User-Centred Requirements for Document Structure in the Humanities

JEREMY GOW
University College London

GEORGE BUCHANAN
Swansea University

ANN BLANDFORD, CLAIRE WARWICK & JON RIMMER
University College London

Electronic document standards such as TEI and METS provide the means to digitally encode sophisticated hierarchical document structures. This naturally leads to digital libraries using document structure to support user navigation, which can be particularly useful for large or complex documents that are studied closely by their readers, e.g. those found in large historical and literary source texts. In response to this growing use, we present a set of user requirements for document structure in humanities digital libraries, illustrated with three digital collections from this domain. We discuss how well existing technologies satisfy these requirements and make recommendations for the use of document standards and the design of digital libraries.

Categories and Subject Descriptors: H.3.7 [**Information Storage & Retrieval**]: Digital Libraries—*system issues, user issues*; J.5 [**Computer Applications**]: Arts & Humanities—*literature*

General Terms: Design, Human Factors

Additional Key Words and Phrases: Humanities, digital libraries, document structure, interface design

1. INTRODUCTION

Hierarchical structure is ubiquitous in digital and non-digital documents, from the chapters of a book to the movements of a symphony. In the non-digital domain these structures are often indicated through media-specific conventions: typographic conventions draw attention to the chapters of a book or the stanzas of poem [Dori et al.

Authors' addresses: J. Gow, A. Blandford (j.gow, a.blandford@ucl.ac.uk), UCL Interaction Centre, UCL, Remax House, 31–32 Alfred Place, London WC1E 7DP, UK; G. Buchanan (g.buchanan@swansea.ac.uk), Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea SA2 8PP, UK; C. Warwick, J. Rimmer (c.warwick, jon.rimmer@ucl.ac.uk), School of Library, Archive and Information Studies, UCL, Gower St, London WC1E 6BT, UK; Project webpage: `http://www.uclic.ucl.ac.uk/projects/ucis/`

1997]. More subtle structures, such as a thematic section of a novel, may also be of interest to the reader. Indeed, a single document is always open to multiple structural interpretations, because several types of hierarchy may be present [Renear 1997] and there may be disagreements about how a particular type of hierarchy is applied [Butler et al. 2000].

An advantage of the digital domain is that multiple structures can, in principle, be explicitly represented and exploited for browsing, searching and referencing. Many document formats are capable of handling hierarchical structure, either directly or as metadata, e.g. XML, HTML, PDF. Markup standards have also developed in recent years. Both the Text Encoding Initiative (TEI) guidelines for markup of literary and linguistic texts [Sperberg-McQueen and Burnard 2002] and the Metadata Encoding & Transmission Standard (METS) [McDonough 2006] provide support for hierarchical document structure. As standards have developed the scope for using such structure within a digital library for browsing, searching and other tasks has grown.

Allowing the user to work with document structure in a digital library brings with it new interaction issues. Moreover, problems can stem not just from interface design, but also from the encoding of the original documents. Electronic encoding of documents from the original source material is often a separate activity from organising the electronic documents in a digital collection. The two activities may be performed by distinct organisations, perhaps years apart, and are often uncoordinated. So, considering the user's experience of document structure is more than just 'designing the interface right': we need to consider the hierarchical structure represented in the digitised documents, that represented in the digital library interface, and the differences between them. We can distinguish *structural* issues caused by a deficiency in — or a mismatch between — these document structures, versus other issues caused by the way in which the structures are presented and manipulated. For example, a lack of section titles in either representation is a document structure issue, whereas the font used to display section titles is not.

In this paper we examine document structure from the perspective of digital libraries in the humanities. Based on three case study collections built using Greenstone (section 3) we present an analysis of some general structural issues that affect library systems in this domain (sections 4 to 8), and provide a set of user-centred requirements for document structure aimed at document encoders and digital library designers working with humanities material (section 9). Some of these new requirements can be satisfied with judicious use of existing document standards and digital library technology, while others are aimed at developing technologies — in both cases we discuss current state-of-the-art approaches.

2. BACKGROUND

Our recent work has looked at supporting humanities scholars' use of digital libraries [Buchanan et al. 2005; Rimmer et al. 2006]. Document structure is of particular importance in the humanities, where users are often engaged in close analytical reading of texts which are themselves objects of study. As a result there is a greater need for precise navigation, search and referencing within documents than in e.g. the sciences, as well as an emphasis on preserving the structure of original

source material. A range of unconventional structures may be found, especially in historical material [Buchanan et al. 2007]. Internal document structure is often formalised as a hierarchy, such as the 'ordered hierarchy of content objects' (OHCO) model [Renear 1997; Biggs and Huitfeldt 1997].

Despite the rich variety of structures found in the world of documents, they fall outside of the traditional scope of library studies — though become more relevant when we consider libraries in the digital domain.

Various approaches to finding user issues and requirements for digital libraries have been previously explored. Protocol analysis was used by Blandford et al. [2001] to identify design issues with the use of multiple digital libraries, including the importance of *discriminability* of possibilities. In contrast, Theng et al. [1999] used a mixture of task completion rates and questionnaires to compare the use of three digital libraries by computer scientists. While a range of issues have been explored in the literature, there has been little work on the use of document structure in digital libraries.

One of the most common uses of document structure is the table of contents (ToC). Stelmaszewska and Blandford [2004] found that the ToC was used by readers in the evaluation of books found in physical libraries. Their use in the digital domain is widespread, including non-textual documents like multimedia presentations of lectures [Allen 1995]. Hunt et al. [1993] showed that multiple ToCs that displayed alternative conceptual structures were beneficial to users of online help systems.

Other studies have looked at interaction techniques for displaying digital ToCs: Chimera and Shneiderman [1994] explored the advantages of expandable ToC and multipane ToC interfaces over static presentations. Fisheye views can be used to selectively display ToCs in relation to the 'current section' by trading off importance and distance of other sections [Furnas 2006]. Graphical visualizations are also possible: Lin [1996] advocates 2D display of ToCs generated by a self-organizing feature map algorithm. These studies tell us that document structure is widespread and has diverse applications. However, insofar as they address usability, the main concern is the particular interaction technique at hand, rather than specifically structural issues with the underlying documents.

Document structure has also been exploited in search. The Superbook system improved search performance by combining search results with a ToC [Egan et al. 1989] . In an analogous domain, grouping together search results from similar web categories has been shown to be more effective than list presentations [Dumais et al. 2001].

2.1 Terminology

Given the diverse and diffuse nature of the literature on document structure, it does not tend to use a consistent and clear terminology. Following the OHCO model [Renear 1997] where the document is an XML-like hierarchy, we employ a typical tree terminology: the hierarchy is an acyclic graph of *nodes* with a single *root node* being the ancestor of all other nodes. In the literature nodes may be referred to as sections, subsections, divisions etc.

Each node contains an ordered list of objects, made up of *child nodes* and *content objects*. We define the *content* of a node as the ordered concatenation of the contents of its subobjects. The *size* of a node is the size of its content. Nodes with children

Collection	English Poetry	18th Century Fiction	The Bible in English
Abbreviation	EP	ECF	BIE
Period	16th to 18th C.	1700–1780	16th to 20th C.
Documents	4470	96	21
Lines of SGML*	17.5 million	1.8 million	2.4 million
Mean lines per document†	4000	19000	109000

Table I. Case study Greenstone collections. *Nearest hundred thousand. †Nearest thousand.

are *branch* nodes, those without are *leaf* nodes. Nodes also have one or more (possible empty) *node labels* that can be used for presentations purposes. Labels may be based on content, but are separate from it. The nature of the content objects depends on the type of document — for simplicity we will assume these are text objects. However, much of our discussion generalises to other structured media.

3. CASE STUDY COLLECTIONS

We illustrate our discussion of document structure issues in digital libraries in the humanities with three collections of primary source texts (see Table I), developed as part of our ongoing research into digital library use by humanities scholars. They were based on electronic materials kindly donated by ProQuest Information and Learning [1], a well-known international provider of information resources, consisting of source texts marked up in SGML and supplemented with images. Documents were encoded with rich and diverse hierarchical structures designed for a commercial humanities market. A considerable range of hierarchy sizes were present, so these collections were well-suited to our analysis of document structure issues.

Table I is an overview of the case study collections, giving the total size of the collection in document and in lines of SGML data, and the mean document size in lines of data. The three varied considerably in size, with *English Poetry* collection (EP) being by far the largest in terms of individual documents and lines of data. However, by line count EP had the smallest mean document size, with the average *18th Century Fiction* (ECF) document an order of magnitude larger and the average *Bible in English* (BIE) document an order larger again.

The collections were built using the Greenstone 2 digital library system [Witten et al. 2000], which has support for hierarchical document structure. We aimed to make the structure accessible to the user in the final collections in order to facilitate the identification of structural problems. Figure 1 shows a screenshot from the Eighteenth Century Fiction collection as implemented in Greenstone. Each document is displayed with a ToC (B–D in Figure 1), with the selected node (C) highlighted and the text (E) for that node (if any) shown. The menu displays the direct ancestors (B) of the current node, along with its children if it has any and its siblings (D) if it has not. Additional browsing functionality (A) allows users to switch views: between only the current node's text and all the text below the current selected node; to open the text of the current node in a separate window; or to toggle keyword highlighting.

[1] http://www.proquest.co.uk/

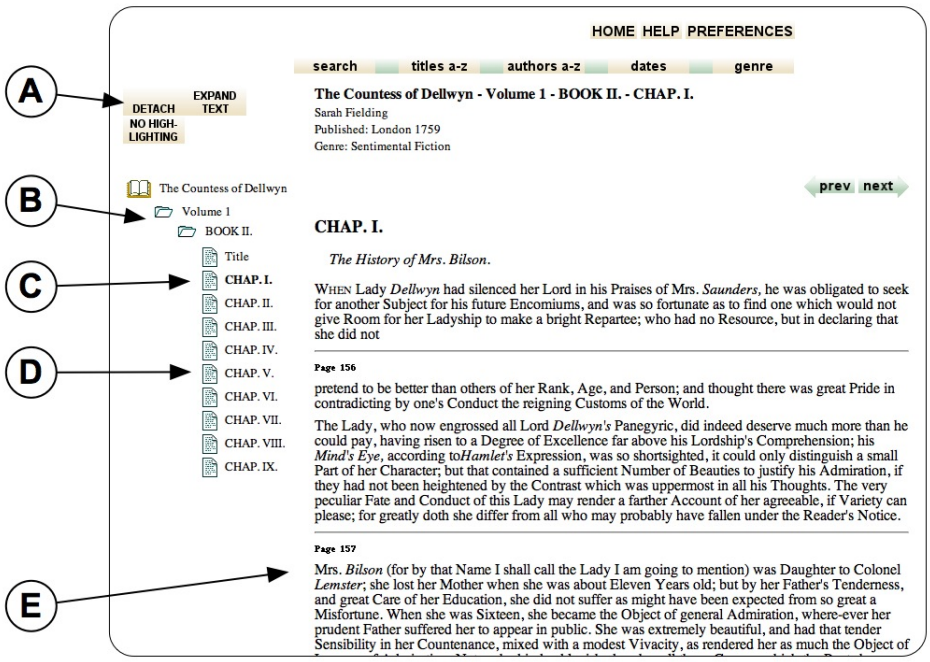

Fig. 1. Screenshot from the 18th Century Fiction (ECF) collection in Greenstone, showing the first chapter of *The Countess of Dellwyn* (1759) by Sarah Fielding. The ToC (B–D) shows the current node (C) highlighted, along with its ancestors (B) and siblings (D). Controls (A) allow the user to change details of the document view (see main text).

Collections may be indexed over nodes so that the user may choose to search over particular levels of document structure, e.g. the Bible collection can be searched over entire works or the text of individual books or verses.

4. METHODOLOGY

The user-centred requirements discussed below are the product of reflection on our development of innovative digital libraries interfaces for humanities scholars. They are not directly based on user studies, but rather on our experience of reusing large structured documents collections and our attempts to achieve a base-level of usability. This base-level is now allowing further user studies to be conducted. Hence the paper discusses some fundamental issues with delivering document structure, and we anticipate that further user studies will be required to refine and extend this work.

We use our case studies to motivate and illustrate general problems with the representation of document structure in the digital library interface. A possible objection is that Greenstone's handling of document structure is just one implementation of one approach — how can we learn more general lessons? However, we are not concerned here with user issues raised by the specific presentation and manipulation of structure in Greenstone. We are interested in *what* structure is

present, not *how* it is presented, and this is relevant to any system which exploits document structure. Indeed, many digital library systems, such as E-Prints [Hitchcock et al. 2000], Fedora [Staples et al. 2003] or DSpace [Smith 2002], do not attempt this at all: the document is delivered to the user to be read using a third-party application (e.g. PDF). Although responsibility for presenting and interacting with the document structure can be shifted to another application, the same structural issues can still arise in this new context. Hence our requirements are also relevant to document viewers used in the Humanities, though for simplicity we assume the within-library approach used in Greenstone. It should be noted that handling everything in the digital library typically allows for more sophisticated and tailored uses of structure than generic document viewers.

Reuse of documents, as done in our case studies, is an important priority for the humanities [Barker et al. 2004; Hockey 2000]. Document encoding needs to account for structure which *may* plausibly be needed in future projects. Likewise, digital libraries need to be capable of handling the same structures if they are to support the full range of this work. Hence our analysis covers structure that could be used by both existing digital libraries and likely future systems.

Considering the design space of digital libraries, document structure can conceivably play a number of roles, including but not limited to:

Document browsing. Structure can be displayed in a table of contents (ToC) to aid navigation whilst *viewing* a document, e.g. evaluating, skimming, or reading in full a document.

Searching. Queries can be made over whole documents or their individual parts, and the search unit may be different from the result unit[Witten and Bainbridge 2003].

Collection browsing. Lists of documents can often be browsed by users, and high-level structure could be used to support this, e.g. to show a given edition of the bible contains these particular books.

Referencing. Parts of documents may be referenced within and between digital collections, e.g. an external reference to a specific chapter of novel in a particular library.

Our analysis focuses on document browsing and searching, as this reflects the current use of document structure in the literature, and Greenstone uses structure in both these contexts. In each of the following sections we discuss a problematic area of the use of structure in document browsing and searching.

5. DISPLAYING LARGE HIERARCHIES

The experience of browsing a document using a ToC presentation of the hierarchy is influenced by the size and shape of the structure [Chimera and Shneiderman 1994]. To a lesser extent this is also relevant if structure is used in search results, e.g. where a retrieved node can be displayed in the context of its ancestor nodes. We can reduce the size and improve these experiences by removing unnecessary structure.

5.1 Node content

Firstly, not all nodes of the hierarchy may be suitable for presentation in ToCs or search results. Some may contain so little content that readers are unlikely to want to browse or retrieve them individually. We call these *low-content nodes* that make the *fine-grained structure* of the document. Likewise, we can define *course-grained structure* as being the *high-content nodes* which make suitable chunks for browsing and retrieval. For instance, one may want to browse an anthology of poems by course-grained structure like pages or poems, but not by fine-grained structure such as stanzas or lines.

In fact, the high/low distinction is not always enough: fine-grained retrieval is useful in some contexts — for example, retrieving individual stanzas or bible verses — but these nodes may have so little content that individual browsing is not suitable. Hence we refine our node categorisation as follows:

High-content. Suitable for retrieving and browsing individually, e.g. chapters.

Medium-content. Suitable for retrieving individually but best browsed in the context of other nodes, e.g. paragraphs or verses.

Low-content. Too small to retrieve or browse on their own, e.g. words.

Note that any node that is considered worth browsing is also worth retrieving individually, so three categories suffice. Also, low-content and medium-content nodes may still be useful during browsing for enhancing the presentation of larger nodes.

These categories are defined with respect to the functions of digital libraries, so classifying specific nodes will in general depend on the context of design. Not all libraries will use any or all of this information, and if they do then they may not use it in the same way. The key point is that making the high/medium/low-content distinction can be useful when it comes to getting documents to work with a particular digital library.

Existing digital library systems do not distinguish these different categories of nodes. Many have a completely 'flat' document model, and of those that represent hierarchical document structure, such as Greenstone, all nodes can be browsed as individual units, i.e. every node is high-content. This discourages the use of lower-content nodes in collections, e.g. for searching, as they would result in extremely unwieldy ToCs, with the user having to constantly select nodes to read through the text, e.g. sentence by sentence. Greenstone has a mechanism for viewing all the text below a given node, but in this case the user will no longer be able to see the medium- and fine-grained structure. Ideally, a digital library should allow presentation of and interaction with low-content nodes, e.g. via highlighting, but not use them in ToCs or as a unit for browsing. The document model (and its documentation) needs to differentiate between the appropriate uses of different levels of structure.

To illustrate the node types, Figure 2 shows extracts of an XML document from the Bible in English collection. Near the root node there are nodes we might designate as high-content, used to markup the Old Testament and Book of Genesis (`<text>`), then its first chapter (`<div>`). The chapter summary (`<argument>`) and individual verses (`<v>`) could be considered as medium-content nodes. The low-

```
<text type="testament" name="Old Test.">
  <head>The olde Testament</head>
  <pb/>
  <text type="book" name="Genesis" test="old''>
    <head> ... </head>
    <runhead r="roman">The creation</runhead>
    <div type="chapter" n="1">
      <head>The first Chapter.</head>
      <argument> ... </argument>
      <v n="1">
        <s><hi>In</hi> <hi>the begin</hi>nyng
        <note type="concord"> ... </note> GOD
        created ye heauen and the earth. </s>
      </v>
      ...
      <pb/>
      ...
      <v n="31">
        <note type="concord"> ... </note>
        <s>And God sawe euery thyng that he had
        made: and beholde, it was exceedyng good.
        </s><s>And the euenyng &amp;amp; the mornyng
        were the sixth day. </s>
      </v>
    </div>
    ...
  </text>
</text>
```

Fig. 2. Structure of the Old Testament, from the Bishop's Bible, based on an SGML encoding by ProQuest. Some text has been omitted to better illustrate the document structure. The markup illustrates high- (`<text>`,`<div>`), medium- (`<argument>`,`<v>`) and low-content nodes (`<head>`, `<s>`) (see §5.1) and separate section and physical (`<pb>`) hierarchies (see §8).

content nodes are those that markup titles `<head>` and individual sentences `<s>`.

In our Greenstone collections we used high-content structure in document ToCs and medium-content structure for section-based search, whereas low-content structure was ignored as it had no functional role in the system. As the high/mediujm/low distinction was not made in the encoded documents we based our decisions on the SGML elements and attributes used — unfortunately these were not always well-documented or consistently used, and our classification involved a great deal of trial-and-error. This approach was far from satisfactory, resulting in a high percentage of misclassified nodes, i.e. extremely small sections processed as high-content and large sections processed as low-content.

5.2 Only-child nodes

One unnecessary feature in our collections' document structure was the only-child node: one which is the lone child node of another. It is not unusual to find only-children of certain types — e.g. a single paragraph may appear in a section of a book — but for the purposes of ToCs and structured search results they can be considered redundant. The nature of this redundancy depends on the design of the ToC interaction, but in general the purpose of such trees is to focus the system on a

```
<doc>
  <text type="edition" name="Mace" ...>            <!-- Only-child A -->
    <head ...>Daniel Mace (New Testament)</head>
    <text type="testament" name="New Test." ...>   <!-- Only-child B -->
      <head ...>New Testament</head>
      <text type="volume" name="Volume I" ...>
        ...
      </text>
      <text type="volume" name="Volume II" ...>
        ...
      </text>
      <back> ... </back>
    </text>
  </text>
</doc>
```

Fig. 3. Course-grained structure of the Daniel Mace New Testament (1729), as encoded in XML by ProQuest. It has only-child `<text>` nodes marked A and B. See Figure 4 for the corresponding Greenstone menu.

subset of the document's content. Parent and only-child have the same content, so at best they offer the user a false choice and waste screen space; at worst they force the user to engage in unnecessary interaction. This means avoiding high-content only-children: structuring a document as, say, a single volume containing a single chapter is unnecessarily complicated.

There are some circumstances under which a high-content only-child is not redundant: if the parent has mixed content (see §7 below) then the parent and only-child content will not be identical, and this is straightforward to determine automatically. More problematic is if a redundant only-child division is deliberately used in the original source — we have not encountered any examples of this, but cannot discount the possibility due to the diversity of humanities source material. In principle, document encoders could explicitly mark these rare occurrences as non-redundant.

We found several examples of only-child nodes in the Bible in English collection. These turned out to be unnecessary structure included because of the particular design of the markup scheme, but that were not needed to accurately encode the original text. Figure 3 shows the XML representation of a New Testament which illustrates the general problem: `<text>` nodes are always used at the edition, testament, volume and book level even when, as here, they are redundant. This information would have been better represented as metadata assigned to the appropriate nodes.

Redundant only-child nodes can be removed automatically by merging them with their parents. In general, this is not a straightforward process: one has to decide which types of nodes are redundant, and in each case a decision has to be made about which set of metadata to use or whether (and how) to merge. However, in our case the relatively small number of instances made this possible. Figure 4 shows the Greenstone menu for the New Testament of Figure 3, before and after the only child nodes have been removed.

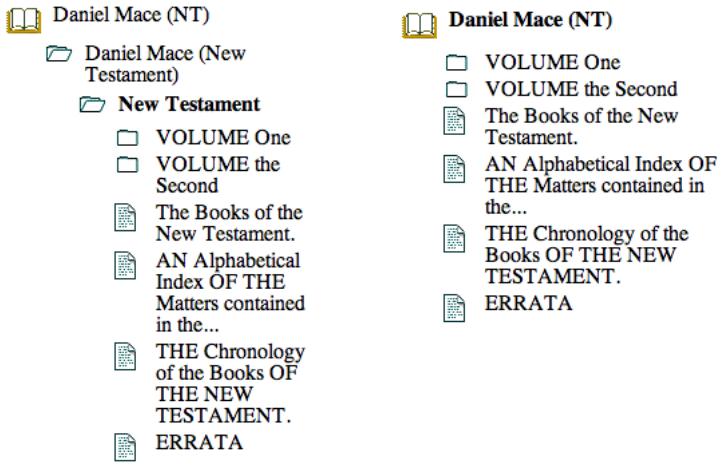

Fig. 4. Greenstone menus for the Daniel Mace New Testament (1729), with (left) and without (right) only-child nodes. Figure 3 shows the corresponding XML.

5.3 Node label length

The size of the hierarchy in ToCs and other overviews is also affected by the length of node labels. Lengthy labels can make the ToC unmanageably large, and trimming does not always produce a clear label. On the other hand, original titles are likely to be very useful for navigation and provide a rich experience of the content. We found numerous examples of extremely long node labels in our collections. For instance, John Sharrock's *The valiant actes And victorious Battailes of the English nation* (1585) contains an introductory verse node entitled "TO THE MOST EXCELLENT and most mighty Princesse Elizabeth, by the grace of God, of England Fraunce and Ireland Qveene, Defendresse of the Fayth, &c." In such cases the document encoding could provide both originals and short alternative labels, e.g. "To Princess Elizabeth." Node labels are not part of the underlying content, so the document itself is not actually changed.

6. NODE SELECTION

Having discussed factors which affect the hierarchy size, we now look at those which affect the ability of the user to select individual nodes within the hierarchy. Displaying a document ToC or search results presents the user with a choice of a collection of document nodes possibly several screen's worth. The digital library interface needs to support this selection task, which is influenced by clarity ("I know what A means") and discriminability ("I know I want A not B" [Blandford et al. 2001]) of the high- and medium-content node labels. These labels can come from a number of sources:

Assigned labels. One or more labels can be explicitly assigned to a node by the document encoder. They are likely to be clear and sufficiently different from one another to be discriminable, as the task of encoding will make the encoder reasonably familiar with the contents, although this will not always be the case. Useful

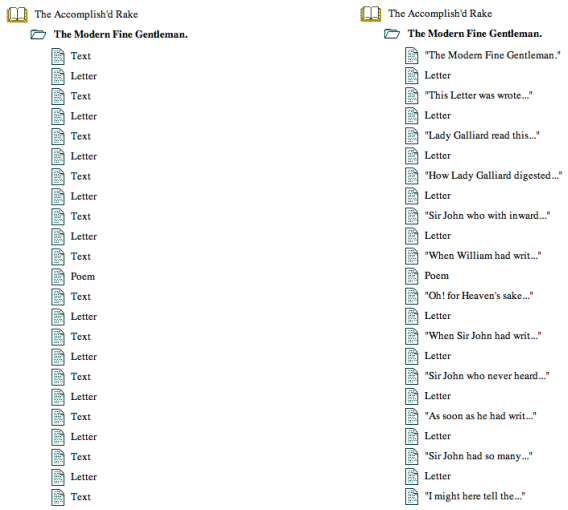

Fig. 5. Two document ToC for *The Accomplish'd Rake* (1727) by Mary Davys. The left-hand ToC illustrates the need for clear and discriminable node labels (see §6) and the 'inbetween' nodes (labelled 'Text') created from the text content between letters (see §7). The same ToC is shown on the right with content labels to used improve the discriminability of the inbetween' nodes. Figure 6 shows the corresponding XML.

alternative labels may be assigned for further clarity: short versions of lengthy labels (see also §5.3 above), or translations for labels in a language unlikely to be understood by the target audience. Mixed language documents are common in the humanities: an example from our collections is Matthew Coppinger's collected English verse (1682), which contains the poem "Omnia mutantur", Latin for "Everything changes."

Inferred labels. If no suitable label has been assigned by the document encoder, one can often be automatically inferred from the node contents when the document is imported. In building our collections we used header or title text (denoted by specific tags) that occurred near the beginning of a unlabeled node. Other strategies are possible for specific types of node, such as using initial text, e.g. first lines of poems, or labeling letters with their correspondents. Success depends on factors like the time available to the collector, the regularity of the source encoding, and the sophistication of any automatic processing. In the latter case, the functionality needs to be developed specifically for the document structure and may be unreliable. For even moderately sized collections one cannot be sure of the reliability of automatic processing.

Stock labels. If inferred labels can not be used, a stock label based on the type of node can be used, e.g. nodes marked up as letters can be labeled 'Letter'. These are less clear and discriminable than assigned or inferred labels, as they are not taken from the node contents and may be repeated if similar nodes are nearby (see Figure 5). Even labels like 'Introduction' or 'Index' that seem reasonably specific are less useful when documents have several such nodes. Discriminability can be slightly improved by adding numbers to differentiate identical siblings (e.g. 'Letter

```
<doc>
  ...
  <div type="Main Text" pn="1">
    <head> ... </head>
    <p> ... </p> ... <p> ... </p>
    <letter>
      <p> ... </p><p> ... </p>
    </letter>
    <p> ... </p>
    <letter>
      <p> ... </p><p> ... </p><p> ... </p>
    </letter>
    ...
  </div>
  ...
</doc>
```

Fig. 6. Course-grained structure of *The Accomplish'd Rake* (1727) by Mary Davys, based on an XML encoding by ProQuest. '...' indicates omitted XML. The `<div>` is a mixed content node (see §7), as it contains both high-content letter nodes (`<letter>`) and low-content paragraph nodes (`<p>`). Figure 5 shows the corresponding Greenstone menu.

3'), but this does not support the user's initial selection. However, this is perfectly acceptable for labeling structure without semantics, e.g. physical divisions like pages.

Default label. In the last resort a generic node can be identified by a default label — in our collections we used 'Text'. This scores badly in terms of clarity and discriminability. Early trials indicated that users were confused by this default label. We speculate that displaying document structure in a ToC naturally gives users the expectation that the structure will be meaningful: but why differentiate this node if it cannot be described more specifically? However, the benefits of ignoring this structure have to be balanced against support for navigating large documents.

Content label. For some types of document it may be useful to display a text label extracted from the node content, e.g. the first twenty words of content. These are likely to be highly discrimnating, but their clarity is more context-dependant: how well it describes the content will depend on the text and the extraction method used. Non-textual content nodes are problematic.

Figure 5 is an example of a poor quality ToC from our humanities collections, though it is representative of a significant proportion of the large documents. The left-hand ToC shows nodes with stock ('Letter') and default ('Text') labels, and the result the ToC is an unclear and undifferentiated list. Ideally, this would be have been avoided by the document encoder explicitly assigning labels. The right-hand menu shows an improved version where the default labels have been replaced with content labels.

7. MIXED CONTENT

Although many issues arise from the identification and labeling of useful structure, a very different set of problems can be caused by fundamental differences between

the hierarchies in document and digital library. One such issue is *mixed content nodes*: those which contains both text and child nodes. The term is borrowed from the identical situation in markup languages, where an element has mixed content if it contains a mixture of text and elements, e.g. `<h1>A <i>B</i> C</h1>`.

Mixed content nodes are common at all levels of the hierarchies in our humanities collections. However, mixed high-content nodes complicate the way browsing is handled in digital libraries, as selecting the mixed node requires both the child nodes and text to be displayed in a way which preserves their relative order within the node. Here the 'text' can include non-browsable objects, i.e. low- and medium-content nodes and text objects. In short, a mixed high-content node contains both high-content children and lower-content objects. It is not unusual to display document structure and text separately, e.g. a ToC displayed in a side-bar or popup, and in such designs it is hard to represent the ordering of the interleaved nodes and text objects. We are not aware of any digital library system that supports this properly.

Greenstone does not handle mixed content nodes adequately: returning to our example, it is analogous to rendering the HTML as **A C**, with ***B*** displayed separately. The Greenstone ToC displays the child nodes of a mixed content node without any indication of how text objects may be interleaved. Conversely, selecting this mixed content node from the ToC brings up a display of the text objects run together without any indication of how child nodes may be interleaved. A workaround for the user is to display all the text beneath the mixed content node, but this negates the benefit of having it structured in the first place. The key problem is that the document model supports mixed content, but the interface does not.

It is worth noting that the Greenstone document model works with mixed high-content nodes if there is only introductory text, i.e. the only text-object is at the beginning of the node's content. This is a reasonable assumption for a system which was designed to primarily import standard document formats (e.g. PDF, Word) where sections typically have introductory text, but not mixed content. However, this does not suffice in general, e.g. for XML and SGML.

Our humanities collections contained numerous examples of mixed high-content nodes with text being interspersed with child nodes, ranging from letters, poems and figures to entire stories within stories. For instance, in *Lucinda* (1739) by Penelope Aubin the text is twice interrupted by a character telling a story of some length, creating mixed content with high-content nodes surrounded by low-content nodes. Another example is shown in Figure 6, an XML encoding of *The Accomplish'd Rake* (1727) by Mary Davys, where the high-content 'Main Text' contains a series of low-content paragraphs (`<p>`), some of which are enclosed in high-content letter nodes (`<letter>`).

In order to represent this document structure faithfully in Greenstone, documents had to be restructured to avoid mixed content. We had the choice of either ignoring mixed structure or restructuring it into a non-mixed form. Ignoring it would have deprived the user of valuable information, with some documents completely unstructured above the paragraph level. Instead, we chose to create new high-content 'inbetween nodes' to hold any problematic low-content text. Returning to Figure 5 from §6, it show the *The Accomplish'd Rake* ToC after the document had

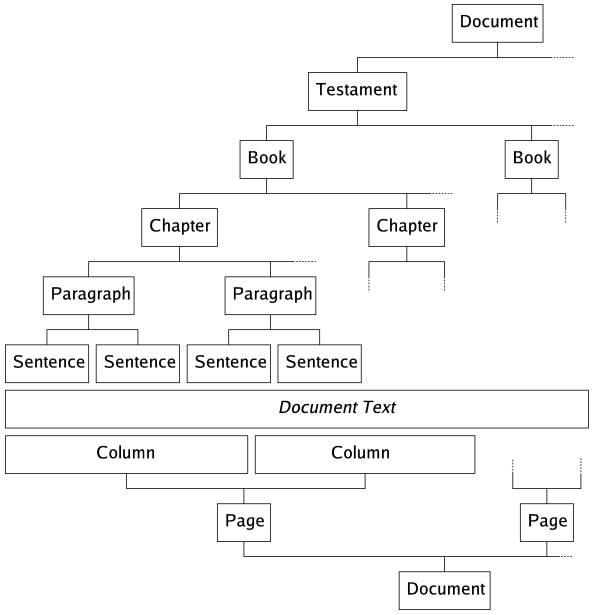

Fig. 7. Multiple hierarchies for a single document: a section hierarchy (above the text) and a physical hierarchy (below). Based on the structure of the Thomas Matthew Bible (1549) as encoded by ProQuest.

been restructured, with the left-hand ToC showing inbetween nodes label 'Text' by default. The right-hand ToC has the default labels replaced by content labels.

An alternative approach would be to remove problematic low-content objects by merging them with adjacent high-content nodes.

8. MULTIPLE HIERARCHIES

Multiple hierarchies allow several concurrent views of a document to be taken, and are considered important for the humanities domain [Renear 1997; Sperberg-McQueen and Burnard 2002]. The need for at least two hierarchies is illustrated by the two distinct structures commonly identified in text documents: the logical hierarchy and the physical hierarchy (illustrated in Figure 7). The logical hierarchy is what we often think of as the section structure of a text — volumes, chapters, paragraphs, sentences — and can feature a diverse range of special-purpose divisions, e.g. entries in an encyclopedia, poems in an anthology or recipes in a cookbook. The physical hierarchy shows how the text has been divided up for typesetting into volumes, pages, columns and lines. This is typically a flatter hierarchy with less diversity in types of node. In the humanities, more than in the sciences, the physical location of text within a document is often significant, as the document may be the object of study in itself [Adams and Blandford 2002].

Structural issues can arise with multiple hierarchies because they may be encoded in the document, they are almost never fully supported in the digital library. Few existing systems handle multiple hierarchies adequately. Some libraries allow

documents to be viewed by page and also browsed using a ToC —- however, in such cases only one hierarchy is fully developed (pages are the only physical node), hierarchies are assigned fixed and separate roles in the interaction (one cannot also view by logical section) and there is no support for the user to choose the hierarchy most appropriate to the task at hand. On the other hand, multiple hierarchies are well-supported by document standards, such as METS [McDonough 2006]. Although support for importing METS into libraries is still limited, both multiple hierarchies and METS are supported by Greenstone 3 — however, as usual each collection must assign the hierarchies a fixed and separate set of interactions.

XML does not support multiple explicit encodings, although implicit encoding is possible. For the three humanities collections we developed, the section hierarchy was encoded directly in XML, whereas the physical hierarchy was represented implicitly by using empty elements to mark the boundaries between divisions, shown in Figure 2. The two `<pb>` elements mark page boundaries, i.e. all the text between the markers belongs to a single page division, and the `<runhead>` element associates a header with this page. This structure was dropped from our collections as Greenstone 2 only allows a single hierarchy and we judged the logical hierarchy to be the more useful of the two for humanities scholars.

Given that the humanities submit documents to particularly close analysis, the potential for exploiting multiple document hierarchies goes beyond sections and pages. There is already some evidence that users of online help systems find alternative conceptual structures useful [Hunt et al. 1993]. One can imagine alternative hierarchies — perhaps supplied by the reader — corresponding to alternative analyses of the text, involving characters, themes or temporal reorderings of the text (i.e. that present a narrative in a correct time sequence).

9. REQUIREMENTS

Our analysis has found a range of specifically structural interaction issues with the way document structure is handled in Humanities digital libraries. In this section we address these with two sets of user-centred requirements: one for document encoding and another for digital library design.

9.1 Requirements for Document Encoding

E1. Documents should classify each node as high-, medium- or low-content. A recurring theme in our analysis is the distinction between high, medium- and low-content nodes: high-content nodes are suitable for browsing and retrieving individually; medium-content nodes are suitable just for retrieving; low-content nodes are suitable for neither. The distinction is important for reducing hierarchy size (§5), for assigning labels (§6) and for identifying mixed content (§7).

As we noted above, one difficulty is that the categories are defined with respect to the *functions* of digital libraries, whereas document encoders are working with declarative data. Hence categorisation, like many other aspects of encoding [Butler et al. 2000], is a judgment about the document that needs to be made by the encoder. They are free to use any particular encoding of this information: the key point is that decisions are documented and the encoding can be interpreted later on so that the low/mid/high distinction can be made in the library. The distinction supports our other requirements.

E2. Every high- and medium-content non-physical node should be assigned a descriptive label. Short and translated additional labels may be assigned when appropriate. As discussed in §6, the user needs labels that they can understand and can distinguish from siblings and other close relations.

E3. Avoid encoding high- and medium-content only-child nodes. These nodes waste screen space and user effort (see §5.2). We discussed how high-content only-children are almost always redundant. For medium-content nodes — used for retrieval but not browsing — the situation is less clear, but avoiding only-children where possible will simplify the use of document structure in search.

E4. Mixed high-content should be avoided where possible. Existing systems do not support this well: the interface needs to represent the relative order of the nodes and text.

9.2 Requirements for Digital Libraries

Our requirements for digital libraries have been inspired by our experience with Greenstone. However, we did not restrict our analysis to the specific ways document structure is used in that system, and so the requirements are relevant to any digital library that supports Humanities users, especially in the context of large source documents. The idea here is to represent the information available in the document, rather than prescribe any particular style of interaction.

L1. Search over high- and medium-content nodes, and browsing over high-content ones should be supported. This is in line with the roles encoded in the document, following requirement E1.

L2. Where possible short node labels in the main collection language should be used in ToC and search results, and longer more descriptive node labels when viewing the document. Any additional labels should always be accessible to the user. Conciseness, clarity and discriminability are important when viewing multiple labels (§6), while the richer original content is more significant in the document itself.

L3. Where no node label is assigned by the document, the library should (in order of preference) infer one from the node content, substitute a stock label, or use an appropriate default label.

L4. Browsing of mixed high-content nodes should be supported, either directly or by restructuring to a equivalent non-mixed form. Direct support is preferable, as introducing 'inbetween' nodes reduces the quality of the ToCs (see §7). However, as noted above existing systems do not support this well.

L5. Where appropriate, at least two document hierarchies, logical and physical, should be supported in browsing and search. The user should have a choice of hierarchy for all node interactions. As discussed in §8, existing systems rarely support multiple hierarchies, and when they do interaction with them is restricted.

10. CONCLUSIONS

Although humanities collections can benefit from rich hierarchical models of document structure, providing a good end-user experience of this structure requires

consideration in both the encoding of documents and the design of digital libraries. Electronic encoding of documents from the original source material is often a separate activity from organising the electronic documents in a digital collection. The two activities may be performed by distinct organisations and may be uncoordinated.

The user-centred requirements we have proposed focus on two separate hierarchical document structures: that of the encoded document and that represented in the digital library. Our analysis of the specifically structural interaction problems that can be caused by differences and inadequacies in these structures has been supported by a case study where ProQuest's data was imported into Greenstone. However, the same problems — and perhaps others — have the potential to arise with other combinations of system and data. Our requirements place constraints on both with the aim of improving the end-user experience and make the task of creating collections easier. This is important for the sustainability and reuse of digital resources.

We intend to continue this work by redesigning the way document structure is handled in Greenstone to reflect our recommendations. Improved versions of the three humanities collections will be employed in future user studies to further our understanding of the use of document structure, and other aspects of digital library use, by humanities scholars. Although this work has been motivated by such research, we hope our discussion and recommendations will be more widely relevant to other domains where document structure can be exploited.

ACKNOWLEDGMENTS

This work was funded by the EPSRC grant "User-Centred Interactive Search with Digital Libraries" (GR/S84798). The authors would like to thank ProQuest Information and Learning for the use of their digital collections, and William Newman for his insightful comments.

REFERENCES

Adams, A. and Blandford, A. 2002. Digital libraries in academia: challenges and changes. In *Digital Libraries: People, Knowledge, and Technology*, E.-P. Lim, S. Foo, C. Khoo, H. Chen, E. Fox, S. Urs, and T. Costantino, Eds. LNCS, vol. 2555. Springer, 392–403.

Allen, R. B. 1995. Interface issues for interactive multimedia documents. In *Digital Libraries, Research and Technology Advances, ADL '95 Forum, McLean, Virginia, USA, May 15-17, 1995, Selected Papers*, N. R. Adam, B. K. Bhargava, M. Halem, and Y. Yesha, Eds. Lecture Notes in Computer Science, vol. 1082. Springer, 179–189.

Barker, E., James, H., Knight, G., Milligan, C., Polfreman, M., and Rist, R. 2004. Long-term retention and reuse of e-learning objects and materials. Tech. rep., Arts and Humanities Data Service. AHDS Reports on Preservation.

Biggs, M. and Huitfeldt, C. 1997. Philosophy and electronic publishing. the theory and metatheory in the development of text encoding. *The Monist 80*, 348–366.

Blandford, A., Stelmaszewska, H., and Bryan-Kinns, N. 2001. Use of multiple digital libraries: a case study. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital libraries*. ACM Press, New York, NY, USA, 179–188.

Buchanan, G., Cunningham, S. J., Blandford, A., Rimmer, J., and Warwick, C. 2005. Information seeking by humanities scholars. In *Research and Advanced Technology for Digital Libraries, 9th European Conference, ECDL 2005, Vienna, Austria, September 18-23, 2005,*

Proceedings, A. Rauber, S. Christodoulakis, and A. M. Tjoa, Eds. Lecture Notes in Computer Science, vol. 3652. Springer, 218–229.

Buchanan, G., Gow, J., Blandford, A., Rimmer, J., and Warwick, C. 2007. Representing aggregate works in the digital library. In *Proc. JCDL 2007*. ACM, 247–256.

Butler, T., Fisher, S., Hockey, S., Coulombe, G., Clements, P., Brown, S., Grundy, I., Carter, K., Harvey, K., and Wood, J. 2000. Can a team tag consistently? experiences on the orlando project. *Markup Languages Theory and Practice 2,* 2, 111–125.

Chimera, R. and Shneiderman, B. 1994. An exploratory evaluation of three interfaces for browsing large hierarchical tables of contents. *ACM Trans. on Information Systems 12,* 4, 383–406.

Dori, D., Doermann, D., Shin, C., Haralick, R., Phillips, I., Buchman, M., and Ross, D. 1997. *Handbook on Optical Character Recognition and Document Image Analysis*. World Scientific, Chapter The Representation of Document Structure: a Generic Object-Process Analysis, 421–456.

Dumais, S. T., Cutrell, E., and Chen, H. 2001. Optimizing search by showing results in context. In *Proceedings of the SIG-CHI on Human Factors in Computing Systems, March 31–April 5, 2001, Seattle, WA, USA*. ACM, 277–284.

Egan, D. E., Remde, J. R., Gomez, L. M., Landauer, T. K., Eberhardt, J., and Lochbaum, C. C. 1989. Formative design-evaluation of SuperBook. *ACM Trans. on Information Systems 7,* 1, 30–57.

Furnas, G. W. 2006. A fisheye follow-up: further reflections on focus + context. In *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, Eds. ACM, 999–1008.

Hitchcock, S., Carr, L., Jiao, Z., Bergmark, D., Hall, W., Lagoze, C., and Harnad, S. 2000. Developing services for open eprint archives: globalisation, integration and the impact of links. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA*. ACM, 143–151.

Hockey, S. 2000. *Electronic Texts in the Humanities: Principles and Practice*. Oxford University Press.

Hunt, W. T., Rintjema, L., and Carey, T. T. 1993. User acceptance of complementary tables of contents for access to online information. In *Human-Computer Interaction, INTERACT '93, IFIP TC13 International Conference on Human-Computer Interaction, 24-29 April 1993, Amsterdam, The Netherlands*, S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, and T. N. White, Eds. ACM, 181–182. Jointly organised with CHI'93, Adjunct Proceedings.

Lin, X. 1996. Graphical table of contents. In *Proceedings of the 1st ACM International Conference on Digital Libraries, March 20-23, 1996, Bethesda, Maryland, USA*. ACM, 45–53.

McDonough, J. P. 2006. METS: Standardized encoding for digital library objects. *Int. J. on Digital Libraries 6*, 148–158.

Renear, A. 1997. Out of praxis: Three (meta) theories of textuality. In *Electronic Text: Investigations in Method and Theory*, K. Sutherland, Ed. Oxford University Press, 107–126.

Rimmer, J., Warwick, C., Blandford, A., Gow, J., and Buchanan, G. 2006. User requirements for humanities digital libraries. In *Digital Humanities 2006, 1st ADHO Int. Conf.* CATI, Universit Paris-Sorbonne.

Smith, M. 2002. DSpace: An institutional repository from the MIT Libraries and Hewlett Packard Laboratories. In *Research and Advanced Technology for Digital Libraries, 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002, Proceedings*, M. Agosti and C. Thanos, Eds. Lecture Notes in Computer Science, vol. 2458. Springer, 543–549.

Sperberg-McQueen, C. M. and Burnard, L., Eds. 2002. *TEI P4: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative Consortium.

Staples, T., Wayland, R., and Payette, S. 2003. The fedora project: An open- source digital object repository system. *D-LIb Magazine 9,* 4 (April).

Stelmaszewska, H. and Blandford, A. 2004. From physical to digital: a case study of computer scientists' behaviour in physical libraries. *Int. J. on Digital Libraries 4*, 82–92.

Theng, Y. L., Duncker, E., Mohd-Nasir, N., Buchanan, G., and Thimbleby, H. 1999. Design guidelines and user-centred digital libraries. In *Research and Advanced Technology for Digital Libraries, Third European Conference (ECDL'99)*, S. Abiteboul and A.-M. Vercoustre, Eds. LNCS, vol. 1696. Springer, 167–183.

Witten, I. H. and Bainbridge, D. 2003. *How to Build a Digital Library*. Morgan Kaufmann.

Witten, I. H., Boddie, S. J., Bainbridge, D., and McNab, R. J. 2000. Greenstone: a comprehensive open-source digital library software system. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA*. ACM, 113–121.