Tools for Converting LaTeX to XML

DECEMBER 10, 2008

I spent some time surveying the available tools for converting LaTeX to XHTML+MathML or, more generally, LaTeX to XML. My criteria were the following:

- The project must be free and open source.
- It should produce clean, semantic XHTML+MathML or XML output.
- It should be able to handle macro definitions using standard LaTeX commands.
- The possibility of adding support for additional LaTeX packages (e.g., `natbib` or `hyperref`) is a plus.
- Tools that require little or no manual intervention or modification of the LaTeX source are preferred.

I was pleasantly surprised at how many projects I found that actually met most of these criteria. Those that I am aware of are described in more detail below.

Overall, I was most impressed with LaTeXML. In my opinion, its usage is the most straightforward and it produces very clean, general XML output which can produce easily customizable XHTML+MathML documents.

LaTeXML

LaTeXML is Perl module which parses the actual LaTeX document and emits XML output for later post-processing (for example, for conversion to XHTML+MathML). With a proper XSLT stylesheet, one can obtain custom XHTML+MathML output. The LaTeXML homepage itself was generated using LaTeXML. The project is still active (at the time of writing), it's very well documented and it has a Trac and a mailing list.

If you use Debian GNU/Linux, you can install the relevant dependencies with

```
sudo apt-get install libparse-recdescent-perl libimage-magick-perl \
  libxml-libxml-common-perl libxml-libxslt-perl
```

The package is installed using the usual procedure for Perl modules:

```
perl Makefile.PL
make
make test
sudo make install
```

The usage is straightforward. First convert the LaTeX document, say `mydoc.tex` to XML and then post-process the XML, converting it to XHTML+MathML:

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

LaTeXML is a project of the NIST and is therefore in the public domain.

Tralics

Tralics is written in C++ and also directly parses the LaTeX source (and it's also extremely fast). It is licensed under the French CeCill open source license which is GPL-compatible.

Compiling it is straightforward:

```
tar zxvf tralics-src-2.13.5.tar.gz
cd tralics-2.13.5/src
make
```

To convert a LaTeX document to XML:

```
tralics doc.tex
```

A file called `doc.xml` will be created. Tralics handles any unknown commands from unsupported package such as `hyperref`, for example, by including an `<error>` tag:

```
<error n='\hypersetup' l='35' c='Undefined command'/>
```

So, apparently it should never fail to parse the document as long as it is valid LaTeX.

The XML file can then be converted to XHTML+MathML using a stylesheet. Several examples are provided in the "Extra files" package.

Hermes

Hermes is a grammar-based DVI-parser for translating LaTeX to Unicode-encoded XML+MathML. It works by first including a set of TeX macros in the original LaTeX document which insert specials in the DVI file. It then constructs XML output by parsing the semantic DVI file.

Some examples are provided here. In particular, there is a collection of articles from arxiv-math that were translated to XHTML+MathML.

Hermes is very complete in terms of functionality, but there are still a few glitches here and there, namely it has trouble handling spaces properly (see some of the examples). It also requires two steps just to get the XML file as you first have to create a "seed" LaTeX document (which essentially just adds a line `\include dtx` line which includes the extra macro definitions).

TeX4ht

TeX4ht, available in the Debian package `tex4ht`, is probably the most widely used LaTeX to (X)HTML tool. It supports conversion to HTML, XHTML+MathML, OpenDocument, and DocBook. Direct XHTML+MathML conversion is possible using a command like the following:

```
htlatex filename "xhtml,mathml" " -cunihtf" "-cvalidate"
```

See the documentation for details about the available options.

The direct XHTML+MathML conversion looks very nice but the output didn't seem very clean or semantic. It seems that it's possible to heavily customize the output if you like, but the methods for doing so aren't exactly obvious. I didn't test its DocBook conversion, although this may also be a promising route.

LXir

LXir is another DVI-parsing LaTeX to XML translator. You must first include `\RequirePackage{lxir}` in your LaTeX document and run `latex` to obtain a DVI file. Then running `lxir doc.dvi` will produce an XML file that can be processed using `xsltproc`.

LXir looks promising but it still has some problems. It will fail if it encounters commands from any unsupported packages. Even after removing all external package dependencies from my document, LXir still failed to process the standard `\author{foo \and bar}` structure. Once I removed that, there were still errors in the generated MathML.

Overall LXir looks promising, and I think it's a project worth keeping an eye on, but it doesn't seem ready for production use (at least not for anything containing mathematics).

GELLMU

There is also an alternative markup language called GELLMU which supports XHTML+MathML, HTML, PDF, and DVI output. While it does meet most of my criteria, I'd rather be able to write real LaTeX, rather than pseudo-LaTeX. It's certainly debatable but I consider LaTeX to be an archival format. At the very least an acceptable LaTeX-to-XML tool will eventually emerge. Clean LaTeX code is very structured and LaTeX is going to be with us for a very long time. Thus, it would simplify things if I were able to store my originals in LaTeX format.