System Description: A Semantics-Aware LaTeX-to-DOCX/ODF Converter

Lukas Kohlhase and Michael Kohlhase

Mathematics/Computer Science
Jacobs University Bremen

Abstract. We present a LaTeX-to-Office conversion plugin for LaTeXML ML that can bridge the divide between publication practices in the theoretical disciplines (LaTeX) and the applied ones (predominantly Office). The advantage of this plugin over other converters is that LaTeXML conserves enough of the document- and formula structure, that the transformed structures can be edited and processed further.

1 Problem & State of the Art

Many researchers in STEM fields only use LaTeX to typeset their documents. However many people still use Microsoft Word/Open office exclusively for their typesetting. When these two groups of people intersect, it can lead to friction, as transforming text to LaTeX is quite trivial but not the opposite. For example if a conference requested documents in Word format, the only recourse is often to just write the document in Word, which is a pain, especially if any Mathematics is to be included.

[1] EdN:1

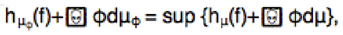
copy from PDF	paste (libreoffice)
$h_{\mu_\varphi}(f) + \int_X \varphi d\mu_\varphi = \sup_{\mathcal{M}(f,X)} \{h_\mu(f) + \int_X \varphi d\mu\},$	 h

Fig. 1. Copy & Paste in Word Processors

There are several methods to transform papers from LaTeX to an office word processor. The first method is to just generate a PDF file and then open this file in Word/LibreOffice. This achieves the goal of looking like the desired PDF document, just in Office. There are two problems with this route:

1. mathematical formulae are not preserved (see Figure 1)

[1] EdNote: Here we state the Problem, some conferences and admin want papers in word format, however LaTeX is superior for various reasons. Hence converter is needed. Two step process, wastes some time.

2. even if the result looks OK the results have lost their links (e.g. for citations/references or label/ref), or become difficult to edit, because they do not conform to the styling system of the word processor.

The fundamental problem is that it converts the appearance of the document and loses meaning due to macro expansion. This is especially blatant when looking at the math in a document. Either it is treated as text, with no meaningful way to distinguish between math and formatted text that happens to contain some mathematical symbols, making automatic treatment of this kind of math difficult, or it is represented by an image of the relevant formulae, which makes editing extremely impractical if not impossible. The same holds true for references, they are essentially treated as parts of text with a linked number in front of them, complicating adding new references substantially.

The other way of transforming LaTeX to Word, by transforming the .tex file directly, does away with some of these issues. Some editors to do this already exist, such as TeX4ht [T4HT]. This already does this admirably, however it is very focussed on Libreoffice, e.g. it can't handle mathematics in docx files.

2 Implementation

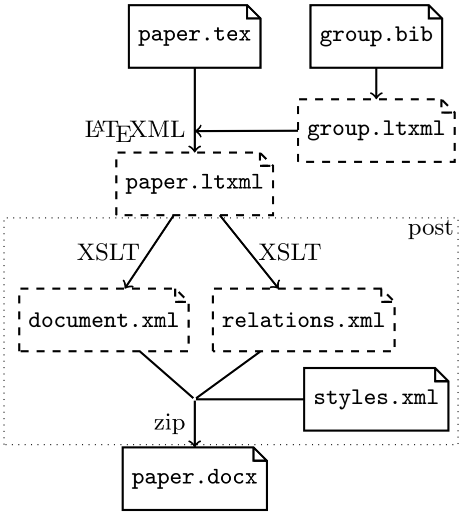


Fig. 2. The Transformation Process

Both docx and odt files share a very similar structure and are almost interchangeable, except for slight differences in syntax and different names. They both consist of zipped up XML files. The main content, such as text, placement of images, tables etc., is written in document.xml. The other important file is relations.xml, which contains information about where in the docx/odt file other supplementary files such as images are contained. Finally the archive contains various other objects such as style files, setting files and images.

To create the .odt/.docx files we first transform the .tex file to an intermediate XML-based format usingLaTeXml. [2]. EdN:2

Then we use an XSLT stylesheet to generate document.xml from the .ltmxl file. For Word files, we use a Microsoft stylesheet to transform the MathML generated by LaTeXML to the docx math format. The other file we generate from the ltxml file using XSLT is relations.xml.The other supporting files such as images are placed into the correct file structure the postprocessor. As the penultimate step some static files, that don't change depending on the input document, are also placed into the correct directories.The main file of interest here is styles.xml, which contains the style information of the document. We had to create this ourselves to recreate the feel of the PDF files generated by LaTeX. Finally the document is zipped to create the docx/odt file.

[3] [4] EdN:3 EdN:4

3 Conclusion

In Conclusion we use LaTeXML and XSLT to transform LaTeX files to Word/Office files semantically in an easy to use process. [5] EdN:5

The LaTeXML Word Processing plugin is public domain and is available from GitHub at [L2O]

References

[L2O] GitHub repository. URL: `https://github.com/KWARC/LaTeXML-Plugin-Doc`.

[T4HT] *TeX4ht: LaTeX and TeX for Hypertext.* URL: `http://www.tug.org/applications/tex4ht/mn.html` (visited on 01/08/2010).

[2] EdNote: Papa Latexml erlaerung einfuegen

[3] EdNote: Screenshot einfuegen

[4] EdNote: Bin eigentlich nicht zufrieden hiermit TT

[5] EdNote: In Conclusion, easy to use