A few words on magic

MJ Suhonos from PKP has patiently explained where I got some things wrong about Lemon8XML in my previous hasty post.

I'd like to pick up one theme from MJ's post. MJ says (with emphasis by me):

> The larger problem, of course, is that **L8X is encumbered, in a way, by the common expectation that it should just "magically" work on whatever format the author or user is providing** -- it is an application that is designed to solve, in part, an infinitely-unsolvable problem. So, the user has to meet the application halfway.

I agree that this expectation that tools should perform **magic** is a problem. We see this in the HTML export from word processors; they take arbitrary input and turn it into HTML. In the inevitable absence of magic you typically get sub-standard output.

I understand the requirement to try to understand the structure of ad hoc documents if you can, but I don't think it's a good idea to encourage people to keep creating them; if L8X has a version of "meet me half way" which involves direct formatting instead of styles then that will be a step backwards in my opinion. My version of meet me half way would be at least to try to get people to use headings. If they don't then the structure guesser will step in, try to guess and **give them their document back to correct** when the inevitable errors occur.

I took a look at the single sample document for L8X on the demo site. It's clear that the structure-guesser part of the application is going to have to be very clever to work well. It seems, for example, that the goal is to detect captions either before or after a graphic or table even when they have no special formatting. Introducing edge cases like short paragraphs both before and after an image seem to cause it problems, including loss of text but I could be wrong, again.

(I've had a look at the document parser code and it is taking into account paragraph length, and doing some reasoning based on text-size and formatting attributes).

So, even though I had some of the architecture wrong, I **still** think that Lemon8 XML would be vastly more useful if it had a two part architecture:

1. **Styled word processing document to XML conversion**, with the obvious caveat that if you're turing a generic format into a domain specific one you're going to be producing stuff that doesn't use the whole of the target format and may have gaps that need to be filled in.

   Lemon8 XML has its own XML format, but I'm wondering if it couldn't just use ODF which is a well specified standard, with the ability to give the document back to the user. (Checking with MJ via email about this).

   The goal would be to get as many people using this mode as possible because it is the least work for everyone – no guessing strucutre required if people can use markup.

2. **Ad hoc-formatting to styled word processing conversion** using the best available heuristics to guess structure and **give the document back to the author in an improved form**. As far as I can tell that's not a goal for the PKP team, but the code is out there so we could do it, using their algorithm. We're looking into it.

It is important to help our colleagues who are authoring documents in word processors to use

styles. It's good for them. It will improve their working lives. And it will open the door for them to start dealing with real eResearch and the semantic web. A project like the TheOREM-ICE would be impossible with documents like the L8X sample document.