Representing Aggregate Works in the Digital Library

George Buchanan[1], Jeremy Gow[2], Ann Blandford[2],
Jon Rimmer[3], and Claire Warwick[3]

[1] University of Wales, Swansea g.r.buchanan@swansea.ac.uk
[2] UCL Interaction Centre, London j.gow,a.blandford@ucl.ac.uk
[3] UCL SLAIS, London j.rimmer,c.warwick@ucl.ac.uk

Abstract. This paper studies the challenge of representing aggregate works such as encyclopaedia, collected poems and journals in the digital library. Reflecting on the materials used by humanities academics, it demonstrates the complex range of aggregate types and the problems of representing this heterogeneity in the digital library interface. We demonstrate that aggregates are complex and pervasive, challenge many common assumptions and confuse the boundaries between organisational levels within the library. The challenge is amplified by referring to concrete examples, and solutions are demonstrated in a well-known digital library system and related to standard digital library architecture.

Keywords: Digital Libraries, Architecture, Collection Building

1 Introduction

Digital libraries started as collections of documents that were commonly of the same genre and form. For example, a library collection may be of sheet music, oral history recordings or development literature. This reflected both historic organisational behaviour in the digital library, and the technical challenges of the contemporary period. Due to the costs of scanning and recording historic material, most of the material used was 'born digital'. However, increases in funding for making vintage pre-digital material available online have increased the number of DL collections of older material.

Our study of humanities academics [4] has increased our focus on the digital provision of 'born paper' literature. Part of the project has been the creation of collections of material dating from the 5th to 18th centuries CE. In undertaking this apparently simple task, we have found significant challenges in representing historic material in a way that is both consistent with the original material and interactional design constraints.

In our research, we have discovered that *aggregate works* are of particular importance to humanities academics. Aggregate works are constituted from a number of different items — e.g., a collection of poetry [12]. Aggregates occur frequently in humanities collections, and discovering the correct item in the aggregate that academic references identify is critical to a correct understanding of journal articles, etc. In repositories of historic material — be they libraries or archives — works originally created (manuscripts) or published separately are bound together for the purposes of (physical) long-term storage. When items are made available electronically, they need to reflect the physical binding that has been referenced in the past. Also, reference can be made to items which though the original is now destroyed (e.g. due to war) still exist in part or whole in earlier publications. The item is still referred to directly, but now needs to be identified in regard to published copies, rather than as a reference to some now non-existent document. As we shall show, the complications of aggregate forms and representation provides some illogical consequences in the digital domain.

Digital libraries commonly store 'documents', which are essentially atomic: e.g. in DSpace [13] a document is represented by a set of one or more binary files and accompanying descriptive metadata. One common and simple aggregate is the journal. If a collection is built of individual journal articles, then one document can consistently represent one article, an issue of the journal is a set of articles, and a volume a set of issues, etc. It would appear logical that a similar approach should be effective for other aggregates. However, that is not the case.

To take a simple example, if a work is bound in two separate volumes, then it would make sense to separate between the two — particularly as references to the content of the work will often refer to a particular page within a particular volume. However, that means that we now have two separate 'documents' in the library, which need to be linked for the purposes of browsing or searching. A counter-example can be found in many 19th century factual works such as Eugène-Edouard Boyer-Peyreleau's "Les Antilles françaises", which is printed in three volumes. However, each volume itself contains a "Book", each of which contains chapters, etc. Thus, in this case the book and volume relationships are reversed: volumes contain many books, just as is the case with the Bible. Similarly, collected volumes of small novels can contain many works within one volume, each of which is commonly referred to under the collective title.

These issues of binding would be simplified if users of libraries only searched by a particular item — the volume or the work it contained. Naturally, however, readers use whichever reference data they have, and weakly defined searches may behave very differently if the library distinguishes between individual works, or subsumes them within some more heterogenous whole. For a library to support effective searching and browsing, retrieval must be reliable under both criteria.

It is worth re-iterating at this point that the examples we have given so far are the most simple cases: as we shall see, the reality with historical material is often significantly more complex due to both the historical production of material and its contemporary academic understanding.

This paper proceeds in three parts: first we enumerate a number of complex cases of aggregate documents, and highlight the conflicting issues of form and identity. Then we discuss the problems identified in the context of current DL systems, and identify some solutions and outstanding issues. The paper closes with a discussion of related literature and the course for future research.

2 Aggregate Structures in Practice

This section introduces some sample problems of aggregate works in historic literature. Before we discuss different aggregate types, it is worth clarifying our model of what an aggregate is. We presuppose the existence of some kind of 'document unit'. An ordered series of these may be collected together to form an aggregate, which can in turn form part of another aggregate. So, aggregate works are ordered trees with documents units at the leaves.

The nature of these aggregated document units is not constrained by the model, although in this paper we restrict ourselves to textual examples. Units may be themselves be documents, or they may be parts of documents — where aggregation stops and internal document structure commences is not necessarily clear, and often depends on the needs of the library users and the resources of the library administration. Such issues are beyond the scope of this paper.

In the following discussion, we enumerate some features of aggregate works that have important implications for the design of digital libraries. In order to clarify the subsequent discussion, each aggregate feature is given a unique identifier 'AG*n*' . Note that the features are not all mutually exclusive, so a single aggregate work may exhibit several of them.

Homogenous Aggregation (AG1) The example above of journal articles aggregated in issues can be termed *homogenous aggregation*. Each aggregated unit is of the same type. However, many journals also contain editorial content and indexes. Articles themselves may be, say, original academic contributions or reviews of other literature (e.g. published books). Therefore, the pure homogenous collection is actually rather rare, as our examples below illustrate.

Heterogenous Digital Forms (AG2) Though an aggregate work may be logically homogenous, its digital form may vary internally. This can arise where an aggregate has been digitised over a period in which digitisation practice shifted. For example, we encountered this situation while handling content from the Oxford Text Archive[4]: some works were initially encoded in transcript version in COCOA/Tact, but later parts were both scanned and supplied with a TEI transcript. Such heterogeneity clearly complicates the ingest of the aggregate.

Serial Aggregation (AG3) Some forms of aggregation emerge from the requirements of publishing. In the case of journals, one common form is the release of an aggregation at regular (or semi-regular) intervals to represent the works from a given period. This can also happen with larger works that are published over many years. For example, Sir John Fortescue's "History of the British Army" was published in twenty parts over a twenty-five year period (in the case of the first edition) and thus each part has its own year of publication. This work is even more complex as some parts are multi-volume themselves, and

[4] http://www.ota.ox.ac.uk/

some parts refer to the same historical period yet were printed at different dates.

Binding Aggregation (AG4) This emerges from the technological limitations at the original time of printing: a work was printed and released as one item, but bound in separate volumes. A simple case would be a multi-volume work in print: though clearly one work, digitally we may need to treat the work as being of separate parts in order to support accurate retrieval (e.g. from references that distinguish a volume of the work).

Composite Aggregation (AG5) Another historic form of aggregation *within the same work* was its release as a serial within a larger aggregate work. This sounds very much like **AG3**; however in that case a work is printed in discrete bound parts that only contain elements of the same work. *Composite aggregation* occurs when a work is published in parts, but each part is itself bound in a different aggregate. In the 19th century, fiction was often serialised in newspapers. Famous works by Charles Dickens and Sir Arthur Conan Doyle were published in this way, e.g. Sherlock Holmes stories in *The Strand* magazine. Though now available bound as separate works, their original printed form is the aggregation of the separate columns and pages in the containing periodical. Thus, what is from one point of view a part of a newspaper is at the same time a part of a novel. This is an important historical aggregation which is of strong interest to humanities researchers.

Containing Aggregation (AG6) A work may be small and unavailable in its own right, but is available contained within larger works. However, those larger works are not simply aggregates themselves. This is a point where aggregation and internal structure of documents collide. A practical example can be found in cases such as poems within A.A. Milne's stories of Pooh: the poems have a "life" of their own, independent of the holding work. However, often such contained works are not even by the same author. For example, T.S. St.Clair's "A residence in the West Indies and America" contains a chapter composed of an independent work by his brother on Martinique. The book as a whole is not a composite, but it does contain in its entirety an article not by the author, and often referred to directly without reference to the enclosing work.

Heterogenous Aggregation (AG7) As noted at the beginning of this section, heterogenous aggregation is more commonplace than we may think — even within modern journals. In what may be termed 'simple' heterogenous aggregation a work is created from units of diverse types. For instance, newspapers and journals contain articles of different types that may need to be distinguished in the DL interface. Another example occurs when a library has bound tracts — small texts printed without hardback bindings — into sets by size. It may be necessary to reconstitute such arbitrary aggregations in digital form for reference purposes.

Supplementary Aggregation (AG8) A common form of aggregation in historical literature is where an original work is supplemented by further material, possibly by another author. For example, a volume of memoirs may be published with a foreword and/or additional notes. These are also known as *augmented works*, but aggregation may be a more appropriate model where the additional material is substantial, or referenced independently.

Incomplete Aggregation (AG9) Some aggregates are incomplete, either because they were not fully published or because a collection may be incomplete. Some works were published with the intention of being part of a series, but that series was never fully published, leading to "missing" volumes.

Variable Aggregation (AG10) Different versions or editions of an aggregate work may bring together different material, or different versions of the same material. For instance, the New Testament is an aggregation of works, and in earlier forms often omits books found in modern versions, or may even contain other books that are now omitted.

As noted above, the boundary between dealing with external and internal document structure is not fixed, and many of the issues discussed above as the aggregation issues may also occur *within* a given document. For example, a journal or diary of historic interest may be considered to be a homogenous aggregate of diary entries, perhaps with supplementary hetergenous content from letters received by the author. What is important, from the view of a DL system, is that the treatment of internal and external aggregation are treated consistently in the DL architecture and also in the user interface, to ease the task of readers and librarians alike. Having surveyed some of the important features of aggregate works, we now look in more detail at their use in digital libraries.

3 Aggregates in the Digital Library

This paper has emerged from our need to represent historic literature in a digital library. Our own experience has been using the well-known Greenstone digital library software [15]. In this section we also refer to other popular DL systems such as DSpace [13]. We commence by studying some simple existing forms of aggregation and composition in the digital library that helps establish foundations from which design choices can be formed. The section then turns to some specific complexities we have encountered, with reference to the aggregate types found in Section 2. The problems that can emerge from simple renderings of these aggregates are then pointed out, and where solutions have been achieved these are reported.

3.1 Simple Cases

Some common forms of aggregation are already well understood in the digital domain. One example is the academic journal, released at (usually) regular time intervals and of a standardised total size of content per issue. The online access of journals is now commonplace. Each issue is usually treated as a homogenous aggregation of articles, and each volume as a homogenous aggregation of issues. Such regular structure allows for a simple representation in the digital library: each article is represented as a separate document; each issue is represented by a node in a hierarchy; and each issue node is itself a child of a volume node. This structure is represented in Fig. 1. However, this treatment proves suboptimal in certain circumstances: when a search is performed, the result list is of documents, and only inspection of discrete document metadata can identify where, for example, a particular issue (e.g. topical special issue) of the journal particularly matches the search.

Users in the humanities often need to search for particular forms of article, e.g. they may seek reviews of books on a topic. This then results in a need to search by genre. Some systems such as JSTOR permit this as an advanced option. This effect is readily achieved by adding the genre as a metadata property of the collection: the collection remains, in indexation terms, homogenous, but selectivity can be achieved by adding metadata criteria to particular searches. Thus, simple heterogeneity can be represented whilst retaining a simple underlying digital representation of the material.

Thus, we can readily represent chronologically serial (**AG3**) aggregates with simple heterogenous (**AG7**) or homogenous (**AG1**) material using standard digital library software. The regular hierarchy of the journal format is easily mirrored in browsing hierarchies and genre distinctions between articles can be made during searches, and potentially in the interface, using document metadata. The only problem that remains is the particular identification of single issues that particularly match the user's interest — which can only be identified by visual inspection of metadata (if visible).

3.2 Conflicting Aggregations

Though the regular form of journal collections results in few problems, it is easy to identify problematic situations that differ in only small, yet critical, details. If the aggregation hierarchy is not the same for all library content, or is even potentially ambivalent, problems rapidly multiply. If a collection is built of pieces of literature, the scale of each item varies from a short novella to a multi-volume "epic". In such situations, if we faithfully replicate the physical text in digital form (critical for supporting referencing to particular pages) some texts will be formed from separate volumes, whilst the majority are contained in only one. Conversely, a single volume may contain several discrete works and be known to the community both separately and as a combined volume. The concept of 'volume' thus becomes problematic.

Indexing a collection by volume only conflates works that share the same volume, whilst indexing by works only conflates volumes of the same work — placing work on the reader to separate the parts after retrieval. Clearly, neither solution is optimal: the natural conclusion is to index by whichever unit (work, volume) is the smaller and aggregate upwards in the collection to unify elements

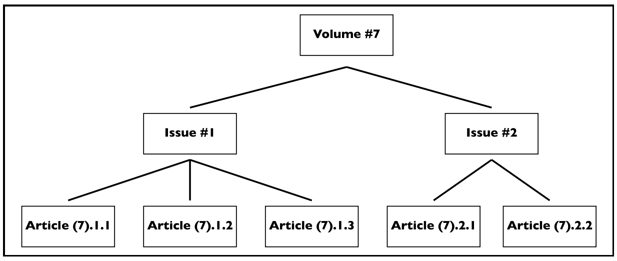

Fig. 1. A simple periodical aggregate

of the same item. This underlying storage can be represented in different ways in the library interface: e.g., matches against a single search for separate volumes of the same work can be unified in the search result list, and retrieval made by volumes. This option is already available in the Greenstone Digital Library (see [15] for details), and can be achieved in DSpace with careful configuration.

During browsing, however, the contradictory use of volume (as a part or as an aggregate) will still emerge in some form or other. One can distinguish the part-of and aggregate-of styles of volume by introducing a three-level hierarchy and using discriminating labels for the top and bottom levels. Many items are represented by only one item at each level, and as reported in [14] such simple single-child relationships should be pruned so that unnecessary interaction is minimised. Thus, to improve the interactional efficiency, the experienced hierarchy becomes irregular. The issues of unifying hierarchy nodes in search result lists remains a problem (though this can be achieved in Greenstone).

3.3 Difficult Cases

So far, we have referred to relatively simple cases of aggregation that can be resolved to some degree in the DL interface. Now we turn to more complex cases that create increasing degrees of difficulty. Here, we focus on *containing aggregation* (**AG5**) and *composite aggregation* (**AG6**). Other forms also produce complications, but here for reasons of space we must highlight only a few.

Composite aggregates (**AG5**) represent particularly problematic structures. Serialised fiction such as Conan Doyles *Gang of Four* disrupts DL assumptions in its original form. If we naively recorded each newspaper or magazine in a collection as a single document, then the reader would need to map their information need (to read the Gang of Four in its original context) to particular editions of the correct publication — something that they may not immediately have to hand. For the period over which the work was published, the reader must then identify which particular editions contains part of the story.

An alternative approach would be to extract and record the elements of the story as one DL document, tidily avoiding the problem for a searcher specifically looking for the Gang of Four, but conversely divorcing it from its original context: to connect each article with its context in the original magazine, the

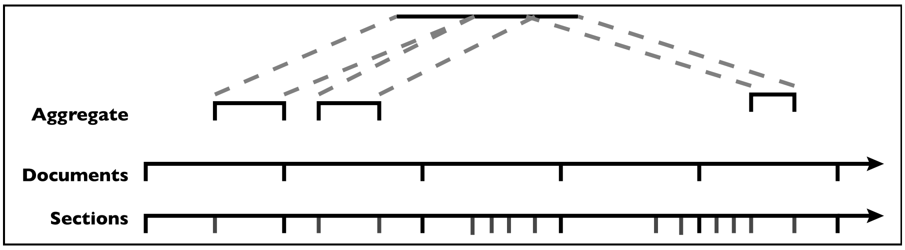

Fig. 2. Indexation of an aggregate

user must in fact engage in the 'hunting' of articles we apparently just avoided. Such contextual interpretation is the nub of many items of humanities research. Clearly, an optimal approach allows both the recovery of the original composited piece, and the magazines of which it was part.

3.4 Indexing Text

Two key architectures of DL systems can be noted: what we may term 'bytestream' architectures [6] that treat documents as a collection of metadata and corresponding binary content that is delivered to the reader; and 'full text' architectures that treat documents as metadata and organised text (where the content is textual) [15]. 'Bytestream' architectures generally do not index the full text of the document, whereas full text architectures both index the text and are aware of its internal structure. Clearly, particular systems and particular installations may vary. For example, DSpace is often installed as a simple bytestream architecture, but can optionally be used with the Lucene XML indexer to index some or all of the content of a collection [13].

In the case of composite aggregates, there are two key indexation options for full-text indexation: first, each aggregate form is indexed independently and as a whole — this results in the duplication of text within the index; second, each unique unit of text is recorded only once, but is referred to by each aggregate that uses it. Hybrid approaches that combine these two methods are clearly possible. The duplication approach can readily be achieved with any library; however, the second approach is not as straightforward to exploit as it may at first appear: additional indexation structures need to be present in the DL.

Like many full-text search engines, MG [16] can index documents at a number of different internal levels: the text is indexed once, and each level of internal document structure maps to a single span across the text index (see. Fig. 2). However, this model of a continuous and contiguous stream of blocks does not support composite aggregation which combines separate, discrete blocks of text. A supplementary map structure links between an aggregate document to its text blocks; the document weight for each aggregate must also be calculated.

Again typically for a search engine, MG expects every document to be delivered for indexation in sequence with its full text. This adds a second complication: when an aggregate is to be processed, it has no content of its own –

that is found in its constituents. Either the data to link it to its constituents can be transmitted to the search engine at this point, or it can be omitted and processed in a separate phase of indexation.

We chose to use this second approach, and added a third phase to MG's two indexation phases specifically for aggregate indexation. This phase records the map structure and calculates document weights for aggregates. In the first two phases (dictionary creation, compression and indexation) document text is indexed and compressed. The only change made is that aggregation data is used to identify when a document or document part is duplicated within the input data. Where duplication occurs, the second and subsequent occurrences are marked, and not passed for processing. Instead, in the second (compression and indexation) phase the document's pointer to its content in the index is made to point to the first occurrence. All multiple occurrences are temporarily stored in a lookup table. When the third, aggregation, phase is run, this receives a hierarchical set of identifiers for each aggregate. Each set identifies which documents or document parts consititute the aggregate. The aggregate record is then completed by translating the input document identifiers with their offsets into the compressed index, and the document weight for the aggregate is then calculated by first creating a word list for the *whole* document and calculating its weight accordingly. For faster processing, MG's approximate weighting system [16] can be used to create an aggregate document weight.

When searches are performed, they can either use the original MG index (retrieving simple documents only), or use the third-level index to return aggregates. In the latter case, the result list can be set to contain only aggregates or both aggregates and documents. Note that a single input fileset may be separated into two 'documents' for the purposes of indexation by MG.

Broder's work on detecting similar and contained documents [2] may seem very close to what we propose here. However, in this context different editions of the same document should not be viewed as identical. Similarly, containment does not simply infer aggregation. To properly identify aggregation one needs knowledge of the literature and its historical context — which cannot be automatically inferred from the documents alone. However, Broder's techniques will undoubtedly prove useful in supporting this activity.

Indexation and Classification In Sec. 3.1 we reported that simple aggregates can be represented through using classification structures, and that simple consolidation of different parts of an aggregate can be achieved in search result lists. We have just reported the incorporation of aggregate items into a text index. It is worthwhile pointing out the different advantages of these two approaches.

Where aggregates are represented by classifiers, they are usually not known to the underlying search engine — particularly for a DL with a componentised services architecture [11, 3]. Thus classifier nodes do not have a document weight and cannot reliably be used for ranking. Conversely, where knowledge of aggregates is supported in the text indexer, different problems emerge: for example, if a specific document that appears in several aggregates matches a search, how should it be displayed? In its own right, or within an aggregate? If within an aggregate, which one? If the aggregated document is a good match, and its aggregating parents a poorer match, then the apparent solution differs from where the opposite applies.

In both cases, there are interactional concerns. For example, the default interface of Greenstone allows the user to choose which level of document structure to search at: e.g. within documents, sections or paragraphs. Given the variable numbers of levels that occur in aggregates, and the fact that the structure of aggregates differs, this simple approach breaks down, e.g. the 'volumes' labelling clash referred to in Sec. 3.2. The manual selection of search granularity also places a burden on the user. It will take considerable exploration and research to find optimum solutions for such problems.

3.5 Reading Text

Aggregation also complicates the delivery of text for reading. For example when a PDF file contains three works, each work could be indexed individually by the DL system. If the user now wishes to read the material, in current DL systems the entire PDF will be delivered. As we observed in [1], users often inspect only the top of any digital document. If the visible head of the document does not match the user's expectations (e.g. the title of a constituent work rather than the aggregate name) it is likely that they will incorrectly conclude that an error has occurred or the document they wish is not available. Therefore, a comprehensive handling of aggregates must deliver material for reading in a way that is consistent with the collection index. For certain file formats (e.g. XML) this is relatively easy for most cases, whereas for other formats (e.g. PDF) it is often more complex. Where delivery of the whole aggregate is necessary or desireable, information should appear in the DL interface to ensure that the user's expectations are aligned with what they will receive if they download the work. Where DLs already allow the download of material in different digital formats (e.g. Greenstone supports delivery in HTML, PDF or Word of the same document) then this interface idiom may be readily extendable to delivering content of different scope (work, volume, etc.). However, the cautions over confusions in download formats also noted in [1] will almost certainly still apply.

4 Discussion

The difficulties of the representation of aggregate works in digital libraries has already received attention. For example, Hickey and O'Neill [5] note a number of problems in applying the Functional Requirements for Bibliographic Records (FRBR) [9]. They propose treating aggregates as published volumes of more than one work, and to avoid recording aggregates as works in their own right. This introduces an inconsistency with the accepted FRBR model where every published volume (*manifestation*) is an instance of a single work.

Two standard electronic document formats allow for the representation of aggregate works: TEI [8] and METS [7]. In each case, aggregates are achieved by pointers, be they between content of the same file or to separate files, to create a whole. TEI primarily uses pointers between parts of the aggregate, whereas in METS a document contains references to part or whole other METS documents – these parts then form sections of the current document.

Aggregates have been poorly represented in digital library systems to date. Though it is conceivable that Fedora's object-based architecture [6] may be able to represent aggregation, we have not been able to discover any published coverage of this issue. Popular systems such as DSpace [13] and Greenstone [15] have focussed on treating collections as sets of objects, with a hierarchical classification structure. Aggregates can be represented using the classification structure, as we demonstrated in Sec. 3.1, but at the loss of consistent treatment of aggregates across both searching and browsing.

One may hope that practice and experience from library science would be helpful. However, the historic need to find and recover texts via bound volumes has emphasised the approaches we have seen in DL systems. Aggregates are generally indexed by part where the parts are discrete works: e.g. the British Library [5] binds brief tracts together in volumes, but each tract in a volume is indexed separately. Conversely, multi-volume works are usually, but by no means universally, indexed with only one entry. In the case of the British Library, practice here varies from work to work.

Svenonius [12], p. 103, notes that there are two potential routes to relating aggregates with their constituent parts: first, formal linkage structures; second, providing descriptive aggregation (meta-)data for each item. The latter approach, though informal and easy to apply, leaves much of the retrieval work with the user, and greater room for mismatches between the descriptive data and the corresponding description of the part or aggregate in the catalogue index.

5 Conclusion

In this paper, we introduced a number of different forms of aggregate works in the digital library. We demonstrated that simple types of aggregation are easily supported in DLs, with only small shortcomings in the representation. However, more complex forms of aggregation which occur frequently in historic literature, both factual and non-factual, map less readily to existing DL architectures and interfaces. We reported on changes to Greenstone and MG that widen the forms of aggregation that can be successfully represented in DLs. We briefly reported on the complications of integrating these changes into the DL user interface, and the corresponding solutions. Though aggregates have only recently started to receive attention in the digital domain, a considerable amount of work will be required to move from the initial steps represented here and in other projects such as the IFLA Working Group on Aggregates [10] to a complete solution. In

[5] www.bl.uk

our case, we wish to investigate further the appropriate interactions to support the occurrence of aggregates in search result lists, and the location of desired aggregates in the course of information seeking.

6 Acknowledgements

This research is supported by EPSRC Grant GR/S84798.

References

1. A. Blandford, H. Stelmaszewska, and N. Bryan-Kinns. Use of multiple digital libraries: a case study. In *Proc. 1st ACM/IEEE-CS Joint Conference on Digital libraries*, pages 179–188, New York, NY, USA, 2001. ACM Press.
2. A. Broder. On the resemblance and containment of documents. In *Procs. Compression and Complexity of Sequences*, pages 21–29, 1997.
3. G. Buchanan, D. Bainbridge, K. J. Don, and I. H. Witten. A new framework for building digital library collections. In *JCDL '05: Proc. 5th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 23–31. ACM Press, 2005.
4. G. Buchanan, S. J. Cunningham, A. Blandford, J. Rimmer, and C. Warwick. Information seeking by humanities scholars. In *Proc. 9th European Conference on Digital Libraries*, pages 218–229. Springer, 2005.
5. T. B. Hickey and E. T. O'Neill. FRBRizing OCLC's WorldCat. *Cataloging and Classification Quarterly*, 39:239–251, 2005.
6. C. Lagoze, S. Payette, E. Shin, and C. Wilper. Fedora: An architecture for complex objects and their relationships. 2005.
7. Library of Congress. *Metadata Encoding and Transmission Standard (METS).*
8. C. Sperberg-McQueen and L. Burnard, editors. *Guidelines for Electronic Text Encoding and Interchange.* TEI P3 Text Encoding Initiative, Oxford, 1999.
9. Study Group on the Functional Requirements for Bibliographic Records. *Functional requirements for bibliographic records.* K.G. Saur, 1998.
10. Study Group on the Functional Requirements for Bibliographic Records. Minutes of the FRBR review group's meeting, Aug 2005.
11. H. Suleman and E. A. Fox. Designing protocols in support of digital library componentization. *Proc. European Conference on Digital Libraries*, 2458:568–582, 2002.
12. E. Svenonius. *The Intellectual Foundation of Information Organization.* Digital Libraries and Electronic Publishing. MIT Press, 2000.
13. R. Tansley, M. Smith, and J. H. Walker. The DSpace open source digital asset management system: Challenges and opportunities. In *Proc. European Conference on Digital Libraries*, pages 242–253. Springer, 2005.
14. Y. L. Theng, E. Duncker, N. Mohd-Nasir, G. Buchanan, and H. Thimbleby. Design guidelines and user-centred digital libraries. In *Proc. 3rd European Conf. for Digital Libraries, ECDL*, pages 125–134. Springer-Verlag, 1999.
15. I. H. Witten, S. J. Boddie, D. Bainbridge, and R. J. McNab. Greenstone: a comprehensive open-source digital library software system. In *Proc. ACM conference on Digital libraries*, pages 113–121. ACM Press, 2000.
16. I. H. Witten, A. Moffat, and T. C. Bell. *Managing gigabytes (2nd ed.): compressing and indexing documents and images.* Morgan Kaufmann, San Francisco, 1999.