Another look at the Article Authoring Add-in for Microsoft Office Word 2007

The "Article Authoring Add-in for Microsoft Office Word 2007" (AAAiMOW[1]) has been turned loose as a release candidate. I looked at an earlier version of this a while a ago.

The name of the thing doesn't let on that it is targeting just one version of what an article looks like, in the form of the NLM schema I'm not sure if that reflects confidence that the NLM schema is generic enough to cope with all articles or anticipates a future version which can support multiple formats.

I had a lot of questions in my previous post – most of which I think are not yet answered, although Pablo Fernicola did drop by my blog and shed light on some of the issues.

This time, with a fresh virtual installation of Windows XP running under VirtualBox on OS X the plugin worked a bit better for me so I could see it in full flight. I still have some serious concerns about this add-in thing and what it might mean for organizations.

I was going to make a few quick comments about usability, preservation and lock-in but htis post kept growing, I emailed Jon Udell for his take and did a few tests, and it's ended up well on the way to 2000 words.

Usability?

I can't find any reports of how this plugin works in real life. Has anyone tried it? Are you all under NDAs?

I'm concerned about the way that you can add NLM structural elements all over the place, and nested inside each other in bizarre ways, but then you can't save to the new proprietary .nlmx format because of validation errors.

It would be pretty easy to show how you can create invalid structures using this plugin but I don't really think that's a useful stunt to pull – what I want to see is what **real** problems, or lack of them people have with the structural stuff.

Me, I found it a bit weird but as I said I didn't try to write an article with it.

There's one interface device that I really like. Each 'section' element gets a little handle above it so you can drag the whole thing around:

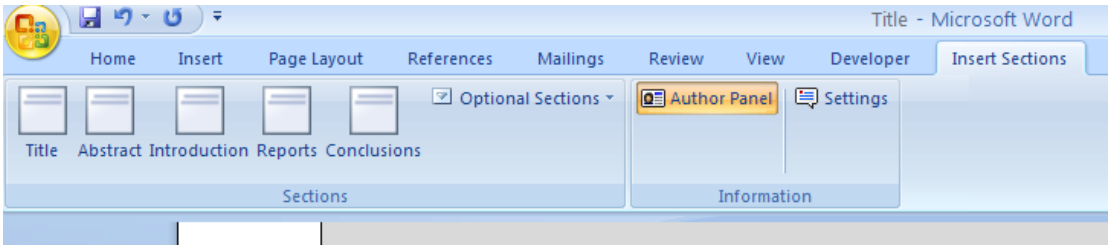
It would be really nice if this applied to the document outline as well as part of the normal Word interface not just to the special embedded XML sections. I could just style a bit of content as `Heading 2`, which is part of the document outline structure, and be able to drag around the whole of that implied section. Word already does something very like this if you

1 Sounds like a noise emanating from a petulant feline.

use the Outline view. Of course, dragging sections in an NLM document doesn't make sense as they're supposed to be in a particular order, but I don't imagine most people would drag the high-level sections. (There's some kind of complex process for dealing with section ordering or editors, I think).

I'm not sure if I get why the embedded XML is any better than just recognizing that the text 'Abstract' in `Heading 2` style is a the start of the Abstract section. Or you could define sub-classes of heading if you really wanted to such as `Heading 2 - Abstract`.

You could still have a toolbar like this so that people can drop in sections where they want them:



Lock in

This add-in represents a new opportunity for Microsoft to lock users in to Word, having just moved on from the proprietary .doc format. This is not just a matter of trying to sell more copies of Microsoft Office it's about encouraging users to create documents that only work with a particular version of Office.

We have just been through a great long debate about standardizing word processing formats. Microsoft got their way and had their OOXML format accepted as an ISO Standard (ISO/IEC DIS 29500). The benefit is supposed to be that when you write a word processing document it can be managed and edited in more than one application but I have always been very dubious about how this fits with the way you can embed arbitrary foreign XML in Word documents. By contrast the Open Document Format approach is an RDF based extension mechanism which seems a lot cleaner.

I tried out some simple interop with an AAAiMOW document.

1. Word 2008 on OS X can open it, and you can edit the document at least a little bit, apparently without breaking it, but anything you add doesn't have the magic embedded XML. It round tripped without error but I assume you could break some of the XML.
2. Neooffice Writer on the Mac can open the .docx file and you can edit, but if you save it and re-open in Word then you get an error . The good news was that Word 2007 was apparently able to rescue the content but the bad news was that embedded XML went AWOL.

At the moment I would not have any confidence that anything except Word 2007 can deal with documents created with the add-in, which is as advertised. Of course, if that's what your team of scientists is using then no problem, provided you think about how you will preserve the outputs (see below).

That quick interop test was using the new .docx format which is **not** the same as ISO/IEC DIS 29500, which won't be available as a Word format until the next version.

One of the features of the AAAiMOW is a new file format. Yes. A new non-standard file format which is a misbegotten mashup of OOXML and NLM. I'm not sure how this is

different from the way the content is embedded in the .docx file. From the readme file:

> Both the article contents and metadata authored through the add-in are stored using XML, as part of a single file, using the Open XML format for the content and the NLM tagset for the metadata. Content which does not have an equivalent in Word, or extends existing Word elements, is stored as custom XML elements within the Open XML data stream. When a file is saved in the NLM format, the resulting XML file is stored within a nlmx file, using the same Open Packaging Conventions used by docx files, providing a single file which can package all related content (such as images) and supports extensibility.

Meanwhile, the next service pack for Word 2007 will add support for the OpenDocument Format (ODF) as a native file format. I'm assuming the plugin won't work with ODF. (Pablo, am I wrong?)

There's some very alarming use of the passive voice in the documentation too, a classic computer industry trick. Say it **can** be done without mentioning who's going to do it and how much it's going to cost.

> Based on the use of Open Packaging Conventions, the Open XML format, and the NLM tagset, tools can be built to access any part of the file, content or metadata, and extract, validate, or add information to the file, as part of the publishing workflow.

"Can be built?" Please. We have one format mixed in to another format using a user interface that is only accessible from an expensive proprietary application. I'm sure I could write a script to pull the NLM bits out of the Open XML but for each new kind of embedded XML I would have to rewrite my code and test that it works with the user interface code that has been added to Word – in this case it involves dealing with some special attributes to re-order sections (I think) – doesn't look easy or pleasant to me.

And it is worth remembering that this plugin is not accessible to the majority even of Windows users. For example here at USQ Word 2007 has not been rolled out yet[2]. And the plugin is not available at all on platforms other than Windows. That's not what I hoped the new standards-wielding Microsoft was on about.

Preservation?

There are going to be serious issues with preservation. What are archivists supposed to do with bastard mashed-up formats like this which depend on a particular package to make sense of them?

It is true that for documents that make it to publication in the NLM XML format this should not be an issue: the resulting XML should be perfect for archiving. But I can see that a lot of things that are of value might not make it through to XML. What about archived author's manuscripts which are one of the backbones of Open Access? What about the original editable files for images drawn using Microsoft Office tools, which are embedded in the source file?

---

2 We're bracing for the onslaught on the help desk as hundreds of users have to re-learn commands they've been using for their whole working lives. It seems to us on the ICE team that this is a perfect time to introduce our users to the copy of OpenOffice.org which will be on their computer.

Think about what would happen if this approach became common for different XML formats – there could be a proliferation of non-standard polluted Word document to deal with in repositories.

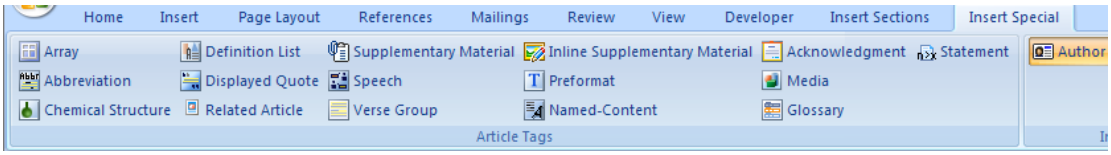
This add-in represents the Microsoft business model in action. See Brian Jones' response to my probing on the issue of how this bastard mashup stuff is supposed to work. I quoted this last time, but it's worth reminding ourselves that this is what Microsoft is about, never mind the standards:

> There is a huge market that exists today for custom Office solutions. People customize the Office applications in all kinds of ways to try to get more out of their documents. By adding the support for custom defined schemas, we made it much easier to build semi-structured solutions on top of Word. Rather than rely on hacks with styles or bookmarks, folks could create a simple schema and add some XML tags into their existing document solutions.
>
> http://blogs.msdn.com/brian_jones/archive/2005/07/08/436973.aspx#452483

Brian Jones calls styles to carry semantics is 'a hack' and yet embedding foreign XML in a Word document and hand-crafting a user interface to deal with the resulting mishmash of tags is somehow not a hack? I agree that styles and bookmarks (and tables – we use them a lot) are somewhat limited carriers for microformats but the XML embedding thing has always looked like a trap to me – too expensive to set up and maintain and too much embedded in the Windows world. As I mentioned above, I think the new extension mechanism for ODF may be a better compromise – maybe we'll see that in the ODF support in the service pack release in 2009.

An alternative

There's an alternative approach which is to use features that are common to word processors in general and which are expressed in the underlying file formats directly, which I wrote about in my last post. There would be some interseting challenges in finding interoperable ways to embed all the 'special' items that are allowed – some of these are already supported in our ICE templates but not quite with the same structural rigour as in this the add-in.



Chris Rusbridge from the lamented in the comments of my last post that we don't do NLM export from ICE – but I reckon we could produce NLM XML from ICE documents with no more subsequent work required of editors than you would get using the AAAiMOW (I'm guessing – we have no data about how well it works and I have yet to work through the section of the documentation for editors).

The readme tells us that styles don't work (emphasis mine):

> Custom XML elements are used to represent other abstractions that exist in the NLM

> tagset, but that are not found in Word, and to do so in a manner that can be presented to the author for editing in a robust way **(unlike the use of custom styles, which was one of the ways to try to solve this problem in earlier versions of Word, and was not very reliable).**

I have no doubt that there are lots of terrible style based systems out there, but we have worked hard on making styles usable, interoperable and easy to apply and providing robust rapid-feedback document conversion.

(Maybe both of us are wrong – MJ Suhonos at PKP thinks that you can create XML without using either styles or embedded XML by using document formatting to infer structure.)

Would anyone care to fund a small project to see if we can use ICE to produce similar overall results (in terms of overall ROI) to the AAAiMOW but in a cross-platform solution? Microsoft? Anyone in the UK with access to JISC funds? A publisher?