Bremen
(first initial).(last name)@jacobs-university.de

July 9, 2009

The retro-born-digital story

- Publishing in Mathematics / theoretical Computer Science / Physics in the last two decades - using TEX/ LATEX
- Growing effort to make mathematical publications more adapted to the Web (than PostScript or PDF)
- MathML seems to be the format of choice for rich mathematical content on the web
- Several tools have been developed to convert TEX/ LATEXto Web and DML-compatible formats (XML + MathML)

Towards a better state of the art

In this paper we try to

- put the choice of converters for DML projects onto a more solid footing
- **encourage competition** and **feature convergence** by **surveying** the LaTeXto XML+MathML transformation market and **comparing** five available systems

The tools

Development seems to follow certain patterns:

- use the TEXengine to parse the LATEXsource
- reimplement a complete TEXparser

Different DML projects use different tools for conversion:

- the arxmliv project at Jacobs University uses LaTeXML
- the NUMDAM and CEDRAM projects use Tralics

Hermes

- A grammar based translator from LaTeX to Unicode(utf-8) encoded XML+MathML+metadata
- Semantically seeds a copy of the TeX source
- Uses LaTeX during the conversion and parses the resulting (output) semantic DVI file
- MathML is the only XML vocabulary supported by Hermes
- Written in C, uses bison and flex.
- Licenced under GPL, and developed by Romeo Anghelache (`http://romeo.roua.org/`). It's development was partially supported by Max Planck Institute for Gravitational Physics, EDPSciences and Design Science

Hermes

Pros

- Easy to install

Cons

- Development discontinued as of 2006
- Little documentation available
- Conversion speed is slow mostly due to the use of LaTeX
- No debugging support
- Needs 4 commands to convert a single file

Tralics

- Translates LaTeX into a custom XML representation with an outlook for a successive conversion to PDF or HTML.
- The software is readily available online and is deployable both from source or a respective binary for either Linux, Mac OS or Windows.
- Extensive documentation available for the developers, but little for the normal users.
- Tralics uses the TeX parser to expand the document recursively, until the pages have been constructed. Consequently, the C++ engine constructs the XML document tree and converts the mathematics to MathML.
- The author of Tralics is Jose Grimm (Jose.Grimm@sophia.inria.fr)

Tralics

Pros

- highly customizable
- supports many output formats

Cons

- needs bindings
- difficult to setup for proper use
- currently low TEX coverage

LaTeXML

- Developed to support *Digital Library of Mathematical Functions* by Bruce Miller (b.miller@nist.gov)
- Written in Perl, tries to emulate LaTeX
- Consists of a TeXemulator, XML emitter and post-processor (latexmlpost)
- LaTeXML processes a TeXor LaTeXdocument and outputs a temporary LTXML document (XML representation of the LaTeXcounterparts
- latexmlpost transforms the LTXML file in a .xhtml file containing the desired format
- Available online as source and packages

LaTeXML

Pros

- very detailed manual
- very well supported mathematical elements
- good configuration and debugging support
- actively developed and supported

Cons

- needs bindings
- rather slow, bindings and Perl modules need to be reloaded each time

TeX4ht

- Developed by Eitan M. Gurari
- Based on TEXto produce the output
- Actual script written in C
- Uses the output of LATEXto generate XHTML + MathML
- Supports a multitude of output formats (HTML, XHTML etc.)
- Widely available to the public (website, packaged for various Linux distributions)
- Since it uses LATEX, it actually supports all the LATEXconstructs

TeX4ht

Pros

- high degree of support for LaTeXconstructs, does not require further bindings
- many output formats
- very well documented
- easy to install, easy to use

Cons

- highly dependent on LaTeX
- lack of debugging support
- speed of conversion
- written in C, but LaTeXinvocations last very long

TtM

- TtM is a TeX to MathML translator that is derived from the HTML translator TtH.
- Works by imitating how LaTeX or TeX work, and is not specifically dependent upon any programs being installed on the system.
- Written using the flex language, from which a C executable is produced, so it's extremely fast in default mode.
- Almost all of TeX's and LaTeX's mathematics is supported.
- Macro definitions are fully supported, however TtM does not understand TeX category codes (catcodes).
- TtM is copyright (c) Ian Hutchinson (hutch@psfc.mit.edu)

TtM

Pros

- portable
- fast
- very well documented
- easy to install, easy to use

Cons

- doesn't recognize catcodes
  - thus low success conversion rate in general
- windows version is not free

Other systems

1. the `BlaTeX`, `itex2mml`, `RiTeX`, MathMLStudio Lite only convert a subset of TEX/LATEX formulae to MathML, but do not seem to have a document level. They are more directed towards authors of mathematical documents on the web rather than born digital DML efforts.
2. The `HeVeA`, and `LaTeX2html`, transform LaTeX documents to HTML, but do not seem to generate MathML output.
3. ORCA - an online service by The University of Western Ontario
4. `LXir` by EDP Sciences under the GPL. Usage instructions are only available in French
5. `Omega` has been discontinued and seems to be merged into `LuaTeX`.

Evaluation

- Five programs have been studied that transform TEX/LATEX sources into XML and can create MathML.
- Systems were compared in three dimensions:
  - Ergonomic factors like documentation, ease of installation
  - Coverage
  - Quality of the resulting documents
- To obtain an objective measure of the latter two, we tested all systems on a set of 1000 articles randomly picked from the arxiv ePrint server.
- The results are summarized in the final table

Coverage of the converters

- Choice: arXiv corpus - one of the most comprehensive sources of heterogeneous TEX/ LATEXdocuments
- More than **500,000 documents** from various fields (e.g.: Physics, Mathematics, Computer Science etc.)
- Still, running the converters on the whole corpus would take very long (orders of processor-year)
- Choose **1000 random documents** from the corpus
- The behavior of the converters can be categorized:
  - **incomplete** the converter did not finish the transformation (fatal error, crash)
  - **complete with errors** the converter finished transforming, but signalled errors
  - **success** the converter finished processing with no problems or only warnings

MathML quality

- Overall methodology: establish a FQC ("Formula Quality test Corpus") based on a small set of non-trivial formulae randomly chosen from the arXiv corpus
- Example:

$$
\begin{aligned}
4r^2 \int_0^{\pi/2} \cos^2\theta \, d\theta &= 4r^2 \int_0^{\pi/2} \frac{1}{2}(1+\cos 2\theta) \, d\theta \\
&= 2r^2\theta\Big|_0^{\pi/2} + 2r^2 \int_0^{\pi/2} \cos 2\theta \, d\theta \\
&= \pi r^2 + 2r^2(\sin 2\theta)\Big|_0^{\pi/2} \\
&= \pi r^2
\end{aligned}
$$

Figure: An example formula to check the quality of the converters

MathML quality criteria

- XHTML + MathML quality
  - Presentation vs Content MathML
  - formula tree quality
  - CSS usage
- Is the resulting XML valid ?
- Are formulae like $x + y^2$ semantically disambiguated
  - $+$ is an operator
  - x and y are variables
  - ^{2} means squared

Test case: Eqnarray* environment

$$\begin{aligned} 4r^2\int_0^{\pi/2}\cos^2\theta d\theta &= 4r^2\int_0^{\pi/2}\frac{1}{2}(1+\cos2\theta)d\theta \\ &= 2r^2\theta|_0^{\pi/2}+2r^2\int_0^{\pi/2}\cos2\theta d\theta \\ &= \pi r^2+2r^2(\sin2\theta)|_0^{\pi/2} \\ &= \pi r^2 \end{aligned}$$

(a) Hermes

$$\begin{aligned} 4r^2\int_0^{\pi/2}\cos^2\theta\,d\theta &= 4r^2\int_0^{\pi/2}\frac{1}{2}(1+\cos2\theta)\,d\theta \\ &= 2r^2\theta\Big|_0^{\pi/2}+2r^2\int_0^{\pi/2}\cos2\theta\,d\theta \\ &= \pi r^2+2r^2(\sin2\theta)\Big|_0^{\pi/2} \\ &= \pi r^2 \end{aligned}$$

(b) Tralics

$$\begin{aligned} 4r^2\int_0^{\pi/2}\cos^2\theta d\theta &= 4r^2\int_0^{\pi/2}\frac{1}{2}(1+\cos2\theta)d\theta \\ &= 2r^2\theta|_0^{\pi/2}+2r^2\int_0^{\pi/2}\cos2\theta d\theta \\ &= \pi r^2+2r^2(\sin2\theta)|_0^{\pi/2} \\ &= \pi r^2 \end{aligned}$$

(c) LaTeXML

$$\begin{aligned} 4r^2\int_0^{\pi/2}\cos^2\theta\,d\theta &= 4r^2\int_0^{\pi/2}\frac{1}{2}(1+\cos2\theta)\,d\theta \\ &= 2r^2\theta\Big|_0^{\pi/2}+2r^2\int_0^{\pi/2}\cos2\theta\,d\theta \\ &= \pi r^2+2r^2(\sin2\theta)\Big|_0^{\pi/2} \\ &= \pi r^2 \end{aligned}$$

(d) tex4ht

$$\begin{aligned} 4r^2\int_0^{\pi/2}\cos^2\theta\ d\theta &= 4r^2\int_0^{\pi/2}\frac{1}{2}(1+\cos2\theta)\ d\theta \\ &= 2r^2\theta|_0^{\pi/2}+2r^2\int_0^{\pi/2}\cos2\theta\ d\theta \\ &= \pi r^2+2r^2(\sin2\theta)|_0^{\pi/2} \\ &= \pi r^2 \end{aligned}$$

(e) TtM

Test case details

- Representing Eqnarray (requires a table representation in XHTML) solutions:
  - use MathML $< mtable >$: Tralics, tex4ht, TTM
  - use HTML $< table >$: LaTeXML
  - both : Hermes
- Operators and symbols
  - start with the $< mo >$ element (Hermes)
  - add attributes: tex4ht - "class" attribute (better rendering), Tralics - "form" attribute (better positioning)
  - LaTeXML - "movablelimits" to achieve a deterministic rendering of scripts
- Other (e.g.: Math spaces, integrals, noise)

Evaluation

System	Documentation	Installation	**Coverage**	% incomplete	% with errors	% success	Quality	Speed	Ease of Use	Extra
Hermes	–	++	–	65	0	35	–	o	–	–
Tralics	+	++	–	0	98	2	o	+	– –	o
LaTeXML	++	+	**+**	10	36	54	+	–	+	+
TeX4HT	++	++	**o**	34	28	38	++	–	++	++
Ttm	++	++	–	27	65	8	o	++	+	–

Figure: Comparison Table for the systems

Conclusion

- Current efforts are independent and isolated
- Grounds for collaboration
  - The different converters introduce different good features
  - Various approaches to MathML symbol generation, most could be integrated together
- Heterogeneously motivated, the different tools have different strengths in terms of:
  - breadth: defined macros
  - depth: quality of conversion
  - usability: user base and documentation
  - performance: online vs offline use case