Silk purses from sow’s ears

Lou Burnard
April 2003

Four favoured foibles

“Plain vanilla ascii” looks like it came from a typewriter

All My Own Work Err, what’s that {>!22345 _} code for?

Word Processor output Looks wonderful...

. . . if you have the right version of word

HTML Looks wonderful . . .

. . . if your browser and mine agree

The format nightmare

- These formats are BAD
  - They are not portable
  - They focus on the appearance of text, not its meaning
  - Analysis software is not the same as display software
- But they are ubiquitous
  - So what tools can we use to bring texts using them back to the paths of righteousness?

“Plain Vanilla ASCII”

- Search and replace techniques will usually capture
  - Paragraph structure (blank lines)
  - Headings (lines in caps)
  - (sometimes) emphasis (strings between _ or *)
- Watch out for
  - Markup characters in the text
  - Metadata information
- Use:
  - Your favourite editor
  - Perl
  - (once you have a wfd) xslt transforms

“Plain Vanilla ASCII” : case study

The Xara tutorial describes how we convert a set of plain ASCII files to XML. Here is a sample file; here is the driver file which embeds 20 such files into an XML structure; and here is the complete XML file generated by running this stylesheet against the driver.

“All my own work” markup

- Same principles apply
- But extra vigilance is needed for pseudo-XML
- and the documentation may be hard to find
- Probably best treated with general purpose programming tool such as perl, or hard slog with an editor

Escaping from Word

- Several strategies are known to work
  - Save as HTML and then run **tidy**
  - Use Xmetal’s built in Journalist convertor
  - Use a special tool e.g. doc2xml
- GIGO applies:
  - if the Word document uses styles consistently. . .
  - otherwise you’re stuck
  - watch out for graphics, tables...

Using Xmetal to escape from Word

- Xmetal has a customisable interface using standard Windows scripting tools
- One application which comes with it will convert Word documents to valid XML, using a simple "journalist" DTD
- The tagging is (fairly) intelligent, recognising any structure that is available in the file
- But it’s slow. . .
- Once in XML, you can convert it. . .

Here is a two page word document; here is the output from the Xmetal conversion; here is an XSLT stylesheet which turns it into TEI XML; here is the output from the XSLT conversion.

HTML Tidy

- Takes any old HTML and gets rid of most known lunacy
- Generates XHTML
- Extracts styling information into CSS classes
- . . . your best friend in partnership with XSLT transformation

Here is a web page found in the wild; here is the same page run through tidy. This can be indexed with xara directly.

Detecting the structure

Functions can be (partially) deduced from the formatting:

- a para of class XX is probably a <div><head>. . . </head>
- but where does the </div> go?
- speaker turns, stage directions, etc. *may* be consistently marked

But expect there to be exceptions...