

Comp 336/436 - Markup Languages

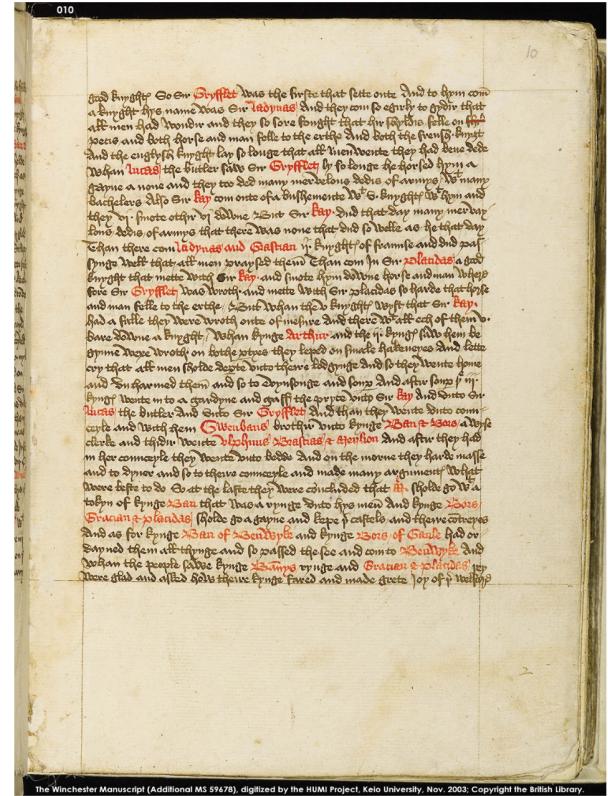
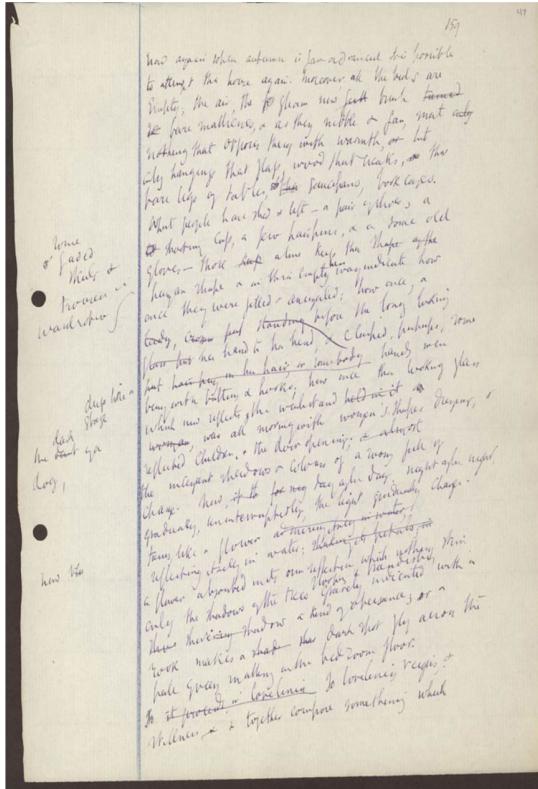
Fall Semester 2017 - Week 2

Dr Nick Hayward

Digitisation - textual considerations

- comparable concerns with music in textual digitisation
- *density of data* is still a concern with text
- should we encode texts with style, punctuation, notes...
- how much do we encode, in particular with manuscripts
- considerations of discipline, project, and personal preferences
- variation in requirements dependent upon manuscript
 - e.g. *modern vs manuscript*

Image - Modern vs Medieval



Modern and Medieval texts

■ Source

- Left = Page 159 of *To the Lighthouse* by Virginia Woolf
- Woolf Online
- Right = Page 10 of the Winchester manuscript of Thomas Malory's '*Morte Darthur*'
- Malory Project

Digitisation - shall we use markup?

- another option for digitisation of textual material
- advantages such as complete machine readability
- markup may take many different forms
 - *format (bold, italic, underline etc...)*
 - *logical structure (eg: sections, item lists, tables, ...)*
 - *context*
- all deal with the classification of components of a document

Digitisation - encoding

- encoding schemas capture structural and descriptive aspects of a text
- e.g. they might
 - *identify all dates and names*
 - *indicate whether something is a footnote, a chapter title, or a caption*
 - *precisely specify indentations, margins and poetry line breaks*
 - *or even designate the title of a speaker (eg: King, President)*

Digitisation - use some markup

Lou Burnard explains that markup makes

"*explicit (to a machine) what is implicit (to a person),*"

and adds

"*value by supplying multiple annotations*"

and facilitates

"*re-use of the same material in different formats, in different contexts and for different users.*"

Digitisation - fidelity

- attempt to recreate text with greater visual fidelity
 - also examine text in more complex ways...
- e.g.
 - search only notes, headings &c.
 - query for a given word, name or phrase...
 - manipulate, rearrange texts based upon given criteria
 - e.g. date, author, editor...
 - generate an index of all editorial notes by a given user
 - all books cited in a collection of papers...
 - ...

Markup - conforming to a standard

- exciting opportunities are possible when we all conform to a standard
- document markup predates the internet and computers
- separation of content from format
- tradition of markup with copy editors
 - *manually marking up manuscripts for typesetters*
 - *e.g. a particular chapter heading in a given font size and style*

Markup - historical context

- in 1967, an engineer named William Tunnicliffe suggested need for updates
 - *previous computerised system of codes for styles &c. too specific*
 - *codes were specific to a given program*
 - *codes should be replaced with a separation of content from format*
- in 1969, GML (Generalised Markup Language) was created at IBM
 - *used idea of generalised codes suggested by Tunnicliffe*
- GML later emerged as the international standard SGML
 - *SGML (Standardised Generalised Markup Langauge)*

Markup - SGML

- SGML did not provide predefined classifications or markup tags
- SGML was a 'meta-language'
 - *a grammar and vocabulary used to define any set of tags*
- different disciplines, industries &c. could define their own specialised languages
- DTD (document type definition) required
- new language would be based on meta-language of SGML
 - *or a pre-existing specialised language also based on SGML*
- SGML often perceived as complicated, time-consuming, expensive...
- SGML became known as,

Sounds good, maybe later

markup - text encoding

- practical value and importance to many disparate fields
 - *different domains, communities, organisations...*
- considerations of usage, depth, and scope
 - *lightweight markup options*
 - *prescribed schemas*
- often considered within a given context
 - e.g. *critical editions, indices, concordances...*
- can also be considered within broader context of new media
 - e.g. *multimedia, interactivity, networking...*

markup - typesetting

information formally distinct from the character sequence of the digital transcription of a text, which serves to identify logical or physical features or to control later processing....

- distinct from the text itself
 - serves to *identify logical or physical features*
 - *or to help with later processing*
- unfamiliar expressions or codes
- considered within broader context
 - *computer based typesetting and text processing*
- 1960s to 80s typesetting and text processing offers foundation

markup - early encoding

- encoding was initially specific to an application using
 - *codes for individual characters of the text*
 - *& codes for formatting commands*
- early computerised encoding of documents
 - *enter and store text in a file for future printing*
 - *encode individual characters of the text*
 - *using application specific codes*
 - *and codes for formatting commands*
- output of this process was formatted text

markup - descriptive in nature

- descriptive markup became seen as the fundamentally correct approach
- objective is to decouple a document's inherent structure
 - *decouple from specific processing, rendering, &c.*
 - *often described as semantic*
- descriptive said to identify and describe the parts of a document
 - *instead of providing specific processing instructions*
- procedural was a command or instruction invoking formatting
- logical v graphical

markup - benefits of descriptive markup - simplified composition

- with descriptive markup - intended formatting considerations
 - *make no claim on the attention of the author, compositor, transcriber...*
- with procedural markup - need to remember
 - *intended style conventions*
 - *specific commands required by formatting software for different effects...*
- with descriptive markup
 - *simply identify each text component as is*
 - *appropriate formatting may take place automatically*
- descriptive markup helps an author
 - *to work at an appropriate level of abstraction*
 - *TEI vs HNML*

markup - benefits of descriptive markup - structure-oriented editors

- descriptive markup supports *structure-oriented editors*
 - *know about patterns of components*
 - *components found in a given genre of document*
- editors may use this knowledge to assist the author or compositor
- e.g. autocomplete, suggestions, syntax highlighting, linting...
- many different editors for markup encoding support this feature
- schema specific support in some editors
 - e.g. *Oxygen with TEI...*

markup - benefits of descriptive markup - alternative document views

- output different views, rendering, formats for a given text
 - e.g. *an outline view of a text can be done automatically*
 - *use descriptive markup for chapters, sections, and headings*
- more detailed or specialised renderings and output
- use identified discipline specific components, e.g.
 - *equations*
 - *examples*
 - *cautions*
 - *lines spoken by a particular character*
 - ...

markup - benefits of descriptive markup - generic formatting

- procedural markup
 - *appearance of paragraphs &c. edited with formatting commands*
 - *precede each paragraph in a page &c.*
- descriptive markup
 - *a formatting rule is specified for a paragraph*
 - *separation of concerns, abstraction of formatting*
- helps control formatting
 - *easier to markup and maintain*
 - *less error prone markup...*
 - *helps ensure consistency in projects and domains*

markup - benefits of descriptive markup - extras

- descriptive markup helps support *textual apparatus*, e.g.
 - *creation of indices, appendices...*
 - *groups of lines, verses, quotes &c.*
- easily generate groupings of content
 - *tables, equations, plates, figures...*
- it offers device specific support
- descriptive markup may also be considered
 - *portable and interoperable*

markup - benefits of descriptive markup - retrieval & analysis

- offers support for information retrieval
 - *fielded content may be systematically accessed*
 - *request all equations, headings, verses...*
 - *combine fields in queries for greater depth...*
- offers support for analytical procedures
 - *content analysis, statistical studies...*
 - *e.g. analysis of spoken language and style in a play...*

HTML - brief intro

- HyperText Markup Language (HTML)
- HTML relies on keywords or element tags
- HTML can also use attributes within opening element tags
- keywords follow a rigidly defined syntax
- HTML creates web pages that web browsers can view
- an error or bug may cause the page to not render or simply render incorrectly
- to understand the current core of web page designing you need to know at least the basics of HTML

HTML - elements and attributes

Element syntax

- start with an opening element tag, and close with a closing tag
- content is everything between opening and closing element tags
- elements can contain empty content
- empty elements should be closed in the opening tag
- most elements permit attributes within the opening tag

Attribute

- attributes provide additional information to the parent element
- always added to the opening tag
- standard syntax of name/value pairs, `class="401"`
- standard attributes include
 - `class`
 - `id`
 - `style`
 - `title`

HTML - structure of HTML

- basic HTML tag defines the entire HTML document

```
<html>
  ...
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

HTML - within the <body> - basics

- to define the main body of the web page we use the `<body>` element
- headings can be created using variants of
 - `<h1>, <h2>.....<h6>`
- we can now add some simple text in a

element

- `<p>...</p>`
- add a line break
 - `
`
- add a horizontal line
 - `<hr />`
- comments can also be added through our HTML
 - `<!-- comment... -->`

References

- MDN - HTML Block-level vs Inline
 - https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements#Block-level_vs._inline
- MDN - HTML Global Attributes
 - https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes
- MDN - HTML Heading elements
 - https://developer.mozilla.org/en-US/docs/Web/HTML/Element/Heading_Elements