Embedding XML in word processing documents (if you really must)

Rick Jelliffe has posted a comparison of how foreign XML can be embed in OOXML (that's the XML format for Microsoft Office) and ODF (the Open Document Format).

Rick starts with:

> First the caveat: Word and OpenOffice are not general-purpose XML editors.

Right. That means that if you do decide that there's a case for embedding extra XML in OOXML or ODF then you are going to have to supply add-ons to the applications in question to edit it. So what does this mean for the two formats? (As usual I'll just talk about the word processing format here and ignore spreadsheets and the rest.)

For OOXML, you would have to create a Word Addin such as the one I've looked at here before. There could be business case for that, but you'd have to accept that your documents were only going to be editable in Word 2007+. I gather from recent posts that Rick does some work on projects where this does make good business sense.

For OpenOffice.org you're out of luck. Rick's tests show that OpenOffice.org strips out foreign markup. It's unclear whether this is conformant behaviour or not:

> But the bottom line for foreign elements as wrappers in ODF and OOXML is that ODF allows them to be stripped out while OOXML doesn't allow that; neither of course require that any particular application understands them. The bottom line for OpenOffice and Office seems to be that OpenOffice strips them (dangerously, but perhaps allowed because of bad drafting of that part of the ODF standard) while Office 2007 does allow them.

As I've covered here many times ODF interoperability between applications is basically non-existent except between Microsoft Office and OpenOffice.org and its derivatives where some things work quite well. Bottom line is, ODF doesn't have any formal notion of what's conformant – it's up to application developers to implement the bits they feel like implementing.

> The OpenDocument specification does not specify which elements and attributes conforming application must, should, or may support. The intention behind this is to ensure that the OpenDocument specification can be used by as many implementations as possible, even if these applications do not support some or many of the elements and attributes defined in this specification. Viewer applications for instance may not support all editing relates elements and attributes (like change tracking), other application may support only the content related elements and attributes, but none of the style related ones.
>
> http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.1-os.pdf

I think for most uses a much better bet is to use microformats which leverage the built in features of the formats. These not only work in the aforementioned major applications for OOXML or ODF, in many cases they interchange between the formats quite nicely as well.

What's a word processing microformat? One example would be using a one-cell borderless table with a paragraph in it of style 'h-warning' to indicate a bit of content that's a warning, to use Rick's example. Ok, so using a table is inelegant, but it works in both Word and OpenOffice.org writer and will survive round tripping between .doc and .odt and .docx. You could use a frame, which is a more semantically neutral element and sacrifice some interop, or you could use styles only, which is a bit harder for users to manage and more error prone. Actually, Rick gives an example of a styles-based microformat approach.

We use this kind of technique to do things like generate slide shows from text embedded in documents, and we're developing methods for embedding metadata in documents using styles.