Digital Libraries Without Databases: The Bleek and Lloyd Collection

Hussein Suleman

Department of Computer Science, University of Cape Town
Private Bag, Rondebosch, 7701, South Africa
hussein@cs.uct.ac.za

Abstract. Digital library systems are frequently defined with a focus on data collections, traditionally implemented as databases. However, when preservation and widespread access are most critical, some curators are considering how best to build digital library systems without databases. In many instances, XML-based formats are recommended because of many known advantages. This paper discusses the Bleek and Lloyd Collection, where such a solution was adopted. The Bleek and Lloyd Collection is a set of books and drawings that document the language and culture of some Bushman groups in Southern Africa, arguably one of the oldest yet most vulnerable and fragile cultures in the world. Databases were avoided because of the need for multi-OS support, long-term preservation and the use of large collections in remote locations with limited Internet access. While there are many advantages in using XML, scalability concerns are a limiting factor. This paper discusses how many of the scalability problems were overcome, resulting in a viable XML-centric solution for both greater preservation and access.

1 Introduction and Motivation

Digital Library Systems (DLSes) have shared a close relationship with database systems as such databases are often the underlying data storage fabric of the DLS. The almost inseparable nature of this relationship is clear in popular tools such as EPrints [10] and DSpace [9], that nominally use mySQL and Postgres respectively to hold their primary metadata repositories. The databases provide an efficient mechanism to add, update and retrieve items from a collection. Alternatives to databases that fulfil the same needs may be feasible replacements.

It can be argued that in some situations a database may not be the most desirable or effective storage mechanism and that efficient solutions can be crafted from other structured data storage technologies. In the digital preservation arena, it has already been suggested and demonstrated that XML and its sister technologies may be better suited for aspects of digital preservation [2] [11].

A convergence of factors provides additional support for this hypothesis. Firstly, there is a growing acceptance that managing large quantities of data is one of the key challenges for digital libraries. The Storage Resource Broker [8] abstracts the details of data access in dealing with scalability, thus creating middleware that is independent of actual underlying databases or filesystems. Secondly, most modern DLSes have accepted the importance of interoperability. This is usually achieved by supporting DL standards such as the Open Archives Initiative Protocol for Metadata Harvesting [5] and/or Web standards such as Really Simple Syndication (RSS) [14]. In both cases, metadata is transferred from one collection to another without any notion of how the source or destination metadata is stored or processed. Finally, archival storage mechanisms such as the OAI Static Repository Gateway Specification [6] are being defined at the level of structured data in XML, rather than database-centric tables. All of these factors indicate a higher abstraction for structured data, that is increasingly making it possible to connect and architect systems without knowing about the underlying database used for storage, or even not using an underlying database at all.

In comparing database-centric solutions to XML-centric solutions, databases have a clear advantage in some areas, including:

- the efficiency of insertion and retrieval operations;
- a well-established and standardised query language;
- and existing installations of the software on most servers.

There are, however, a few challenges in using databases, and most of these are in fact addressed by XML data stores and XML-specific data manipulation tools. Some key issues are enumerated in Table 1.

Based on this comparison of data representation approaches, it does appear that XML-based solutions may be more appropriate for some problems. However, efficiency and scalability are still concerns that must be addressed.

This paper discusses how an XML-centric solution was devised for the Bleek and Lloyd Collection; and how the scalability and efficiency concerns were addressed for this project.

2 The Bleek and LLoyd Collection

The Bleek and Lloyd Collection [13] is a collection of paper-based artefacts that document the culture and language of the |Xam and !Kun groups of Bushman people. It is widely held that the Bushman people are among the oldest known ethnic groups in the world. Alas, the Bushman way of life - including oral traditions, language, morality and relationship with the natural environment - has been largely subsumed by the onslaught of Western civilization, and the |Xam and !Kun languages are already extinct. In general, it is estimated that within only a few years the last generation of Bushman people who are knowledgable in the ancient customs will have passed away. This impending loss of an entire ancient culture underscores the importance of any and all preservation activities. Digital preservation of the Bleek and Lloyd Collection is currently underway in this context.

The books and drawings that constitute the bulk of the Bleek and LLoyd Collection are jointly owned by the National Library of South Africa, the Iziko

Table 1. Challenges in using database systems and how these map to XML-centric solutions

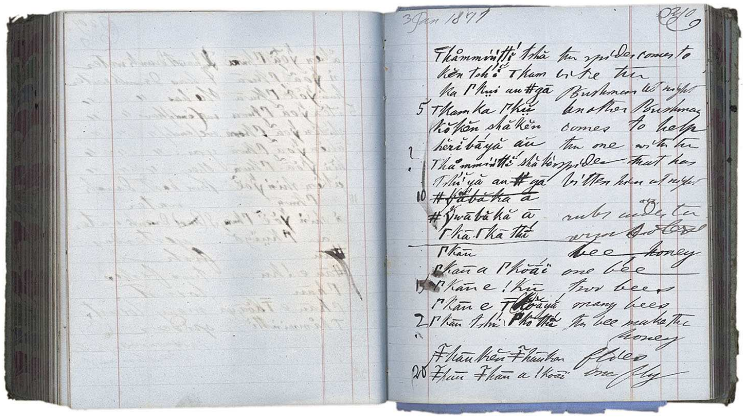
Issue	Database Systems	XML Systems
Installation	Needs to be installed and running, and on multi-user systems is often owned by the administrator	No need for daemon or administrator privileges, and many tools are commonly embedded in Web browsers
Platform	Systems are not usually platform-independent because of performance tuning	There are many tools to manipulate XML and modern Web browsers integrate some of them, e.g., DOM parsers
Processing	Data must be extracted before it can be processed	Backups, data transformations, etc. require only handling of flat files so can be conducted at the OS level
Long-term preservation and access	Databases are usually stored in binary formats for efficiency, therefore their data is not human-readable	XML data is always human-readable

National South African Museum and the University of Cape Town. The books record information obtained by Wilhelm Bleek and Lucy Lloyd in the 19th century from prisoners interned at the Breakwater Prison in Cape Town. Drawings done by these same individuals supplement the narratives in the notebooks. Figures 1 and 2 provide examples of the book page images and drawing images respectively.

In 1997, these artefacts were added to the UNESCO Memory of the World register. This stresses the need for them to be preserved at all costs. At the same time access needs to be granted to researchers and scholars around the world, including in Africa, where Internet bandwidth is often poor or non-existent.

In 2003, the Lucy Lloyd Archive and Research Centre at the Michaelis School of Fine Art arranged for all artefacts to be scanned at high resolution and then generated metadata and re-keyed the text for each granular object [7]. These images and metadata were then used as the basis for this digital library system.

There are a total of 157 notebooks, containing a sum of 14128 page images. The page images correspond primarily to pairs of facing pages, but they also include inserts, covers and spines. The average size of image files is approximately 172 kilobytes - these are low-resolution versions of the images that are to be made available on a DVD-ROM version of the collection. The page images are in JPEG format.



3 Jan 1871

Thåmmṑttĕ tkhā	the spider comes to
kĕn tkhō tkam	bite the
ka tkui au ‡gā	Bushmen at night
tkam ka tkui	another Bushman
kŏpĕn shă kĕn	comes to help
kĕribāyā āu	the one with the
Thåmmṑttĕ shā kĕn	spider that has
tkhūyā au ‡gā	bitten him at night
~~‡kăbāka ā~~	
‡kwābăkă ā	rubs under the
tkhā tkhā tkhē	arm to ~~[illegible]~~
tkān	bee honey
tkān a tkhōāī	one bee
tkān e !kū	two bees
tkān e tkhoāyā	many bees
tkān tkhi Phō tkhē	the bee make the honey
‡kān kĕn ‡kān kan	flies
‡kān ‡kān a !khoāī	one fly

Fig. 1. Page image from notebook

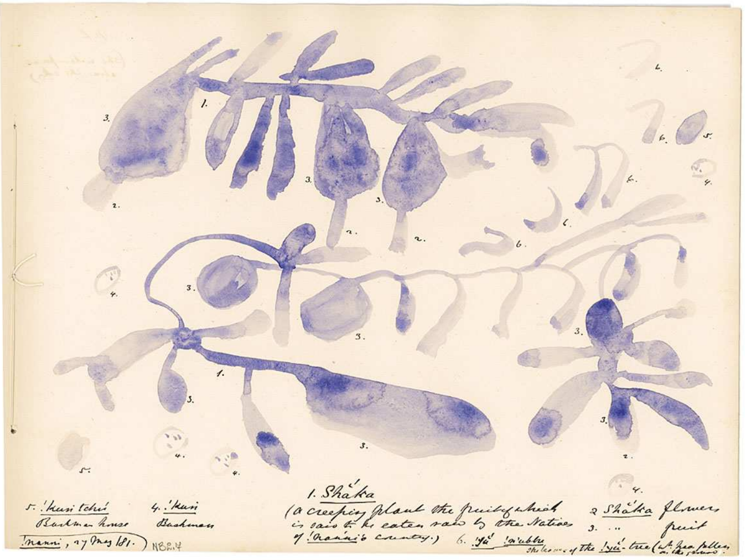

1.
2.
3.
4.
5.
6.
5. !kusi tchui
Bushman house
4. !kusi
Bushman
1. Shăka
(a creeping plant the fruit of which is said to be eaten raw by the Natives of !nanni's country.)
2 Shăka flowers
3. " fruit
6. !yū !n'ubbe
the leaves of the !yū-tree (wh. has fallen on the ground.
!nanni, 27 May 181.
NB2.4

Fig. 2. Image from drawing

There are a total of 752 drawings. These are scanned versions of single pages. The average size of each drawing file is approximately 15 kilobytes and these images also are stored in JPEG format.

Each image filename is unique and the images are arranged into a few high-level directories corresponding to notebook number or source of images.

In both cases, Excel spreadsheets contained all metadata. The spreadsheets for drawings contained one row of fields per drawing. The spreadsheets for notebooks contained one row of fields for each story. These stories were manually determined from the notebooks and each is associated with a list of page ranges, possibly across notebooks. All spreadsheets were exported to XML as a first step in processing the information therein.

3 Data Pre-processing

A usable meta-structure was first built by pre-processing the source datasets. For this and further data transformation, the notebooks were dealt with separately from the drawings, using similar techniques.

As a starting point, the Excel-generated XML documents were interpreted, cleaned and checked for correspondence with the image files.

The source data contained inconsistences introduced by human editors. For example, a page range was indicated in many different ways (e.g., from `A1_1 - A1_10`, `A1_1` to `A1_10`) as was a single page (e.g., `A1_1.JPG`, `A1_1`). In both instances, the pre-processor determines which is intended (using pattern matching) and includes the appropriate individual pages.

Some data cleaning also was necessary. Various characters were used incorrectly (e.g., I and 1, O and 0) and typographical errors need to be fixed so that linking of files would occur correctly. All source data was in Unicode because some of the characters used in the !Kun and `|Xam` languages are not easily representable otherwise.

Finally, filenames generated from the metadata were matched with actual filenames of images in the filestore. This 2-way check ensured that no files were left out and no missing files were referenced in the metadata.

The end result is one clean and consistent XML file containing structured metadata for the notebooks and another containing structured metadata for the drawings. Excerpts of different portions of the former XML file are shown in Figure 3.

4 Early Alternatives

4.1 Greenstone

Initially, Greenstone was considered to be the ideal solution for the Bleek and Lloyd Collection because it is one of the rare DLSes that will export a collection to CD-ROM for wide distribution and offline viewing [15]. However, Greenstone requires some basic software installation and this creates a portability problem if the system is meant to work on arbitrary operating systems.

```
<data>
   <stories>
      <story><id>1</id><collection>Wilhelm Bleek
       Notebooks</collection><title>Covers and first
       pages of Bleek&apos;s Book I or BC151_A1_4_001
       </title></story>
      ...
   </stories>
   <categories>
      <category><id>2</id><name>Words and sentences
       </name></category>
      ...
   </categories>
   <authors>
      <author><id>2</id><name>Adam Kleinhardt</name>
       </author>
      ...
   </authors>
   <keywords>
      <keyword><id>2</id><kw>vocabulary</kw>
       <subkw>|xam</subkw></keyword>
      ...
   </keywords>
   <pages>
      <story id="6">
         <page>A1_4_1_00160.JPG</page>
         <page>A1_4_1_00161.JPG</page>
         ...
      </story>
      ...
   </pages>
   <books>
      <collection name="Wilhelm Bleek Notebooks"
       source="images/bleek_nb_lowres">
         <book name="BC_151_A1_4_001">
            <page>A1_4_1_FUCOV.JPG</page>
            ...
         </book>
      </collection>
   </books>
</data>
```

Fig. 3. Excerpts from source XML data file

4.2 Single PDF

As a prototype, a single PDF was created to store the notebooks. An XSLT transformation first mapped the internal XML format to XSL-FO [1], a page layout language, then the XSL-FO data was converted into PDF output.

This solution had very poor scalability. The PDF file for just 20% of the notebooks was approximately 182MB in size and took approximately 30 seconds to load on an average Pentium 4 PC (circa 2003). This PDF file included only thumbnails of the page images, with links to the local directory for the full versions. The advantage of a single PDF file was its self-contained search, browse and linking capability. However, this solution led to slow response times as well as low-quality presentation of information. In addition, as PDF sizes increase, accessibility to the collection is compromised even over reasonably fast Internet connections (whereas linked XHTML pages can be accessed easily and quickly from a local drive or online).

PDF was thus abandoned at an early stage in favour of an XHTML rendering. Separate PDF files could be created for each book, but indices across book boundaries will require other techniques.

5 Scalable Hyperlinked XHTML

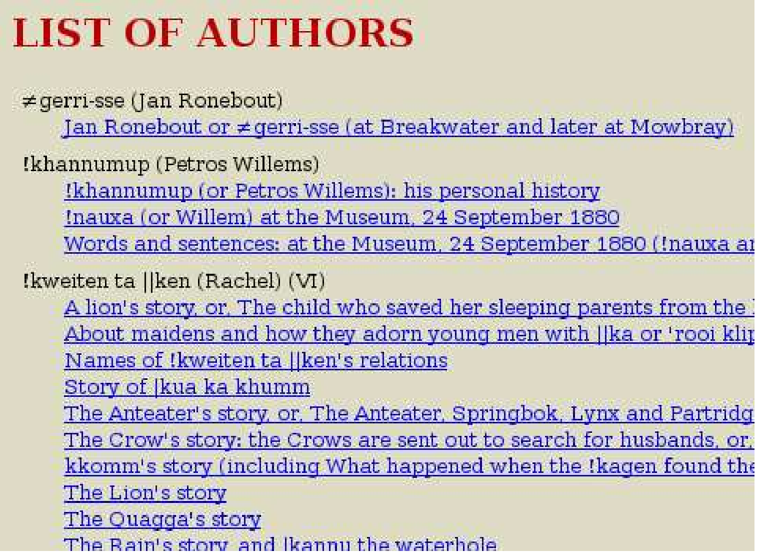
5.1 Overview

XHTML pages were pre-generated from the XML source data using XSLT stylesheets. A better alternative may be to generate these client-side on demand, but some major browsers (e.g., Opera) do not support client-side XSLT.

The collection can be browsed and individual items accessed using hyperlinks. An Ajax-based search system was integrated into the XHTML pages - pre-generated inverted files were stored in an XML format and a Javascript routine performed the query operation. Thus searching can be conducted completely within the browser, with no server-side search engine necessary.

The following discussion focuses on scalability concerns addressed in the generation of hyperlinked XHTML files from XML. Searching is discussed elsewhere.

Given a source XML document, an XSLT transformation was created to generate either an index page or a list of individual pages for each subsection of the navigation. For notebooks, the page images can be listed by author, keyword, category, book or story. Thus, for example, the stylesheet can generate a list of authors and the stories attributed to each or a set of pages corresponding to each of the stories with full details on that story. The entire collection was represented in the source XML document so that it is possible to generate next/previous links in some places and also to perform iteration over subsets of the collection within the XSLT. Figure 4 displays a listing of authors and their stories, as generated by this process while Figure 5 displays a single page corresponding to one story.



LIST OF AUTHORS

≠gerri-sse (Jan Ronebout)

Jan Ronebout or ≠gerri-sse (at Breakwater and later at Mowbray)

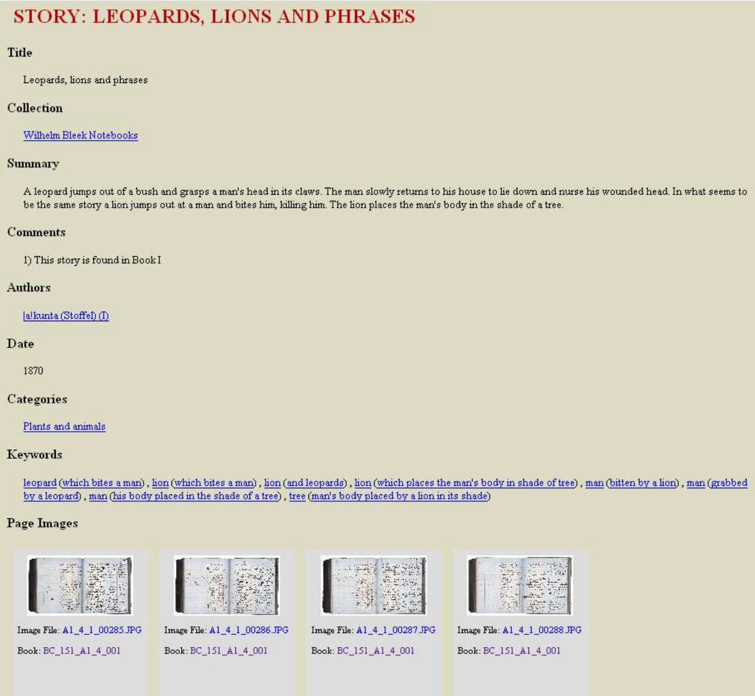
!khannumup (Petros Willems)

!khannumup (or Petros Willems): his personal history
!nauxa (or Willem) at the Museum, 24 September 1880
Words and sentences: at the Museum, 24 September 1880 (!nauxa ar

!kweiten ta ||ken (Rachel) (VI)

A lion's story, or, The child who saved her sleeping parents from the l
About maidens and how they adorn young men with ||ka or 'rooi klip
Names of !kweiten ta ||ken's relations
Story of |kua ka khumm
The Anteater's story, or, The Anteater, Springbok, Lynx and Partridg
The Crow's story: the Crows are sent out to search for husbands, or,
kkomm's story (including What happened when the !kagen found the
The Lion's story
The Quagga's story
The Rain's story, and |kannu the waterhole

Fig. 4. List of authors and stories from each author



STORY: LEOPARDS, LIONS AND PHRASES

Title

Leopards, lions and phrases

Collection

Wilhelm Bleek Notebooks

Summary

A leopard jumps out of a bush and grasps a man's head in its claws. The man slowly returns to his house to lie down and nurse his wounded head. In what seems to be the same story a lion jumps out at a man and bites him, killing him. The lion places the man's body in the shade of a tree.

Comments

1) This story is found in Book I

Authors

|a!kunta (Stoffel) (I)

Date

1870

Categories

Plants and animals

Keywords

leopard (which bites a man) , lion (which bites a man) , lion (and leopards) , lion (which places the man's body in shade of tree) , man (bitten by a lion) , man (grabbed by a leopard) , man (his body placed in the shade of a tree) , tree (man's body placed by a lion in its shade)

Page Images

Image File: A1_4_1_00285.JPG
Book: BC_151_A1_4_001

Image File: A1_4_1_00286.JPG
Book: BC_151_A1_4_001

Image File: A1_4_1_00287.JPG
Book: BC_151_A1_4_001

Image File: A1_4_1_00288.JPG
Book: BC_151_A1_4_001

Fig. 5. Story listing with metadata and thumbnails of page images

5.2 Memory

XSLT v1.0 only creates a single output document for each transformation. Thus, in order to create multiple files, these were structured into subsets of a single large tree which was interpreted externally to create the actual files. The ability to output multiple files is available as extensions and in XSLT v2.0. However, portability and memory management were concerns that led to this not being pursued.

The size of source and destination XML documents during the transformation process determined how much memory was used during transformations. The source XML is fixed but the destination XML depends on the number of subsets of the XHTML data being created simultaneously. All XSLT transformation engines crashed when trying to generate the entire set of XHTML documents at once. To deal with this scale issue, the external application controlling the process passed in a parameter to specify precisely which subsets of the XHTML pages to generate. Then the XSLT processor was executed multiple times to generate the entire set of pages.

5.3 Indexing

XPath expressions were used to locate matching nodes and automatically insert extensive cross-referenced links in the generated XHTML pages. These expressions, however, quickly proved to be too slow for practical reasons as they often seemed to be implemented in the XSLT processors as linear time scans. To alleviate this, XSLT keys were used to create indices for particular complex structures. This is akin to database indices and the search performance increased substantially, as expected. Multiple keys also were used just as they would serve the same purpose in databases.

5.4 Single XML Source

Keys used to cross-reference data also crossed boundaries across metadata fields. For example, an author may be indexed to a story in one instance but elsewhere the author may need to be indexed to a book. This high degree of linking is the reason why a single source XML document was used, even when XSLT supports multiple source documents. Initially, multiple smaller XML documents were used, but as more links were inserted into the target XHTML, some indices needed to cross source XML document boundaries. This is not currently supported by XSLT.

5.5 Grouping

While the source data contained many authors in arbitrary order, the generated pages contain a listing of all stories, grouped by author. The standard solution to generate such a listing is to search for all names that are different from the immediately preceding name. However, finding a preceding sibling is expensive for a large dataset (algorithm complexity $O(n)$) and very slow if this has to be done for every item in a list (algorithm complexity $O(n^2)$).

The Muenchian Method [12] was used as a more efficient alternative. This technique involves using keys to order the names. Then, for each name, the current node can be combined with the first result found from a key search to create a nodeset. If the nodeset contains only a single node it means that the current node is in fact the first one found during a key search. Thus, unique names can be found with algorithm time complexity $O(n)$, which was sufficiently efficient for the datasets in this project.

This technique was extended to other indices. The keyword index (keyword to subkeyword to list of pages) used multi-level grouping, also with linear time complexity.

5.6 Performance

The 2 steps of the process were individually timed on a dual-CPU 3GHz Xeon 2GB RAM machine running FreeBSD 5.

Generation of the source XML document took 13.14 seconds for notebooks and 2.67 seconds for drawings.

Generation of 15426 XHTML documents related to the notebooks took 1 minute and 3.73 seconds. Generation of 1059 XHTML documents related to the drawings took 3.53 seconds. Given that this collection corresponds to the contents of a complete DVD-ROM, and the collection is not expected to grow in size, this is an acceptable processing rate.

Any alternative that did not reuse images or included embedded images in document formats (such as PDF by default) would not have been feasible.

Navigating through the collection is near-instantaneous. This was tested off a local drive, a network shared drive, a local and remote website and a DVD. Viewing the collection smoothly across multiple technologies is a distinct advantage over popular DLSes and database-driven approaches.

6 Conclusions

This paper has demonstrated how XML+XSLT+XHTML can be used to generate a usable and useful static and portable digital library.

XML-centric solutions have been recommended for heritage-based digital collections because of the expected long-term preservation of data. This paper has illustrated how such a solution was implemented for a real-world collection, addressing scalability and efficiency concerns in both generation of and access to the collection.

Databases may be most suitable for some problems, such as institutional repositories; but XML-centric solutions surely offer greater advantages for other problems, such as heritage preservation.

7 Future Work

Probably the most interesting future project would be to create tools for the automatic generation of XML-based collections and renderings thereof without having to hand-craft either data formats or transformations. This is similar to the Greenstone approach and one solution may be to have Greenstone generate static XML collections instead of or in addition to its own internal collection formats.

An alternative is to transform Greenstone so that a user can access a collection without any installation of software.

Scalability is a major current concern in digital preservation/libraries [3] [4]. While current XML technology is reasonably scalable, as shown in this paper, it is necessary to devise similarly portable techniques to deal with arbitrary and massive quantities of information.

As more tools are generated to deal with static DL collections, it may be necessary to formalise how such collections are represented and include support for import and export of such collections in keeping with current content packaging and representation standards, such as VRA-Core and METS.

8 Acknowledgements

This project was made possible by funding from University of Cape Town, NRF (Grant number: GUN2073203) and the Lucy Lloyd Archive Resource and Exhibition Centre.

References

1. Berglund, Anders (2006), Extensible Stylesheet Language (XSL) Version 1.1, W3C Recommendation, W3C, 5 December. Available http://www.w3.org/TR/2006/REC-xsl11-20061205/
2. Digitale Bewaring (2002) XML and Digital Preservation: Digital Preservation Testbed White Paper, Dutch National Archive. Available http://www.digitaleduurzaamheid.nl/bibliotheek/docs/white-paper_xml-en.pdf
3. Haedstrom, Margaret (2003), "Research Challenges in Digital Archiving and Long-term Preservation", NSF Post Digital Library Futures Workshop, 15-17 June 2003, Cape Cod. Available http://www.sis.pitt.edu/ dlwkshop/paper_hedstrom.html
4. Imafouo, Amlie (2006), "A Scalability Survey in IR and DL", TCDL Bulletin, Volume 2, Issue 2. Available http://www.ieee-tcdl.org/Bulletin/v2n2/imafouo/imafouo.html
5. Lagoze, Carl, Herbert Van de Sompel, Michael Nelson and Simeon Warner (2002), The Open Archives Initiative Protocol for Metadata Harvesting – Version 2.0, Open Archives Initiative, June 2002. Available http://www.openarchives.org/OAI/2.0/ openarchivesprotocol.htm
6. Lagoze, Carl, Herbert Van de Sompel, Michael Nelson, Simeon Warner, Patrick Hochstenbach and Henry Jerez (2004), Specification for an OAI Static Repository and an OAI Static Repository Gateway, Open Archives Initiative, April 2004. Available http://www.openarchives.org/OAI/2.0/guidelines-static-repository.htm

7. Lucy Lloyd Archive and Resource and Exhibition Centre (2007), Lloyd Bleek Collection, University of Cape Town. Available http://www.lloydbleekcollection.uct.ac.za/index.jsp
8. Moore, R., C. Baru, A. Rajasekar, B. Ludascher, R. Marciano, M. Wan, W. Schroeder, and A. Gupta (2002), Collection-Based Persistent Digital Archives Parts 1 and 2, D-Lib Magazine, April/March 2000. Available http://www.dlib.org/dlib/march00/moore/03moore-pt1.html and http://www.dlib.org/dlib/april00/moore/04moore-pt2.html
9. Smith, Mackenzie, Mick Bass, Greg McClellan, Robert Tansley, Mary Barton, Margret Branchofsky, Dave Stuve and Julie Harford Walker (2003), DSpace: An Open Source Dynamic Digital Repository, D-Lib Magazine, Volume 9, Number 1, January 2003. Available http://www.dlib.org/dlib/january03/smith/01smith.html
10. Sponsler, E, and E. F. Van de Velde (2001), Eprints.org Software: A Review, Sparc E-News, August-September 2001. Available http://resolver.library.caltech.edu/caltechLIB:2001.004
11. Strodl, Stefan, Andreas Rauber, Carl Rauch, Hans Hofman, Franca Debole and Guiseppe Amato (2006), The DELOS Testbed for Choosing a Digital Preservation Strategy, in S. Sugimoto, J. Hunter, A. Rauber and A. Morishima (eds): Proceedings of 9th International Conference on Asian Digital Libraries (ICADL 2006), 27-30 November, Tokyo, Japan, Springer-Verlag, pp. 81-90.
12. Tennison, Jenni (2007), Grouping Using the Muenchian Method. Available http://www.jenitennison.com/xslt/grouping/muenchian.html
13. University of Cape Town (2003), Jewel in UCT's crown to be digitised for world's scholars, Monday Paper, 31 March 2003. Available http://www.uct.ac.za/print/newsroom/mondaypaper/?paper=114
14. Winer, Dave (2002), RSS 2.0 Specification, Berkman Centre for Internet and Society. Available http://blogs.law.harvard.edu/tech/rss
15. Witten, Ian, Sally-Jo Cunningham, Bill Rogers, Roger McNab and Stefan Boddie (1998), Distributing Digital Libraries on the Web, CD-ROMs, and Intranets: Same information, same look-and-feel, different media, Proc First Asia Digital Library Workshop: East Meets West, edited by J. Yen and C.C. Yang, Hong Kong, pp. 98-105. Available http://nzdl.sadl.uleth.ca/gsdl/collect/publicat/index/assoc/HASH0199/95cc7f7f.dir/doc.pdf