Some comments on OOXML, ODF and Microsoft Word

There is a conversation about file formats and word processors going on between Peter Murray-Rust and Glyn Moody.

Peter made some comments about wanting access to chemistry publications in Word format so he can better extract chemical information embedded in them, which sparked some push-back regarding the Microsoft OOXML format.

Unsurprisingly I have a few opinions on all of this. In this post I'll:

1. Outline the state of conversion software between OOXML and ODF. Specifically, .odt and .docx, the word processor formats.
2. Look at what we can do to support researchers right now.
3. Comment on the issues Glyn Moody raises re Microsoft lock-in and OOXML.

File format conversions

Peter writes:

> I may be optimstic but it [OOXML] can also be converted to ODT. See the WP entry:
>
> http://wwmm.ch.cam.ac.uk/blogs/murrayrust/?p=1078

Yep. He's optimistic. There are two freely available converters that I know of, but both of them have a number of practical issues, starting with which one to choose?

The most mature converter is the ODF Add-in for Microsoft Word, Excel, and PowerPoint which is open source but sponsored by Microsoft. I put together a table to try to get my head around the options. All the converters use the ODF add-in apart from the last one. I only care about the word processor format, so that's what I looked at.

Platform	**Application**	**What does it do?**
Windows	Microsoft Word 2003+	The ODF add-in will read .odt files and turn them into .docx. It's slow. (I couldn't make it run on my version of Word 2007 because of what I think are conflicting versions of .NET)
Windows / Linux	OpenOffice.org or another ODF aware application (but see below – are there any?)	OpenOffice.org Ninja; a little program that intervenes when you click on a .docx file and converts it it .odt, slowly.
Windows / Linux	Novell's version of OpenOffice.org writer	Uses the ODF Addin and allows you to edit a .docx file. Open and save are, you guessed it, slow.

Mac OS X	Microsoft Word	Currently no options for reading ODF AFAIK
Mac OS X	NeoOffice	Contains a the Novell plugin, so you can open .docx files (slowly, of course).
Windows, OS X, Linux	OpenOffice.org Writer version 3 (currently in beta)	Contains a new different converter which is much faster than the ODF add-in. But Sun are not aiming to provide round-trip editing of .docx files. This is intended to be an import filter only.

So how to the two technologies perform?

You know, I'm pleasantly surprised to to be reporting that using NeoOffice on the Mac a complex ICE test document seemed to round trip from Neo to Word 2008, where I made a few changes and then back to Neo with no visible problems. This ODF add-in thing has improved a lot since the first time I tried it.

But I think Microsoft's approach of sponsoring an open source project instead of doing the work themselves is very dodgy. The ODF converter site is hard to use, lots of the documentation is in Word and Excel downloads instead of proper web pages and it requires an obsolete version of the .NET framework to run. The site is also big on frightening lists of incompatibilities between the formats.

All this seems to me to be designed to say "Told you so! Word and ODF are not compatible cos they use fundamentally different document models."

There will never be 100% translation of all the stuff you can do in the two formats because they are **both** built around existing implementations that had different feature sets and different document models. Gly Moody puts it like this:

> They probably won't ever work very well because of the proprietary nature of the OOXML format: there's just too much gunk in there ever to convert it cleanly to anything.
>
> http://opendotdotdot.blogspot.com/2008/05/word-in-your-ear.html

You can call it *gunk* but it's just the reality of a legacy format. Me, I'm glad to have all the gunk out in the open.

But it's not just OOXML that's gunky. Take a look at the list model in ODF some time; it has this hierarchical list style model that comes straight out of OpenOffice.org version 1. The user interface in Writer has never been any good at dealing with these list styles because the whole thing is just wrong for a word processor. KWord just ignores them, meaning that it won't interoperate with Writer even though the developers claim ODF support. Gunk.

Back to testing the ODF converter add-in. Using an ICE document which is all styles based, the NeoOffice version of the ODF add-in converter seems to pretty much get it right. I'm impressed.

So how about Sun's importer? In the beta of Writer 3 it has only one advantage over the ODF add-in. It's fast. Fast but broken – lists and document outlines are an absolute mess. Of

course we already have a document repair function for documents that follow the ICE style conventions so we could work around this with our users, but it's not ready for use in the real world.

And the politics! Sun have OOXML import in OpenOffice.org version 3 but don't plan to give you the ability to edit OOXML files directly. And the Microsoft sponsored project takes a similar approach. You can open ODF files but they automatically get turned into OOXML although you can at least save to .odt.

What can we do now?

Now the obligatory ICE-plug.

In the ICE project we have devised a set of styles and associated templates that can be used for interoperable document authoring. Like Peter Murray-Rust I want to support Word because that's what researchers are using. Unlike Peter Murray-Rust my group has not accepted any money from Microsoft.

Yet.

We have defined a set of structural and semantic styles that you can use in any word processor – then even if the applications can't understand each others formatting in detail they can still understand **your** documents. If you use the ICE toolbar and follow a few guidelines then we can make good quality (X)HTML from your documents and more-or-less move documents from ODF to Word and back.

Not only that, we're building plugins that understand various semantics, like CML for chemistry, graphs and tables, geographical information.

What we do currently is load Word documents into OpenOffice.org via the old .doc format, save them to .odt (OpenDocument Format Text) and run some fix-up code over them. The fixups do things like remove spurious list-styles that Writer adds and fix the document outline. We have not tackled OOXML, but come Writer version 3 or a version of the Novell plugin that works with mainstream OpenOffice.org then we'll do the same thing with .docx. As long as the style names are preserved we can repair your document. The usual result for Word users is a pretty good XHTML output but some loss of WYSIWYG for the print view of your document.

ICE is still not that easy for people to try out casually, but we're working on that we have the server-based version running in beta now and that's proving much easier for people to get in to.

Some comments on the Microsoft issue

And a note to Peter's correspondent Glyn Moody who writes:

> Word? OOXML??? Come on, Peter, you want open formats and you're willing to accept one of the most botched "standards" around, knocked up for purely political

> reasons, that includes gobs of proprietary elements and is probably impossible for anyone other than Microsoft to implement? *That's* open? I don't think so….
>
> XHTML by all means, and if you want a document format the clear choice is ODF - a tight and widely-implemented standard. Anything but OOXML.
>
> http://opendotdotdot.blogspot.com/2008/05/ooxml-for-petes-sake-no.html

ODT is **not** widely implemented in any meaningful sense. I'd love to be proved wrong, but see my take on this. Really. I'd love to see another word processor that can edit .odt files interoperable with Writer that is **not based on the same code like most of the contenders**.

Take the example of lists that I raised earlier. To implement an ODF word processor you would have to implement something that dealt with hierarchical lists as specified in the standard, and work out how to cope will all the mess that results from a system where a paragraph has can have a paragraph style as well as a list style **and** item-level within the list. And there's a very loose connection between list and paragraph styles, and a very strange mechanism for embedding extra paragraphs in a list item. I have never seen a real-life .odt file that uses the list model properly.

I don't think 'anything but OOXML' is the goal, I prefer **stick to the interoperable subset of OOXML and ODF**'.

If we stick to the interoperable features that work with current software then back-end systems like an ICE server can transform Word documents into ODT and XHTML for archiving, without users having to worry about it. Even though as far as I can tell there's only one code base that really works with ODT at least it is open source.

But at the moment we **do** tell our ICE power-users that they're better off with Writer than Word even if they have to migrate from EndNote to Zotero.

And Microsoft itself seems happy to drive users to OpenOffice.org. I can pick which word processor and operating system I want to use, and I don't like going near Word 2007 for Windows with its stupid ribbon thing. Too hard to find anything; Writer is much more like Word now than Word is and it's easy to download and install.

And on the Mac, while Word 2008 doesn't have the ribbon it also misses out on VBScript, meaning that we can't support it in our ICE system unless we rewrite our toolbar in AppleScript or whatever it's called.

It's unlikely there's a business case for that. I imagine that the same will be true in lots of organizations where templates and tie-ins to corporate systems will no longer be supported on the Mac – but it is possible to port macros to OpenOffice.org and work cross platform. For example, the current ICE toolbar code is actually a single Basic code-base that works on both Word and Writer.

And a final comment about the Microsoft sponsored work that PMR's group are doing. I hope the tools they develop are going to work with OpenOffice.org and ODF as well, at least to the point that we can process the data in ICE.