More on Microsoft Collaboration and word processor interop

It's not like I'm ever going to get the last word on this unless I turn off the comments but I would like to wrap up the discussion that's gone on here about word processor interoperablity and format support in various word processors, which was sparked by Glyn Moody's complaint that the Science Commons people were working with Microsoft and perpetuating their monopoly on word processing and move it onto specifics about what we can do for scholarly communications.

As Rick Jelliffe says I get good quality comments here. (Rick has taken the time to collate some of my more direct statements together into a page which he called Master Blaster. Did I say all that? Lucky the only vendor responding much is Microsoft and they are used to much worse than I dish out.)

I want to just follow up on a couple of those comments an clearly pose the questions I have been asking all along.

Rick points out that I was wrong to suggest that ODF doesn't have a custom XML feature. He's right, and it's much worse than the OOXML one, because OpenOffice.org (which is really the only thing that supports most of the standard) doesn't support it. But I have never seen any kind of proposal that anyone would *use* such a silly feature so I have been ignoring it.

Rick also says:

> The aspect of this is that it seems a bit strange to criticize a general feature that is specifically designed to allow value-added documents to participate in custom toolchains that they impair interoperability. It is like saying that the pen impedes interoperability because it may be used to write in Mongolian, which few people can read.
>
> Interoperability is not the be-all and end-all of qualities for a document format. Customizability and extensibility are useful things, in their place.
>
> So it seems to me that the question should be whether the customXML features are useful or appropriate or optimal for a certain case (or class of cases), rather than blanket statements. I think I am allowed to have different use cases, and therefore features, to you, aren't I?

Yes exactly. I thought that I was making it clear that I am talking about use cases like the Ontology plugin work, i*n the context of academia*. If that was not clear I will state it now, I agree with Rick, and it is none of my business what you do with Microsoft technology in the privacy of your own enterprises. I just go on about interoperability because it's important to me and my colleagues for various reasons not the least of which is that we work in a heterogeneous software environment. I think that was at the root of Glyn Moody's original rant, too.

Ian Easson patiently explained what happened when he tried out the Word 2007 ontology plugin which sparked this whole thing. I won't quote the whole thing here but the essence is that if you use the Add-in, save your document edit in Word 2003, as long as it has been retrofitted with the `docx` reader then your information may or may not survive editing.

Specifically using Ian's example if he changes Potter syndrome (linked to an ontology) to *blah blah* then the custom XML goes away. But why should it it? What if *blah blah* were a synonym of Potter syndrome?

Ian has confirmed what I said in the first place which is that the custom XML may or may not survive the round trip. Anyway, if you save to `.doc` to share with other users be they Word 2003 or Writer users the information is lost.

And we have this from Microsoft's Doug Mahugh:

> Peter, I'm with Ian on the fact that IS29500's custom XML support is every bit as well-defined as OLE embedding is. I agree with your point that it's messy for two different editing applications to round-trip documents with custom markup in them, especially if the document is edited in both apps, but that's not the typical use case I've seen in custom XML scenarios. **The tagging of content is usually done immediately before some type of automated processing by a non-desktop app, so the issue of what an editor should do with that markup doesn't come up very often.**

See the last sentence there, which I emphasised? Here's Doug talking about the kind of situation where this kind of XML work makes sense – but I think this is at odds with a plugin which might be used by a group of collaborators working on a paper as I would expect to get with the plugins coming out of Microsoft Research.

So I'm going to ask the same question I asked before. **Why couldn't the ontology add-in (which is a good idea, don't get me wrong) store data using the simplest most robust method possible; using a link?**

The Human Diseases ontology cited by Ian doesn't seem to be web-referenceable but I found one that is. Geonames.org provides a linked-data endpoint for Toowoomba. See what I did there? I linked to it. And look how you can get RDF. That will survive round tripping between Word and OpenOffice.org, saving as HTML, saving as .doc, sending to a publisher, and it's perfectly usable without an add-in. Could someone explain to me why you would need to use custom XML here given the simplicity, robustness and interoperability of the alternative?

And I'll say one more time that this is not incompatible with the idea of having an in-application Add-in to help you find locations or diseases (Toowoomba is considered by some to be both) and link them. If Word has this, then Microsoft might sell a few copies or prevent a few researchers from moving to OpenOffice.org by providing a better tool, not by locking them in no matter whose 'fault' that is.