Lecture 7 – Review of Document Management & Introduction to Markup

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

Objectives

- Review Document Management
- Look at XML based markup languages
- Today's practical
  - Embedded objects, tables, graphics and templates in Word

Issue – Longevity

- Some documents exist for a very long time
  - Sometimes for archiving
  - Sometimes still being updated
- Other examples
  - Legal documents
  - Census data
- When choosing the medium for long lived documentation
  - The feature set of currently available word processors is not a major consideration
  - 18 month upgrade cycles are NOT welcome

Issue – Incremental Updates

- There can be considerable savings through incremental updates
- But incremental updates imply:
  - Page numbering by sections, not continuous
    - e.g. Page 2 – 14 – 3
  - Table of Contents and Index must follow likewise
  - There may be “knock-on” updates to following pages
    - Double sided publication is quite common
  - There may be a need for a “List of Effective Pages”
    - Each page has a version number
    - The LEP shows the version for each page

The Re-use Problem

- Consider a set of related documents
  - They may be for different variants of a “product”
  - They may have different target audiences
- There may be a high proportion of common content
  - Large amounts of re-used text
  - Graphics may also be re-used
- Common text and graphics should be edited ONCE
  - Edited in one place, re-used in others

Issue – Content Reuse

- Some documents are made up from “fragments”
  - Fragments of text (“boilerplate”)
  - Graphics (in the broadest sense)
- But some content is specific to the document
  - Introductory material
- Some elements are “generated”
  - Table of contents
  - Table of figures
  - List of Effective Pages
- How are changes and versions managed?

Other Issues

- Multiple Authors
  - Many large documents have more than one author
    - Simultaneously
    - Over the lifetime of the document
- Structured review and approval cycle
  - Documents may be subject to a specific review process
  - There may be a formal approval process
- Multiple output formats
  - Documents may be simultaneously published
    - On the web
    - On paper
    - On CD

Embedded Codes

- In the old days…
  - Everything was ASCII text, special formatting was indicated by “conventions” – special characters

    -
      ```
      This is how we do
      .B bold
      text in nroff, a unix text processing package
      ```

    -
      ```
      This is how we might achieve the same \emph{bold}
      in Tex, another text processor
      ```
  - WordPerfect (now Corel) was probably the last word processor to use this type of coding scheme
- Graphical “WYSIWYG” word processors make formatting explicit (visible)
  - Even if “codes” are used internally

Markup Languages

- Markup Languages developed from proprietary codes
  - SGML – Standard Generalises Markup Language
    - Used in large, complex documentation systems
    - Long history, original expectation was manual typing of codes, thus efforts to minimise typing effort
    - As a result *very* difficult to parse & read
  - XML – Extensible Markup Language
    - Actually a simplification of SGML
    - Much easier to parse and use
    - A little more verbose
    - Expectation is that computer will create the codes

XML Fundamentals

- An XML document has a header

  -
    ```
    <?xml version=“1.0”>
    ```
- It has one “top level” element, which contains all the content

  -
    ```
    <stuff>
    This is the content of the document
    </stuff>
    ```
- Other elements can be inserted in a hierarchy

  -
    ```
    <stuff>
    This is <more-stuff>inside the </more-stuff>
    document
    </stuff>
    ```
- Elements can have attributes

  -
    ```
    <stuff importance=“low”>
    ```

XML Rules

- Elements must be nested correctly
  - `<stuff><more-stuff></stuff></more-stuff>` = ✗
- Elements must be terminated
  - `<stuff>and nonsense` = ✗
- Unless they have no content
  - `<empty-element/>`
- Angle brackets (and some other characters) must be escaped
  - `2 &gt; 3, for sufficiently large values of 2`

Other Bits of XML

- Entities
  - Pre-defined content (a bit like macros, but more general)
- Processing instructions
  - Special instructions for specific purposes
- Comments
  - `<!-- This is a comment, I go anywhere content can go-->`
- And that's about it…

What’s So Good About That Then?

- We can define (and enforce!) our own hierarchy of elements
  - Using a DTD – Document Type Definition (old)
  - Using a Schema (new and much more powerful)
- It is a portable, text based format
  - Easy to store, transport, compress
- There are lots of tools to do things with XML
  - Parsers – read and create XML hierarchies by program
  - Editors – authoring XML documents
  - XSL – XML Stylesheet Language for device independent formatting
  - XSLT – To transform XML into something else

What Is It Used For Then?

- The format of choice for the complex, technical documents we discussed last week
- Device independent data transfer
  - Especially e-commerce
- New format for Web documents
  - XHTML
- Open standard for Office Documents
  - OpenOffice.org (originally StarOffice, now owned by Sun Microsystems)
  - For comparison, Microsoft used proprietary, unpublished binary formats

What Are The Implications?

- If anyone can read Office Documents
  - There is no “lock-in” to a particular vendor
  - Products should become interoperable
  - Innovative ways of producing office documents can arise
  - Platform independence becomes easier
  - Documents can be re-used in many different situations

Today's Practical

- Microsoft Word
  - Tables
  - Graphics
  - Embedded Objects
  - Templates
  - Styles
- Try the additional exercises, especially the equation
- **REMEMBER TO SIGN OFF ON THE REGISTRATION SHEET!**