ICE and word processor / HTML interop, the ugly, uglier, ugliest

[Update: finished an unfinished sentence, added a link]

We have started having post-Friday-morning-tea information sessions in the ADIF technical team. Ron Ward kicked the series off with a great overview of Javascript which is both the world's most widely deployed and worst misunderstood language and Greg Pendlebury introduced us to some of the wonders of spelling suggestions in Apache Solr indexes (he works nights on the VuFind project).

My contribution is this presentation on ICE, below, looking at how it converts word processing documents into HTML, and some of the complexities that the ICE code has to deal with. These complexities are technical, horribly so in some cases. Many of the complexities are present not for good technical reasons but because of business and politics; competing vendors and competing standards have left us a really nasty mess to deal with.

My talk started with some history. When I first got my hands on a WYSIWYG word processor, around 1988 or so, the first thing I did was go through all the menus and work out what everything was *for*. For my honours thesis in 1990 I naturally used styles for the headings, so I could compile a table of context and the automatic numbering feature for figures, EndNote for references and so on. I was far too lazy to do any of that by hand.

Twenty years later, those features are still in the word processors we all use (although Google Docs is a disappointment), and we now have Zotero for reference management, to which we in ADFI made a modest contribution. But I still meet senior academics who have no idea how to use styles to make a table of contents, don't use reference management systems and consequently are unable to help their students to work efficiently.

There a couple of interesting questions, which I'm not going to answer here, but which are important to the research planning process at ADFI:

1. What are the word processors of the future going to look like? It's clear that we will still need to 'process words' and some of them will still be strung together in long sequences, like books and theses, but other more dynamic genres are emerging. I've written here about Google Wave (which seems to have receded leaving behind only plastic bottles and dead fish) and various other new approaches to editing. I don't know what's coming but I am sure it will be collaborative.
2. How can we help people to use these programs for processing words?
3. Shouldn't our training programmes for academics be more like “Word processing techniques for academic writing, using Word 2007” than the current “Word 2007 basics”? How would we go about making that change?
4. More generally, how can we encourage the kind of laziness that prompts some of us to discover the bleeding obvious things that are under out noses, by doing things like RTFM? And how do we create a culture where that know-how can spread?
5. And what do we do when what we need is not there right under our noses? For example we need the ability to produce Scholarly HTML from our documents but current tools just don't do that. How can we get the academic community and software vendors, including us in the open source world, in-sync to fill in the gaps?

   Next week I have a call with Microsoft Research, and I'll be talking about getting Word to be able to produce web-ready, journal ready, semantically rich documents, with useful metadata and seamless links to data and interop with other programs like OpenOffice.org. Note, I'm not looking at this from the point of view of a 'standards war'; I'm trying to get thing **done**.

When I ran through the slides below today we came back to these questions. We didn't get any answers in half an hour, but my summary of what we should be doing is:

1. Finally get around to setting up one or more cohorts of postgrad students with an end-to end system for managing their thesis starting from day one, and including a facilty for annotation/comment by supervisor, cohort and finally examiners.
2. Work out how to educate (not train) the trainers about some of the principles behind academic use of tools.
3. Work with our Learning and Teaching Support Unit to start changing the corporate training culture away from teaching specific tools with a generic focus to empowering academic users to do academic things.

What and why?

ICE is, at it's core a way of turning word processing documents into **proper** web pages.

Word processors are:

1. **Ubiquitous in academia**. Still. Even with the advent of Moodle's built in HTML editor and Google Wave.
2. **Much maligned**, particularly by computer scientists, engineers and expert tinkerers who have much better ways of making web pages and printed documents from the same source (until you ask them to show you).
3. **Full of useful features** for long documents, including:
   i. Drawing tools
   ii. Data visualisation tools (charting)
   iii. Reference and bibliography management (usually best with a plugin)
   iv. Change-tracking (and some have synchronous collaboration)
   v. HTML export...

HTML export?

OK, so the HTML export in word processors is universally crappy.

This is one of the great mysteries of the universe.

Actually it's not – Microsoft made a concerted attempt in the late nineties to make their office suite part of the web, and to add their own extensions to HTML By 2000 they were the only major player left in the word processing world and I guess they thought they had a shot at owning the web. It has taken nearly 10 years for us to get back to something like a multiple-horse race. Ironically the widely ridiculed Word 2000 save as web format was actually very easy to process into XML and to use to produce good quality web pages; it took years to generate the current OOXML approach which is much less useful. (I wrote about his for XML.com).

It is a mystery why Google Docs is so bad, though. I had corresponded with one of the people from Google Docs about making it better, but he lost interest.

In 1995...

The technical writing team at TAFE NSW information systems division decided to put our technical bulletins on the web. (We started this work before HTML even had tables, BTW).

- We used word processing styles, because that was (and still is) what sensible, rational people do.
- I rolled my own process based on: RTF via Rainbow maker to SGML, to ESIS via nsgmls to HTML via Perl + SGMLSpm.

  The current ICE conversion system is quite similar – uses an event driven XML (SAX process and a scripting language, it's just that word processors now produce their own XML.

  In between I have used Omnimark (which was just plain weird), DSSSL (which was Scheme, which is a kind of Lisp), XSLT (dumbed-down DSSSL with XML syntax) which we dumped in ICE for performance/maintainability reasons, and the free-as-in-beer ACE language that came with TeraText (from RMIT).

In those days:

- Image handling was tricky – it was hard to convert embedded objects to web formats.
- I thought that the next version of Word would ship with syle-to-web mapping and I could stop working on this distraction*.

Styles

In 2010, we're writing software at ADFI that uses very similar styles to the ones used in 1995 at TAFE, which I refined for use at Standards Australia then NextEd before starting the ICE project at USQ. The ICE style-set is defined in my 2006 paper.

I promised you ugliness

Styles are sensible and useful. But:

- **There are no standards**. The only thing close to a standard is the use of Title, Heading 1 … Heading n. How do you handle a mixture of numbered headings and non-numbered headings. What about chapter numbering?
- **There is no standard even for a single vendor for an ordinary paragraph**: Normal, Default Default Text etc.
- Even individual **vendors don't stick to or ship a set of styles** for things like lists or quotes in their sample templates.
- Word and Open Office and Google Docs **all** produce **different** rubbish when asked to make web pages and make almost no use of styles in doing so.

* I was just taking a short break from computational linguistics which turned out to be easier than making web pages, in retrospect.

And worse...

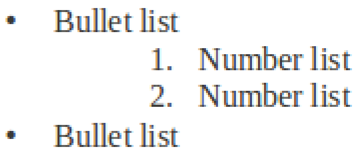
Like here are some of the favourites from the team:

- There are some circumstances under which Word 2007 does not export graphics when saving as HTML (happened to Ron – not sure if it is reproducible).
- Page breaks and section breaks probably cost society a lot more than terrorism and smoking combined.
- We have not forgotten or forgiven the paper-clip; have they apologised yet?
- Word used to (and still does) get into states where things fly around randomly.

Look, if O'Reilly publishes a book entitled “<name-of-your-product> Annoyances” you have a problem. If nobody bothers, as with OpenOffice,org then you probably have a bigger problem.

And worse...

Here's a list I created using the bullets and numbering toolbar in Writer:



- Bullet list
    1. Number list
    2. Number list
- Bullet list

ODF has hierarchical list structures (one of the worst decisions ever). Now, you'd think that if we were going to have hierarchical list structures this might be, you know hierarchical, like one list embedded in another. Instead we get this :

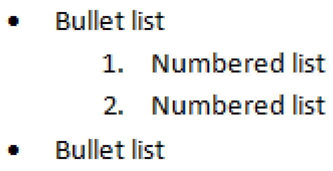
```
<text:list xml:id="list1916064342" text:style-name="L1">

<text:list-item><text:p text:style-name="P1">Bullet list
</text:p></text:list-item>

</text:list>

<text:list xml:id="list1908078027" text:style-name="L2">

<text:list-item>

<text:list>

<text:list-item>

<text:p text:style-name="P2">Number list </text:p></text:list-item>

<text:list-item>

<text:p text:style-name="P2">Number list</text:p></text:list-
item></text:list>

</text:list-item></text:list>

<text:list xml:id="list322315948" text:style-name="L3"><text:list-
item><text:p text:style-name="P3">Bullet list</text:p></text:list-
item></text:list>
```

Bizarre. Truly. The 'main' list is three lists. That 'middle' numbered list is a single element list with no text with a two element list embedded in it.

I promise all I did was push the buttons on the toolbar. Pushing them in a different order gets you different craziness.

Word's OOXML has a more rational approach

OOXML is flat - the list formatting is implied. Each paragraph is a paragraph and the fact that they belong in a list structure is indicated with attributes. Given that word processors have a paragraph based interface this is rational.



- Bullet list
    1. Numbered list
    2. Numbered list
- Bullet list

This kind of markup scares some people, but it's very efficient:

```
<w:p w:rsidR="009B3D58" w:rsidRDefault="002D3C0B" w:rsidP="002D3C0B"><w:pPr><w:pStyle w:val="ListParagraph"/><w:numPr><w:ilvl w:val="0"/><w:numId w:val="1"/></w:numPr></w:pPr><w:r><w:t>Bullet list</w:t></w:r></w:p>
<w:p w:rsidR="002D3C0B" w:rsidRDefault="002D3C0B" w:rsidP="002D3C0B"><w:pPr><w:pStyle w:val="ListParagraph"/><w:numPr><w:ilvl w:val="1"/><w:numId w:val="2"/></w:numPr></w:pPr><w:r><w:t>Numbered list</w:t></w:r></w:p>
<w:p w:rsidR="002D3C0B" w:rsidRDefault="002D3C0B" w:rsidP="002D3C0B"><w:pPr><w:pStyle w:val="ListParagraph"/><w:numPr><w:ilvl w:val="1"/><w:numId w:val="2"/></w:numPr></w:pPr><w:r><w:t>Numbered list</w:t></w:r></w:p>
<w:p w:rsidR="002D3C0B" w:rsidRDefault="002D3C0B" w:rsidP="002D3C0B"><w:pPr><w:pStyle w:val="ListParagraph"/><w:numPr><w:ilvl w:val="0"/><w:numId w:val="1"/></w:numPr></w:pPr><w:r><w:t>Bullet list</w:t></w:r></w:p
```

But not that rational

So – Word's better, right? Not really. There are several ways to make lists in Word:

- **Ad hoc lists** using the toolbar buttons which is likely the only thing casual users will discover in Word 2007 (there are no list-type styles showing in any of the ribbon galleries).
- Using **anonymous multi-level list structures** – you can define these via the styles part of the ribbon, but not name them.
- Using **named list outlines** which you get to from a DIFFERENT place on the ribbon (not the styles part) but which point back to styles.
- Using **list styles** – which were introduced in Word 2003 – but you know, I can't find them in Word 2007 although apparently they're there somewhere.

Here's the attempted explanation on the Word blog. Read that and I'm sure you'll feel much better about the ribbon in Word 2007. It was **for your own good**.

So in ICE we try to do the right thing

We try to:

- Produce sensible HTML using whatever cues the document contains:
  - Styles, obviously make that very clear, but even with styles if someone has a third level list style following an ordinary paragraph ICE will not try to create some kind nested list structure or add a big margin it will produce sensible structured HTML.
  - Lately, we have been adding code to try to interpret direct formatting but this approach is never going to be as robust as using styles and it is very hard when using, say Word and Writer on the same document.
- Provide a toolbar which tries to generate sensible structure – eg if you hit the block-quote button in a paragraph under bullet list it will indent the quote under the list.

This post was written in OpenOffice.org, using templates and tools provided by the Integrated Content Environment project and published to WordPress using The Fascinator.