Lemon8 XML beta released

The PKP people have released a beta of Lemon8-XML, (L8X) their journal-oriented word processor-driven XML publishing system.

I tried out the demo server with an ICE test document.

The bad news is that the service had significant problems with my document; It could not locate author metadata, incorrectly identified some ordinary text as being citations, and lost most of the document text, which is obviously a very major issue.

The good news is that MJ Suhonos from PKP was onto me straight away with an email and is keen to work on support for styles in general and ICE styles in particular. (It's in the FAQ that we will collaborate on this).

If the PKP team can get a decent structure guessing application to work on arbitrary input that would be great, but even better would be to close the loop and give back documents with more structure than you put in. At the ICE project we will help however we can.

If it was me doing this I would break this problem into two parts:

1. Build a converter that can take **structured** word processing documents and map them to the NLM XML format used by L8X. ICE offers one well worked out structure for generic documents, others may exist for specific formats.
2. Build a structure-guessing application to **add structure** to word processing documents (something which Ian Barnes has been chipping away at for a while).

With both of these in place you can improve documents in the wild as you go; every time someone submits a draft add styles and give it back to them, rather than trying to guess structure at the end. I would like to see this embedded in the OJS journal management system from PKP so that authors get rapid and continual feedback every time they upload a draft. This would allow some editorial and review processes to take place in an HTML interface as well – rather than via PDF on word processing files.

If you leave L8X as the final step, authors will have little feedback as to how they can improve the structure of their drafts.

My two-part plan would re-ordering sections in L8X become redundant – word processors have outlining tools with which you can reorder content, so why try to do it through an HTML interface?

On a technical note, last time I looked at L8X I concluded that Docvert is a weak link – it tries to to use XSLT to guess structure; our experience with ICE was that XSLT (version one at least) was not a productive way to do this as the austere functional programming environment in XSLT made the structure-reasoning code very hard to maintain and very slow, so we moved to more traditional parser written in Python which is much easier for typical programmers to work with.