A Literature Review on Institutional Repositories, DSpace and migrating to DSpace from legacy systems

DARRYL MEYER, University of Cape Town

Abstract

Research intuitions are adopting institutional repositories to collect and preserve their intellectual works. This literature review looks at institutional repositories and digital repositories, migrating from a legacy database system and methodologies seen to be effective for migrations from legacy systems. A description of the DSpace digital repository and different ways of importing data into DSpace is discussed, ending with a look at case studies from other institutions that have adopted institutional repositories.

• Applied computing~Digital libraries and archives

Additional Key Words and Phrases: Institutional Repositories, Legacy database systems migration, DSpace

1. INTRODUCTION

We are to migrate two legacy database systems for the National Research Foundation (NRF) to the DSpace digital repository system. A complaint of the NRF is that the interfaces to manage the legacy databases are time-consuming to use. By migrating to DSpace, the NRF hopes to reduce the amount of time it take to deposit data into the databases. To better understand a migration from a legacy database system to the DSpace digital repository system, it is necessary to first understand the topics of repositories and what is involved in such a migration. The topics analysed in the literature review begin with digital repositories, which is the technology behind DSpace, then move on to the specialization of institutional repositories. Following that is a closer review of the workflow and technical details of DSpace and migrations from legacy database systems and the methodologies found to work effectively and efficiently. Then a look at methods of importing into DSpace and case studies from other institutions that have setup and migrated to DSpace.

2. DIGITAL REPOSITORIES

A Digital Library (DL) can be thought of in terms of its physical counterpart, with the exception that a DL can include things that a physical library cannot, such as algorithms, programs and other digital works that can not be easily expressed on paper [Fox and Sornil 2003]. Following from DLs, a digital repository is a collection of digital objects but, as proposed by Heery and Anderson [2005], differs from other digital collections in that the architecture of the repository allows for the management of the content as well as metadata related to the content and that content can be deposited into the repository by distributed third parties [Heery and Anderson 2005]. Allowing third parties to deposit content to the repository is of particular importance to the NRF because they want to enable universities to deposit to the repository directly.

The definition of an Open Access (OA) digital resource is that it is available online, free of charge and free of most copyright and licensing restrictions [Suber 2007]. For a repository to be categorised as an OA repository it is necessary for that repository to allow OA to the content that it stores, provided there are no legal constraints on

Author's email address: myrdar003@myuct.ac.za

the content, such as in the case of an embargo, and the repository must allow OA to the metadata of the content in such a way that it allows harvesting of the metadata [Heery and Anderson 2005].

3. INSTITUTIONAL REPOSITORIES

An institutional repository (IR) is a DL, in that they both collect and preserve digital works, but an IR is specialized to preserve and disseminate the intellectual output of a university or research institution and members of its community [Adewumi and Omoregbe 2011]. Lynch [2003] observes an IR as a set of services that are made available by the institution for its community members to enable storage and dissemination of their intellectual output.

Traditionally the intellectual output of an institution tends to be sent off to different, subject specific publishers. This means that the work of an institution can become dispersed across different publishers and different library collections. An IR, on the other hand, allows an institution to store its intellectual output on a platform that encourages information sharing and is also an effective means of displaying the academic quality of the institution [Yeates 2003]. It is the responsibility of universities and research institutions to preserve and make their intellectual output available; these responsibilities are both provided for by setting up an IR [Lynch 2003].

3.1 Institutional Repository features and specifications

There are both proprietary or open source offerings of IR software; DSpace is an example of an open source solution. IRs are also available as software or as a hosted service; however, the software solution is by far the most popular option available as many of the institutions setting up IRs have the necessary requirements to run an IR locally [Adewumi and Omoregbe 2011].

In an IR the information describing the digital content is called metadata; this term is, however, not unique to IRs. In order to enable interoperability between the IRs of different institutions for the harvesting of content, IRs need to support metadata standards, the most popular of which is the Dublin Core standard [Smith et al. 2003]. Another mechanism for describing and packaging information about digital objects is the Metadata Encoding and Transmission Standard (METS) [Cundiff 2004]. METS is a metadata standard that packages together the descriptive, administrative and object references of a digital resource, whereas Dublin Core is a set of standard fields to describe a digital resource [Hillmann 2003]. A standardised metadata format allows for the content in the IR to be harvested, thus enabling support for the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [Open Archives 2002]. "Harvesting" is the term given to the ability to programmatically retrieve metadata from several remote sources and collate the information back into the system doing the harvesting [Tansley et al. 2003]. DSpace attempts to improve metadata consistency and completeness by simplifying the metadata-creation process by including mandatory and automatically completed fields [Kurtz 2010]. This effort to improve consistency and completeness is valuable when harvesting content from other institutions' IRs to ensure that content harvested has the same level of completeness and consistency as the metadata in the IR doing the harvesting.

In an OA IR anyone can view the content on the IR, but not all IRs are OA and some may require user authentication. It is necessary for the person to be depositing information to the IR to be authenticated to ensure that the content stored in the IR is valid and true [Adewumi and Omoregbe 2011]. This is a particularly important feature for institutions, such as the NRF, where unauthenticated depositing to the IR can lead to somebody claiming they have a qualification that they have not earned. IRs support multiple depositing roles and an owner role that can validate all submissions before they are made public. This functionality will be used by the NRF to validate submissions before they are made public.

3.2 Implementation considerations for an Institutional Repository

Implementing an IR, such as DSpace, into an environment that does not already have an IR requires steps to be taken to ensure the correct installation and customization of the IR [Wang 2011]. Such steps include initial installation, training staff and users, ensuring financial sustainability of the IR, ensuring the submission process is part of the Libraries' submission policy and management of the IR [Baudoin and Branschofsky 2003]. For an institution that already has a previous IR installation, the implementation process is much simpler. This is the case with this Honours Project, where the NRF already has one instance of DSpace running and therefore the implementation steps such as ensuring the required hardware and software capabilities for DSpace, training staff and users and ensuring financial sustainability have all previously been dealt with.

4. DSPACE DIGITAL REPOSITORY

DSpace is an open source digital repository system that supports the capture of digital works, distribution of those works over the Internet through a search and retrieval system and ultimately the preservation of those works [Yeates 2003]. DSpace was developed by collaboration between Massachusetts Institute of Technology (MIT) Libraries and Hewlett-Packard Labs. DSpace was designed to mainly serve as a repository for digital research and teaching materials produced by a university or research institution; it is however customisable and can thus be adopted for other digital storage needs [Smith et al. 2003]. DSpace was created to solve a need for a system that could collect, index and distribute the increasing amount of complex digital publications produced by universities and research institutions.

4.1 DSpace submission workflow

Depositing digital works into DSpace is made simple by introducing the idea of "communities." Communities are the sub-units of an organisation, so for a university communities could be departments, labs, or different centres of research [Smith et al. 2003]. DSpace enables each community to customise the DSpace systems to suit their needs and submission process. DSpace is the first open source solution to tackle the complex problem of allowing different submission workflows for different communities because different communities have different restrictions on the digital works they submit, such as by whom, who needs to approve the material and to what collections can they submit. Collections in DSpace allow for the separation of different types or topics of content within a community. Smith et al. [2003] gives the example of two collections: one for working papers; and the other for associated datasets. DSpace allows for different roles within the system, such as "submitters " and "reviewers," which may be the case in a setup of the DSpace system where there are people who submit content then others are able to review the content before it is entered into a specific collection [Garfinkel 2005].

4.2 Content discovery in DSpace

The ability to search and browse through content is an essential component of digital repositories [Tansley et al. 2003]. All content submitted to a DSpace repository is indexed, this enables easy searching and browsing capability. By design, DSpace provides many search features to help users find content easily. Searching for content in DSpace can be done by searching the entire repository for a query term or can be limited to communities or collections within communities. Also important for content discovery is the ability to browse for content. DSpace provides the ability to browse by different properties of the content such as by author, date or title or browse by community or by collection [Tansley et al. 2003].

4.3 DSpace Web User Interface

The Web User Interface for DSpace is built on Java Servlet and Java Server Page technology and provides an interface to the functionality of DSpace through a Web browser [Tansley et al. 2003]. Through their Web browser, users are able to search for and browse content, view collection and community home pages, view individual items pages and view help pages. Authorized users are able to submit content through their web browser and there is an administration section where administrative functions can be preformed. The use of Java Server Pages means that the source code provided by DSpace can be easily customized to the fit the branding of the institution without changing any of the business logic code [Tansley et al. 2003].

4.4 DSpace technical and architectural platform

DSpace was developed to be open source and also makes use of other open source libraries and tools [Smith et al. 2003]. It runs on the UNIX platform, incorporates a relational database management system (PostgreSQL or Oracle), a Web server and Java servlet engine (Apache Tomcat). These characteristics make DSpace easy to be adopted by institutions with minimal resources.

DSpace was built to support customization and enhancement; it thus was designed with a familiar architecture in mind—a layered architecture. DSpace has three layers and each layer consists of different components [Tansley et al. 2003]. The first layer—the storage layer—uses the file system to store content and metadata and is managed by the relational database management system. The storage layer also contains a bitstream storage manager that is responsible for storing bitstreams in a file system. The second layer—the business logic layer—includes the managerial functionality of DSpace, the administration, workflow, search and retrieval and authorization. The last layer—the application layer—houses: the Web user interface and batch submission loader as well as other functionality to communicate with systems outside of the DSpace installation; the Handle server, used for resolving persistent identifiers for the content stored in the repository; and Open Archives Initiative (OAI) support, which allows interoperability between DSpace adopters [Smith et al. 2003, Tansley et al. 2003]. In Figure 1, created by Tansley et al. [2003], separation of components into the different layers is depicted.

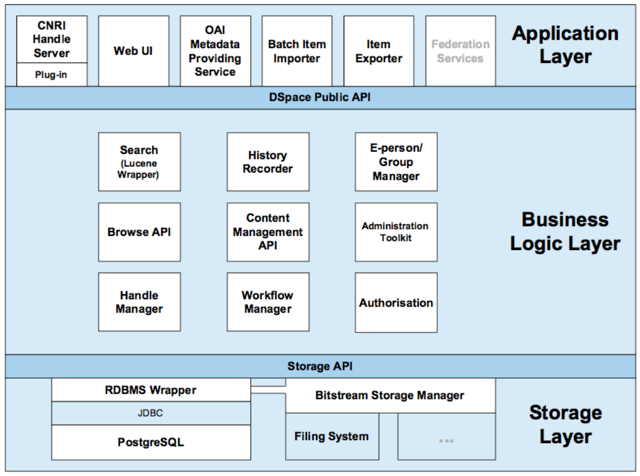

Figure 1: DSpace System Architecture

Each component has a clearly defined API and can be replaced with another component providing the same functionality as required by the system, but possibly allowing additional features to be incorporated, without requiring extensive modification to the remaining components [Tansley et al. 2003].

5. MIGRATION FROM A LEGACY DATABASE

Although migrating to an institutional repository, such as DSpace, is not the same as migrating from one database management system to another, there are lessons to be learned from understanding efficient and effective database migrations strategies. Furthermore legacy systems are usually critical to daily operations and therefore decommissioning them is not an option to many institutions [Wu et al. 1997a].

5.1 Legacy database migration methodologies and strategies

A simple, naïve approach to migrating a legacy database system is to rebuild the entire legacy system using modern tools and database software. The risk of failure is high and the time consuming nature of such an approach makes it infeasible [Bisbal et al. 1997]. Brodie and Stonebraker [[1995], as cited by [Bisbal et al. 1997]] explained three other legacy database migration methodologies, the Forward Migration Method, the Reverse Migration Method and the Composite Database Approach. The Forward Migration Method involves first migrating the legacy data to a modern database system then, following that, migrating the legacy applications and interfaces. The Reverse Migration Method is essentially the opposite of the Forward Migration Method. First the legacy applications and interfaces are migrated, and then the legacy data is migrated to a modern database system. The Composite Database Approach combines functionality from both the Forward and Reverse Migration Methods. During the migration process the legacy and modern system form a composite information system. A co-ordinator system is used to decide which database, the legacy or modern, to access when an update is requested.

In a case study observed by Razavian and Lago [2014] in which two large banks were merged, what was found to be effective was an incremental migration approach. The large volume of data to be migrated and the complex relationships of data in the database motivated an incremental approach. The migration was broken up into 16 increments and could only be completed on weekends. An incremental approach, such as this one, may not be necessary for the NRF database migration because their database may be much smaller but the data in the database is grouped by current works and completed works. An interest of first migrating current works was expressed by the NRF so lessons learnt from this case study may prove to be useful for us.

The main focus of Razavain and Lago's [2014] paper was, however, to introduce a "Lean and Mean" migration strategy. The essence of the Lean and Mean strategy is to separate the parts of the migration process that are common to other migrations, called core elements, from the project-specific parts. Having an understanding of the core elements enables you to make informed decisions about what should be the driving factors of the migration and what activities are best to be performed on those factors.

Wu et al. [1997b] developed the Butterfly methodology to be a simple, fast and safe way to migrate legacy systems to modern target systems without the need of simultaneously accessing both the legacy and target systems. This approach can be seen as a direct counter to the Composite Database Approach mentioned previously, which is inherently complex due to the fact that updates happen to both legacy and target systems simultaneously, needing a co-ordinating sub-system to manage the updates. With the Butterfly methodology the legacy systems remains in production while being migrated to the target system and only once all the legacy data is completely migrated to the target system does the target system become the production system. The Butterfly methodology presents a migration engine that allows the legacy system to be shut down for a minimal amount of time, which is necessary for critical systems that cannot be decommissioned for a long period of time while the migration takes place.

5.2 Legacy application migration strategies

Legacy systems are usually highly customized to suit the business processes and needs of the institution. This is because over the years the institution has adapted to change the legacy system as their business processes change. Cimitile et al. [1999] explain that the migration of legacy systems includes reverse engineering and decomposing the legacy system into its constituents, similar to the Lean and Mean core elements idea as mentioned previously.

6. MIGRATION TO DSPACE

DSpace has a built-in command line feature that allows batch loading of existing collections into the system [Baudoin and Branschofsky 2003]. This is seen as a primitive method for importing multiple items into DSpace as imports are done from a directory on the file system that is in a format recognised by DSpace. It is for this reason that those tasked with importing multiple items into DSpace have often used a custom script to format the existing collections so that they can then be imported into DSpace [Walsh 2010].

6.1 Importing into DSpace

One way to import existing data into DSpace is as The Ohio State University did; they created custom Perl scripts to format existing data so that it can be imported using DSpace's command line tool [Walsh 2010]. The legacy data that they were formatting was in the form of comma separated value files or spreadsheet documents and had to be XML files with metadata in the Dublin Core format with different items in individual folders. The scripts were created on a case-by-case basis for the different import needs and sources but followed the same functionality and improved with accuracy over time. Although batch formatting greatly reduces the time it takes to import content into DSpace the quality of the metadata is directly dependent on the quality of the existing data.

Another way of importing data into DSpace is using the Simple Web-service Offering Repository Deposit (SWORD) protocol [Allinson et al. 2008]. SWORD facilitates depositing items into a repository and the deposits can be done remotely without having to use the repository' native user interface, which is in this case DSpace's command line importer. In cases where the DSpace deposit system has not been agile enough but users still want to use the deposit checks and workflows of DSpace, SWORD can be used to achieve a more agile deposit system while maintaining the formal checks and workflows of DSpace [Lewis et al. 2009].

A way of transferring data between two repositories is using the Open Archive Initiative's Object Reuse and Exchange (OAI-ORE) standards in combination with OAI-PMH, mentioned previously in section 3.1, OAI-PMH being a metadata discovery protocol and OAI-ORE being a set of standards for describing and exchanging digital resources [Open Archives 2008]. In order for this interchange of content between repositories to work, the digital resources have to be able to be harvested, which requires each digital resource to have a URI. Although this allows for an automatic migration from one repository to another, this approach may not be viable as the legacy system may not support the requirements of OAI-ORE and may not implement OAI-PMH [Maslov et al. 2010].

The National Library of Australia created the Australian METS Profile and registered it with the Library of Congress [Library of Congress 2007]. The Australian METS profile describes how METS can be used to collect digital resources in repositories. As an example of this, it was used for the interchanging of data between a DSpace implementation at the Australian National University and a Fez-Fedora implementation at the University of Queensland [Pearce et al. 2008].

6.2 Case studies of installation and migration to DSpace

The Dhananjayarao Gadgil Library of Gokhale Institute of Politics and Economics (GIPE) chose to use DSpace to store their valuable rare book collection. Their installation of DSpace runs off a LiveCD bundled software package called LibLiveCD, which includes Ubuntu, DSpace and all its dependencies [Shewale 2012]. The advantage of an installation like theirs is that it is easy to setup as all the software and dependencies required are prepackaged into a LiveCD. The disadvantages of this approach, however, are that the LibLiveCD has not been updated recently and the highest supported version of DSpace in LibLiveCD is 1.7.0 whereas the latest stable version of DSpace is 5.1 [Dspace 2015, Prasad and Barve 2015]. Their initial process to deposit content to DSpace was to do it manually. This approach was time consuming and error prone so they found a Perl script written by Prof ARD Prasad that was able to perform batch uploads of both the metadata and bit-streams of the digital scans of the books [Shewale 2012].

The Texas Tech University School of Law installed DSpace to collect, share and promote the law school's digital materials; it was the law schools first institutional repository [Wang 2011]. Their install is useful to understand because the staff tasked to install DSpace had little prior understanding of DSpace or other institutional repositories before attempting to install DSpace. They had to go through the task of purchasing an operating system on which DSpace could run; they chose Red Hat enterprise Linux because it is recommended for DSpace. They found the installation difficult because it requires a lot of expertise and found an alternative option that simplifies the implementation process called JumpBox for DSpace Open Source Repository [JumpBox 2013]. This option provides a disk image that comes packaged with DSpace, the Ubuntu operating system and all its required dependencies. The disk image can then be loaded as a virtual machine. This option is a convenient and easy installation option but JumpBox is no longer maintaining support for this DSpace package and thus the latest release only include DSpace version 3.1. The law school ultimately resorted to contracting a company that could install DSpace for them; all they did was design the look and feel of the user interface for the DSpace installation [Wang 2011].

6.3 Migrating to other Institutional Repositories

The Lamar Soutter Library at the University of Massachusetts Medical School migrated their image database to the Digital Commons IR software, after comparing Digital Commons to DSpace and Open Repository, a hosted service based off of DSpace [Piorun et al. 2007]. Their choice was motivated by the Library staff's lack of technical expertise, as they did not have access to on-campus technical personnel, so installing an IR themselves would not be feasible. The Library also did not have the resources that a product such as DSpace required. They chose not to go for Open Repository, as it was too new and untested for their needs. They needed an IR software solution that could be installed and configured by the vendor. That solution was ProQuest from Digital Commons.

7. CONCLUSIONS

Moving forward from this literature review we will be deciding the best approach to use to implement the DSpace digital repository at the NRF and how we should import the data from the legacy database system to DSpace. Having an understanding of methodologies used when migrating from legacy systems will better enable us to migrate the data efficiently whilst ensuring minimal down time for the legacy system during the migration. Understanding the concepts of metadata and the integral role it plays with the consistency and accuracy of the content in digital repositories will further enable us to ensure the legacy data migrated to DSpace is consistent and complete.

REFERENCES

Edward A. Fox and Ohm Sornil. 2003. Digital libraries. In *Encyclopedia of Computer Science,* Anthony Ralston, Edwin D. Reilly and David Hemmendinger, Eds. John Wiley and Sons Ltd., Chichester, UK, 576-581.

Rachel Heery and Sheila Anderson. 2005. Digital repositories review. Joint Information Systems Committee. Retrieved 18 April, 2015 from http://opus.bath.ac.uk/23566/2/digital-repositories-review-2005.pdf

Peter Suber. 2007. Open access overview. Focusing on open access to peer-reviewed research articles and their preprints. (2007). Retrieved April, 2015 from http://legacy.earlham.edu/~peters/fos/overview.htm#journals

Adewole Adewumi and Nicholas Omoregbe. 2011. Institutional repositories: features, architecture, design and implementation technologies. *Journal of Computing* 2, 8.

Clifford A. Lynch. 2003. Institutional repositories: essential infrastructure for scholarship in the digital age. *portal: Libraries and the Academy* 3, 2, 327-336.

Robin Yeates. 2003. Institutional repositories. *Vine* 33, 2, 96-101.

MacKenzie Smith, Mary Barton, Mick Bass, Margret Branschofsky, Greg McClellan, Dave Stuve, Robert Tansley and Julie H. Walker. 2003. DSpace: An open source dynamic digital repository. *D-Lib Magazine* 9, 1.

Morgan V. Cundiff. 2004. An introduction to the metadata encoding and transmission standard (METS). *Library Hi Tech* 22, 1, 52-64.

Diane Hillmann. 2003. Using Dublin Core-Dublin Core Qualifiers. (2003). Retrieved April 19, 2015 from http://dublincore.org/documents/2003/08/26/usageguide/qualifiers.shtml

Open Archives. 2002. Open Archives Initiative - Protocol for Metadata Harvesting - v.2.0. (2015). Retrieved April, 19 2015, from http://www.openarchives.org/OAI/openarchivesprotocol.html

Robert Tansley, Mick Bass, David Stuve, Margret Branschofsky, Daniel Chudnov, Greg McClellan and MacKenzie Smith. 2003. The DSpace institutional digital repository system: current functionality. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries,* IEEE Computer Society, 87-97.

Mary Kurtz. 2010. Dublin Core, DSpace, and a Brief Analysis of Three University Repositories. *Information Technology & Libraries* 29, 1, 40-46.

Fang Wang. 2011. Building an Open Source Institutional Repository at a Small Law School Library: Is it Realistic or Unattainable? *Information Technology & Libraries* 30, 2, 81-84.

Patsy Baudoin and Margret Branschofsky. 2003. Implementing an Institutional Repository: The DSpace Experience at MIT. *Science & Technology Libraries* 24, 1, 31-46.

Simson Garfinkel. 2005. MIT's DSpace Explained. *Technology review* 108, 7, 50-50.

Bing Wu, Deirdre Lawless, Jesus Bisbal, Jane Grimson, Vincent Wade, Donie O'Sullivan and Ray Richardson. 1997a. Legacy system migration: A legacy data migration engine. In *Proceedings of the 17th International Database Conference (DATASEM'97),* 129-138.

Jesus Bisbal, Deirdre Lawless, Bing Wu, Jane Grimson, Vincent Wade, Ray Richardson and Donie O'Sullivan. 1997. An overview of legacy information system migration. In *Software Engineering Conference, 1997. Asia Pacific... and International Computer Science Conference 1997. APSEC'97 and ICSC'97. Proceedings,* IEEE, 529-530.

Michael L. Brodie and Michael Stonebraker. 1995. Migrating legacy systems: gateways, interfaces & the incremental approach. Morgan Kaufmann Publishers Inc.

Maryam Razavian and Patricia Lago. 2014. A lean and mean strategy: a data migration industrial study. *Journal of Software: Evolution & Process* 26, 2, 141-171.

Bing Wu, Deirdre Lawless, Jesus Bisbal, Ray Richardson, Jane Grimson, Vincent Wade and Donie O'Sullivan. 1997b. The butterfly methodology: A gateway-free approach for migrating legacy information systems. In *Third IEEE International Conference on Engineering of Complex Computer Systems, 1997.* IEEE, 200-205.

Aniello Cimitile, René R. Klösch and Hausi Müller. 1999. Guest Editors' Introduction: Migration Strategies for Legacy Systems. *International Journal of Software Engineering & Knowledge Engineering* 9, 1, 1-3.

Maureen P. Walsh. 2010. Batch Loading Collections into DSpace: Using Perl Scripts for Automation and Quality Control. *Information Technology & Libraries* 29, 3, 117-127.

Julie Allinson, Sebastien François and Stuart Lewis. 2008. Sword: Simple web-service offering repository deposit. In JISC CETIS EC and MDR SIG meeting, (Strathclyde University, Glasgow, 2007).

Stuart Lewis, Leonie Hayes, Vanessa Newton-Wade, Antony Corfield, Richard Davis, Tim Donohue and Scott Wilson. 2009. If SWORD is the answer, what is the question? Use of the Simple Web-service Offering Repository Deposit protocol. *Program* 43, 4, 407-418.

Open Archives. 2008. Open Archives Initiative - Object Reuse and Exchange. (2008). Retrieved April, 19 2015, from http://www.openarchives.org/ore/1.0/primer

Alexey Maslov, James Creel, Adam Mikeal, Scott Phillips, John Leggett and Mark McFarland. 2010. Adding OAI-ORE support to repository platforms. *Journal of digital information* 11, 1.

Library of Congress. 2007. Australian METS Profile 1.0: Metadata Encoding and Transmission Standard (METS). (2007). Retrieved 19 April, 2015 from http://www.loc.gov/standards/mets/profiles/00000018.html

Judith Pearce, David Pearson, Megan Williams and Scott Yeadon. 2008. The Australian METS Profile–A journey about metadata. *D-Lib Magazine* 14, 3.

Nanaji Shewale. 2012. Building Digital Library using DSpace: Case Study of GIPE's Dhananjayarao Gadgil Digital Library. *DESIDOC Journal of Library & Information Technology* 32, 5, 417-420.

Dspace. 2015. Latest Release | DSpace. (2015). Retrieved 18 April, 2015 from http://www.dspace.org/latest-release

ARD Prasad and S. Barve. 2015. LibLiveCD. (2015). Retrieved 18 April, 2015 from http://liblivecd.sourceforge.net/help/info.html

JumpBox. 2013. JumpBox for the DSpace Open Source Repository. (2013). Retrieved 18 April, 2015 from http://www.jumpbox.com/app/dspace

Mary E. Piorun, Lisa A. Palmer and Jim Comes. 2007. Challenges and lessons learned: moving from image database to institutional repository. *OCLC Systems & Services: International digital library perspectives* 23, 2, 148-157.