Reflections on Design Principles for a Digital Repository in a Low Resource Environment

Hussein Suleman

University of Cape Town, South Africa
hussein@cs.uct.ac.za
http://dl.cs.uct.ac.za/

Abstract

Low resource environments are characterised by a lack of technical skills, facilities and funding. In such environments, building digital repositories of historical and heritage collections is particularly difficult, exacerbated by the desperate need for digital preservation. This paper analyses design principles for digital repositories and their suitability for low-resource environments. A case study is presented to illustrate how adopting the positive principles in support of low-resource environments can support the creation of a heritage-oriented digital repository appropriate to its environment.

1 Introduction

Digital library or digital repository systems [3] are software systems that manage and provide access to digital objects. Some of the most popular digital repository systems are those at universities, where publications and theses are stored and disseminated from open access institutional repositories (such as UPSpace[1]). Other examples that are closer to the public eye are heritage collections accessible through online repositories (such as the Nelson Mandela Foundation's Archives[2]). All such systems store digital items, and offer users access to the items, including through search and browse operations through some form of Web-based interface.

These digital library/repository systems are known widely by different names in different communities. Many Content Management Systems (CMSes, such as Drupal) also offer the ability to manage digital objects as a secondary service, or offer the ability to connect from a website to a repository seamlessly [15]. However, it is a common goal to easily manage digital content and provide access to it. Many repository toolkits have been developed over the last 20 years to help to meet this goal [2]. DSpace[3] [16]was one of the earliest repository toolkits, originally created in partnership between MIT and HP to serve as the basis for MIT's institutional repository. It has since expanded to include far more functionality to support an ever-expanding community. EPrints[4] at the University of Southampton was its precursor and had similar features. Various other similar tools have been created by other organisations, but all have a common Web server+database architecture [2]. In the digital humanities and heritage domain, Atom[5] and Archivematica[6] are popular choices and operate similarly to the other tools on a technical level.

[1]UPSpace; https://repository.up.ac.za/
[2]Nelson Mandela Foundation; https://atom.nelsonmandela.org/
[3]DSpace; https://duraspace.org/dspace/
[4]EPrints; https://www.eprints.org/
[5]Atom; https://www.accesstomemory.org/
[6]Archivematica; https://www.archivematica.org/

While these tools are indeed popular [1] and work well in their environments, they are not as widely adopted in South Africa and other African countries. Where there has been some adoption (of DSpace in particular), it was enabled with extensive training and community support [23]. There are fewer adopters of other toolkits with 28/38 South African repositories on OpenDOAR using DSpace as of April 2019 [12].

There are many reasons for the status quo, but some problems are related to resources and technical limitations. South African public institutions generally do not have trained technical staff to manage digital repository systems. Government funding for archiving of heritage also is limited and many local archives obtain funding from international donors such as the Mellon Foundation [9]. While most repository toolkits subscribe to the OAIS model [22] for digital preservation, a failure of the core software can result in loss of access to an archive if all access is mediated through software. Internet access in poor countries is also not as reliable and fast as elsewhere; and most current repository toolkits only work over the Internet. Finally, maintenance of complex software systems is never a priority of archives, but is required for modern software systems. Popular repository toolkits are complex software systems but many archives are quite small and therefore do not need this complexity. There is therefore a mismatch between feature-laden general-purpose repository toolkits and small archives with few resources to spare.

This paper argues that the technical architecture of current repository toolkits is not ideal for low-resource environments and an alternative architecture specifically designed for low-resource environments may be a more effective solution, especially for the digital humanities and heritage domain where collections are not funded in perpetuity and technical skills and other resources are limited.

In particular, such a solution would aspire to minimalism and a reduction in dependencies, especially on network and data storage, to ensure long term preservation and access with minimal maintenance. The design of such systems is based on a critical analysis of both popular and targeted design strategies that lead to a superset of principles for the design of a modern repository in a low-resource environment.

This analysis and the related principles are presented in this paper, preceded by a discussion of related models and designs, and followed by a discussion of a case study mapped to the identified design principles.

2 Analysis of Prior Work

2.1 Overview of Related Work

One of the earliest works that embodies the notion of a low-resource digital repository was the creation of Project Gutenberg [7], which is a digital repository to distribute out-of-print and free ebooks. Along with the noble goals of supporting education and literacy, Project Gutenberg was designed with the explicit philosophy of minimalism, where every ebook is required to be in the simplest format (often ASCII) to ensure longevity and the widest possible access. The website has evolved over a 50 year period (making it one of the oldest digital repositories in existence) but its content is still available in simple formats, with simple mechanisms for discovery.

Early models for digital repository systems were strongly influenced by the emergence of Web technology and the evolution of software systems from monolithic systems to modular ones. As an example, the Computer Science Teaching Center [10] was a classic database-driven Web application, what was referred to as a LAMP (Linux-Apache-Mysql-Perl) application. The Dienst framework [11] for digital libraries, based on the Kahn-Wilensky framework [8],

attempted to create a generic network-based system, using generalised objects, unique identifiers and service modularisation.

In contrast, the Greenstone digital library software [24] offered a slightly different architecture because it was used to distribute development-oriented information in poor countries. As such, its architecture required minimal services and that it work on the widest possible range of computers. Its traditional focus is, however, on indexing and search, rather than preservation, and its core architecture is arguably based on older Web techniques.

The OpenDL/Diligent [4] and Open Digital Libraries [17] approaches were based on components but the granularity of components often led to complex management systems [21] and this did not appropriately address the needs of low-resource environments. The 5S model [6] and its derivatives provided abstractions of digital library systems, but they did not address operational design.

The Bleek and Lloyd project [18] was a key milestone, as a bespoke demonstrator of a radically different architecture for a digital repository system that was appropriate for low-resource environments. The project demonstrated that core services such as search could be provided without a client-server architecture. This project led to a broader questioning of what the most appropriate digital repository technology and design principles are for African countries, in the context of resource limitations and cultural differences [19]. Some early experimental systems were developed to test different aspects of low-resource repositories. Caljax [20] addressed the question of an offline repository that needed periodic updates and showed that it is possible for an in-browser engine to incorporate a large offline collection and a small online update for a seamless user experience. The Bonolo system [14] [13] used a file-based non-database-driven repository as a general-purpose institutional repository solution.

2.2 Analysis of Design Philosophies

Four design strategies are discussed in detail. These are the design decisions of: the DSpace document repository toolkit; Project Gutenberg's electronic text archive; and the Greenstone digital library system. The simplyCT principles were specifically developed to support low-resource archives, and thus they serve as a baseline. The comparison with other design philosophies aims to determine if there are additional or implicit design goals that should be included; as well as if there are some specific design goals that should be avoided as they inhibit the creation of archives in low-resource environments and/or they are irrelevant to the context of heritage/historical archives.

2.2.1 DSpace

The DSpace design has evolved over many versions, but the core principles remain as they were originally [16], and this includes the following principles.

It should be easy for users to deposit items in the repository as deposit of items is a key function of an institutional repository; this is not necessarily a priority for many heritage collections.

The user interface should be based on modern Web technologies; this has evolved over the years, but is generally based on a Java Servlet container and current approaches to creating customisable and generic User Interfaces.

All items submitted should pass through a workflow management system, with customisable roles, actions and steps; this is especially well-suited to large organisations.

Items should be located in collection and subcollection containers that then belong to top-level community containers.

Items should be assigned globally unique and opaque identifiers on submission, using the Handle system.

2.2.2 Project Gutenberg

The principles underlying the work of Project Gutenberg [7] focus partially on the electronic texts themselves and partially on the underlying software, with some degree of interplay between these concepts and how they relate. The principles include the following.

The format of electronic texts is required to be as simple as possible, typically plain ASCII; users of the texts are permitted to create more advanced versions with markup, but it is argued that plain ASCII will reach more users and will outlast other formats and standards.

In order to get the greatest possible benefit for the largest number of users (99%), the project concentrates on texts and services that most users want; thus the focus is never on completeness of collection or esoteric texts and formats.

The texts are themselves produced at low cost and distributed at low cost so cost is never a factor; this means that the sizes of texts should be small, whatever small is within a specific context and time period.

The texts should be easy to use, refer to, quote, etc.

2.2.3 Greenstone

The key principles stated for Greenstone [24] include the following.

Digital objects of any kind could be imported through the use of configurable and easy-to-create plugins.

The system was designed for use in environments with no Internet through the use of a local Web server that is installed on a user's desktop computer.

Search and browse are provided as distinct services for information discovery.

Collections are pre-indexed and processed to enable services, and this process can be easily repeated when changes are made.

Identifiers for objects are persistent across builds so history and log data is not lost.

The system had versions that would work on any operating system, with some limitations (such as that the CDROM version would only work on Microsoft Windows).

Later versions used a Web Services design strategy, where services are located by network service endpoints.

2.2.4 simplyCT

This set of design principles was derived and generalised from the Bleek and Lloyd project [18] and include minimalism, no imposition on users, no API, no Web, simple preservation, any object support, service repeatability and support for superimposed information.

Only the bare minimum amount of system infrastructure should be needed, to lower the cost of maintenance.

Users should not be expected to conform to system requirements for formats, identifiers, metadata, structure, etc.

File access should be a primary form of accessing data, rather than a Web Services interface.

The data should be accessible whether or not there is a network/Web available, whether through a mediated interface or directly through the filesystem.

Preservation should be as simple as possible, by copying files and directories rather than using a database.

Table 1: List of design principles that support archives in low-resource environments. Y means that it is a design principle while N means that it is not. Codes within parentheses are principles that are implied but not necessarily explicitly stated.

Principle	**DSpace**	**Greenstone**	**Gutenberg**	**SimplyCT**
Minimalism	(N)	(N)	Y	Y
Do not impose on users	(N)	(Y)	(N)	Y
Web or No Web	(N)	Y	Y	Y
No API	(N)	(N)	Y	Y
Simple Preservation, Low Cost	(N)	(N)	Y	Y
Any Objects	(Y)	Y	(N)	Y
Everything is repeatable/flexible	(N)	(N)	–	Y
Superimposed information	N	N	–	Y
Hierarchical organisation	Y	Y	Y	(Y)
Platform agnostic	(Y/N)	Y/N	–	Y
Collection building	N	Y	Y	(Y)

Any metadata and digital objects should be supported.

Services should be generic and not strongly tied to a particular segment of the data.

Finally, additional services can be layered on top of a basic archive, thus providing more information in layers rather than complex linked objects.

2.3 Designing for Low Resource Environments

A comparison and merging of the various principles discussed in the previous section is presented in Table 1.

Minimalism is explicit in models that aim for simplicity, but not in the other models, where advanced software architectures are adopted for currency. *Imposition on users* is an issue in DSpace and Gutenberg, where users who submit items must conform to the specified formats and structures. Regarding *No Internet* as a requirement, DSpace does not work except as an Internet service. *Simple Preservation* is not possible in DSpace, where data for preservation/rescue/backup must be accessed through an API. Most systems allow *Any Objects*. Most systems do not support *flexibility* in data architecture, so the use cases are limited. *Superimposed information* is therefore also not possible in those cases. All designs support *hierarchical organisation* of information. In terms of *platform agnostic* operations, DSpace and Greenstone support multiple platforms but not as widely as could be possible. Finally, *collection building* is only possible for designs that allow non-online operation, hence this excludes DSpace.

Design principles that are equally applicable in low-resource and high-resource archives include: standards-compliance, search and browse as features and persistent identifiers.

Some of the principles that guided the design of DSpace in particular have no equivalence for heritage collections and can be argued to be inhibitors in low-resource environments, such as generic Workflow support, which introduces complexity and is often not needed.

This analysis illustrates that a superset of explicit design principles to support low-resource environments is desirable. It also highlights the shortcomings of systems that were designed for other purposes and that do not explicitly take into account the needs of low-resource environments. It is not that these systems are not useful in other contexts; it is just that their design

is less than optimal in the low-resource context.

The Five Hundred Year Archive prototype was therefore designed to conform to the identified set of low-resource archive design principles, as discussed in the following section.

3 Case Study: Five Hundred Year Archive

The Five Hundred Year Archive (FHYA) [5] is based around a collection of digital objects and associated metadata that are related to pre-colonial history in South Africa. This project is based at the Archives and Public Culture Centre at the University of Cape Town. Much effort has gone into content curation, selection, organisation and creation of metadata. The final step is the creation of a digital repository that will enable user access and engagement with the archive.

Given that the archive is hosted within a low-resource environment, with a particular shortage of technical skills and funding for ongoing maintenance, a low-resource design is being explored for the digital repository.

The repository system is centred on a set of directories containing digital objects (data/) and metadata (metadata/) . There is then a script that ingests a spreadsheet of metadata by converting the metadata into individual XML files in the metadata directory. A second script then generates an HTML representation of each metadata file using an XML Stylesheet Transformation, with thumbnails and pointers to the digital object; the thumbnails are incorporated into a Javascript slider to display a series of images within a fixed space. An indexing system creates inverted files to support searching and browsing via a Javascript-based faceted search engine. All of these operations result in a static website with the full functionality of a read-only archive.

Changes to the collection, such as the addition of comments, are implemented as Web scripts that store the comment, then regenerate the pages for the digital object and the user. Thus, the addition of the comment makes a change to the static repository and there is no dynamic page generation needed.

The technologies underlying the system include: a scripting language optimised for text processing (Perl/Python); XML for storing structured data in flat files; stylesheets for transformation of XML (XSLT) and styling of HTML pages (CSS); and a simple Web server to serve pages online for those cases where online access is needed.

Each of the identified principles from the previous section can be mapped to the system design decisions, as outlined below:

- **Minimalism.** The explicit aim is to have as little software as possible, with short scripts instead of multi-layered abstractions or agent-based or Web Service-based architectures.
- **Imposition on users.** Metadata already exists in spreadsheets so this is used for batch import. Items already contain assigned identifiers and these have meaning to users so these are reused. Similarly, the pre-defined structure of the collections is maintained.
- **No Internet.** All read-only access happens through direct access to HTML files, without any mediation through a Web application. Search and browse services are provided through Javascript applications that read static files. Only contributions to the archive require the use of an online version of the archive and an Internet connection; otherwise the read-only archive can be distributed on offline media.
- **Simple Preservation.** All data objects are in their original locations and metadata files are created as simple XML files, with associated HTML files generated by a stylesheet

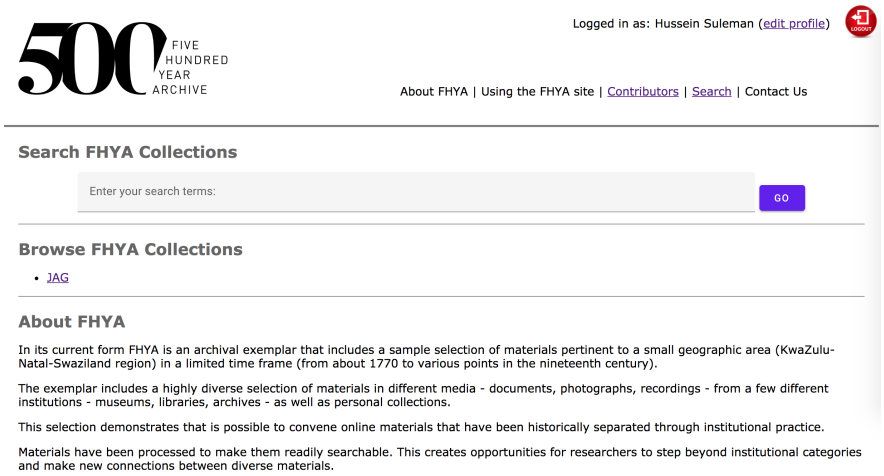

Figure 1: Screen snapshot of front page of user interface.

transformation. Preserving the original data and metadata requires only copying of the files, while migration can be achieved through transformation of the metadata files.

- **Any Objects.** Any data file format accessible through a Web browser is supported; effectively this is any object since a Web browser will allow a user to download any non-viewable file.

- **Flexibility.** Search and browse services can be defined to operate over any collection of metadata. User accounts are stored as a metadata collection, and these can therefore be searched and browsed using the same tools as the objects themselves.

- **Superimposed information.** Comments on objects are stored as a superimposed layer of information, in a separate file directory structure, which does not disturb the original objects. When a view of the object is being updated, the comments are drawn into and added to the view, thus creating a seamless experience for the user.

- **Hierarchical organisation.** All data and metadata is stored in hierarchies of directories and sub-directories.

- **Platform agnostic.** Since the system is composed of flat files and HTML files, these can be read on any operating system where a Web browser is available. Thus, this is even accessible on all tablets and phones, irrespective of operating system, with no changes to the data or system necessary.

- **Collection building.** All data is pre-processed whenever new items are ingested. Thus, the system is always in a static state with the fastest possible access to items.

Figure 1 shows the front page of the system, Figure 2 shows a page of search results in the Javascript-based search and browse interface and Figure 3 shows a typical digital object page with thumbnails and metadata.

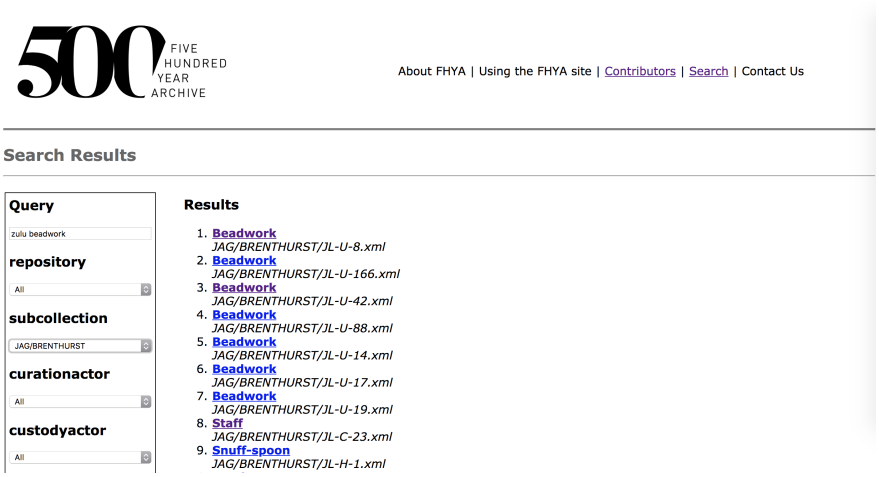

Figure 2: Screen snapshot of search/browse interface.

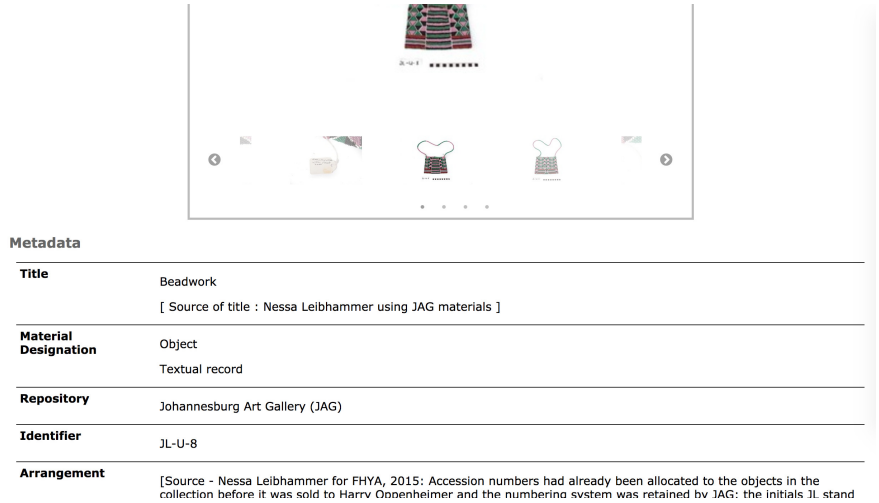

Figure 3: Screen snapshot of metadata and object display page.

4 Conclusions and Future Work

This paper has discussed design principles from prior work that inform the creation of digital repositories and their applicability in low-resource environments. Arguably, many of the principles adopted by current systems violate one or more requirements and environmental conditions, such as the lack of technical skills and resources. An alternate low-resource design is illustrated in the context of the Five Hundred Year Archive, to demonstrate the feasibility of such principles.

Low-resource repositories have very broad applicability. Lowering the barrier to the creation of heritage and historical archives will benefit everyone, not just those in traditional low-resource

environments. It can also enable participation on an individual level through the creation of personal and community archives, which is still an elusive goal around the world.

Current and future work includes the creation of additional services, derivation of a reusable toolkit and the application of this toolkit to other use cases to demonstrate its flexibility. While it was deemed out of scope for the current project to consider software maintenance, current approaches to software maintenance can also be investigated in future.

The design of low-resource digital repositories explicitly allows for easier migration to other software systems, which are not necessarily designed for low-resource environments. However, while some may think this is the goal, this is arguably a step in the wrong direction as it would take collections that are easily preserved and destroy the very attributes of simplicity that make the collections preservable.

5 Acknowledgments

Thanks go to the curators of the Five Hundred Year Archive, who have collaborated in the creation of this case study.

This research was partially funded by the National Research Foundation of South Africa (Grant numbers: 85470 and 105862) and University of Cape Town. The authors acknowledge that opinions, findings and conclusions or recommendations expressed in this publication are that of the authors, and that the NRF accepts no liability whatsoever in this regard.

References

[1] Mufazil Ali, Fayaz Ahmad Loan, and Rabiya Mushatq. Open access scientific digital repositories: An analytical study of the open doar. In *2018 5th International Symposium on Emerging Trends and Technologies in Libraries and Information Services (ETTLIS)*, pages 213–216. IEEE, 2018.

[2] Mathieu Andro, Emmanuelle Asselin, and Marc Maisonneuve. Digital libraries: Comparison of 10 software. *Library Collections, Acquisitions, and Technical Services*, 36(3-4):79–83, 2012.

[3] William Y Arms, Christophe Blanchi, and Edward A Overly. An architecture for information in digital libraries. *D-lib magazine*, 3(2), 1997.

[4] Leonardo Candela, Fuat Akal, Henri Avancini, Donatella Castelli, Luigi Fusco, Veronica Guidetti, Christoph Langguth, Andrea Manzi, Pasquale Pagano, Heiko Schuldt, et al. Diligent: integrating digital library and grid technologies for a new earth observation research infrastructure. *International Journal on Digital Libraries*, 7(1-2):59–80, 2007.

[5] Archive & Public Culture. The five hundred year archive.

[6] Marcos André Gonçalves, Edward A. Fox, Layne T. Watson, and Neill A. Kipp. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM Trans. Inf. Syst.*, 22(2):270–312, April 2004.

[7] Michael Hart. The history and philosophy of project gutenberg. *Project Gutenberg*, 3:1–11, 1992.

[8] Robert Kahn and Robert Wilensky. A framework for distributed digital object services. *International Journal on Digital Libraries*, 6(2):115–123, 2006.

[9] Trywell Kalusopa. Preservation and access to digital materials: Strategic policy options for africa. In *Handbook of Research on Heritage Management and Preservation*, pages 150–174. IGI Global, 2018.

[10] Deborah L Knox. The computer science teaching center. *ACM SIGCSE Bulletin*, 31(2):22–23, 1999.

[11] Carl Lagoze and James R Davis. Dienst: an architecture for distributed document libraries. *Communications of the ACM*, 38(4):47, 1995.

[12] OpenDOAR. The directory of open access repositories, 2019.

[13] Lighton Phiri and Hussein Suleman. Flexible design for simple digital library tools and services. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, SAICSIT '13, pages 160–169, New York, NY, USA, 2013. ACM.

[14] Lighton Phiri, Kyle Williams, Miles Robinson, Stuart Hammar, and Hussein Suleman. Bonolo: A general digital library system for file-based collections. In *International Conference on Asian Digital Libraries*, pages 49–58. Springer, 2012.

[15] Nick Ruest and Kirsta Stapelfeldt. Introduction to islandora, 2014.

[16] MacKenzie Smith, Mary Barton, Mick Bass, Margret Branschofsky, Greg McClellan, Dave Stuve, Robert Tansley, and Julie Harford Walker. An open source dynamic digital repository. *D-Lib Magazine*, 9(1):1082–9873, 2003.

[17] Hussein Suleman. *Open Digital Libraries*. PhD thesis, Virginia Tech, 2002.

[18] Hussein Suleman. Digital libraries without databases: The bleek and lloyd collection. In *International Conference on Theory and Practice of Digital Libraries*, pages 392–403. Springer, 2007.

[19] Hussein Suleman. An african perspective on digital preservation. In *Multimedia Information Extraction And Digital Heritage Preservation*, pages 295–306. World Scientific, 2011.

[20] Hussein Suleman, Marc Bowes, Matthew Hirst, and Suraj Subrun. Hybrid online-offline digital collections. In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, SAICSIT '10, pages 421–425, New York, NY, USA, 2010. ACM.

[21] Hussein Suleman and Siyabonga Mhlongo. A flexible approach to web component packaging. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 257–266. South African Institute for Computer Scientists and Information Technologists, ACM, 2006.

[22] Robert Tansley, Mick Bass, and MacKenzie Smith. Dspace as an open archival information system: Current status and future directions. In *International Conference on Theory and Practice of Digital Libraries*, pages 446–460. Springer, 2003.

[23] Martha Johanna Van Deventer and Heila Pienaar. South african repositories: Bridging knowledge divides.

[24] Ian H Witten, Rodger J McNab, Stefan J Boddie, and David Bainbridge. Greenstone: a comprehensive open-source digital library software system. In *Proceedings of the Fifth ACM Conference on Digital libraries*, pages 113–121. ACM, 2000.