

Running head: DSPACE PROJECT ANALYSIS

The DSpace Digital Repository: A Project Analysis

Steven Chabot

Faculty of Information Studies

University of Toronto

FIS 1311, Wednesday Section

Prof. Gord Nickerson

November 9, 2006

This paper is released under a Creative Commons Attribution-NonCommercial 2.0 license as outlined  
at <<http://creativecommons.org/licenses/by-nc/2.0/>>

## The DSpace Digital Repository: A Project Analysis

### Introduction

DSpace is an advanced digital repository system that aims to simplify the long-term archivization and access of digital research objects in any format. DSpace is an open-source, web-based system which can be remotely accessed by submitters, administrators and the general public, and can be modified to suit a particular institution's needs. Furthermore, while DSpace's flexibility allows it to be used in a variety of scenarios ("Introducing DSpace", 2006), this paper will examine the usefulness of DSpace as a research repository implemented by the library of a large university for use of its faculty and departments. Here we will examine the installation, implementation, and usage of a DSpace set-up, and address some problems or questions that may arise. A test installation of the software is beyond the scope of this analysis, but reports from other users will be cited. In the end we will conclude that any limitations of DSpace are minor, and that it would be a highly useful tool for any university to implement.

### Project Summary

DSpace was completed in November 2002 through a joint effort between Hewlett-Packard Labs (HP) and the Massachusetts Institute of Technology (MIT), who have released the resulting code under an open-source licence, specifically the permissive BSD license (Smith et al., 2003). This means that end-users can adjust, modify or improve the code as they see fit, and furthermore the project developers do evaluate and reincorporate any improvements made by users into the main distribution (Smith et al., 2003). As of this writing the software is hosted on the open-source repository Sourceforge which currently offers version 1.4 of the software, indicating the project is beyond beta testing ready for end-users ("DSpace", 2006). DSpace Federation's unofficial list has over 100 institutions using DSpace ("DSpaceInstances", 2006). We can conclude that the software is well tested and supported by

a community of users. However, as the software is open-source, neither MIT nor HP offers official support (Smith et al., 2003).

The project was designed to be a tool for institutions, in MIT's case a university, to implement a central location where faculty, departments, disciplines, labs and research centres could store their published and pre-published research for access by others and long-term archivization. The developers claim that the software was build to support "every function that a research organization needs to run a production digital repository service, but as simply as possible" (Smith et al., 2003). Furthermore, the software was designed to be multidisciplinary: it is designed around the idea of the "Community," which designs its own work flows and manages its own deposits, which we will examine under "Usage and Institutional Policy." Communities can be any size, from labs to departments to entire institutes of research (Smith et al., 2003).

As well, the repository does not simply archive text as some other e-print servers, but anything that may be part of faculty research. Text, audio and video are the most obvious data formats, but the system will except anything in any format for viewing with the appropriate software: data sets, complex computer models and simulations, even binary software (e.g. .EXE files) ("EndUserFaq", 2006). The software goes beyond the needs for eTheses and pre-print servers, although these have been implemented with DSpace (Jones, 2004; Nixon, 2003). The director of the project, MacKenzie Smith, envisions a future where scholarly journals are removed from the publishing process and universities self-publish faculty research with the help of software like DSpace ("Interview: A journey into DSpace", 2003). DSpace is a robust and flexible repository implementation that, with the right policies, will be able to handle any research users would wish to deposit in it.

### Technology Considerations

*Requirements.* DSpace is designed to run on a standard UNIX system with minimal resources (Smith et al., 2003), which should already be in place in most university environments. The system itself is composed of a standard open-source database (PostgreSQL) and web-server (Apache and Tomcat) software. The back end of the service runs on Java, and theoretically it could run on any operating system environment, but this is untested by the developers (Smith et al., 2003, although see

the DSpace Technical FAQ <<http://wiki.dspace.org/index.php//TechnicalFAQ>> ). The DSpace Foundation recommends IT support by someone with both UNIX administration experience and Java programming ability (“DSpace System Manager: Implement DSpace”, 2006), although this may only be necessary if an institution were looking to heavily modify their local installation. Given someone familiar with UNIX software installation and networking, a basic system could be installed very quickly and simply (Horsman & Pompe, 2005).

*Support.* While neither MIT nor HP offers official support, there is a very active community around the software, and it is in active development. Beyond the DSpace Wiki <<http://wiki.dspace.org>> which addresses both technical and non-technical questions, there are also general, technical and development mailing lists at <<http://dspace.org/feedback/mailling.html>> which are very active and bugs are actively tracked on the Sourceforge site <<http://sourceforge.net/projects/dspace/>> . There may be some issues with universities who are not experienced with the support process regarding open-source software and are more familiar with commercial customer support. Nevertheless, most large university libraries do have IT staff with the recommended level of experience who should be very familiar with open-source software.

#### Usage and Institutional Policy

*Submission.* After installation the system is accessed through a set of three web-based interfaces (Smith et al., 2003). One is for the end-users, one for those in the submission process, discussed below, and one for administrators (Smith et al., 2003). Those formats viewable from within the browser are loaded on demand, with all other formats available for download and viewing with the required software (Smith et al., 2003). In examining the system from the perspective of a submitter or an administrator, an installation was beyond the scope of this analysis, but we can cite other users’ impressions of the software. Nixon (2003) outlines a seven step process for depositing materials: three Description steps, Upload, Verify, Licence and Complete. These steps are tracked by a progress bar, and the submitter is free to move back and forth between the steps. For ease of use the submitter, who might not be technically inclined, does not have to know the file format of his submission as DSpace analyses the file and assigns an appropriate designation upon upload (Nixon, 2003). One issue Horsman and

Pompe (2005) found that the upload process was slow, particularly for larger files, although this may have been improved in a successive version. Lastly, the submitter can select a licence for their submission, allowing for the choice of an open-source (i.e. Creative Commons) licence if desired.

*Communities.* The submission process itself depends greatly on the policies of a particular “Community” as understood by DSpace. As noted, communities can be of any size, from a small lab to a large institute. They are defined by the internal policies regarding submission and access to the research of that group. Submitters are not bound to a particular community, but they do have to select which community their work will be submitted to (Nixon, 2003). Users of the system with different levels of involvement work within a community to access the submission and prepare it for archivization, a work not being archived until it goes through the community’s process (Smith et al., 2003).

*Policy.* While it could be the policy of a community to allow any of its faculty to submit papers which are automatically archived, a more complex example may have a group of people designated as reviewers, a member who is responsible for metadata (discussed below) and a project co-ordinator who gives final approval (Smith et al., 2003). A research object would need to be reviewed and edited according to the community’s policy before it were ultimately archived. Each person with a role in the process can log on to the system to see what objects are at what stage of review, and what action must be taken by the various members of the process. The developers of DSpace call this a “workflow,” (Smith et al., 2003) and have designed the system to be flexible enough to handle the work flow of all researchers, from sole English professors to complex bio-chemical medical research teams.

There can be problems, however, with the implementation of communities. Nixon (2003) found the communities too “flat” as sub-communities were not implemented. However, I believe this critique misunderstands the role of the community. Communities are not, primarily, for organization of the archive, which can easily be handled by metadata, but are necessary for the submission process, which can be radically different not only for different departments across the university, but also “sub-communities” within each department. Nevertheless, Nixon (2003) does state that sub-communities were added as of version 1.2 of DSpace.

## Metadata and Access

*Metadata.* DSpace archives all research objects under a qualified Dublin Core metadata standard (Smith et al., 2003). This is recorded at the time of submission, is displayed with the item when accessed, and items can be searched by their metadata by end-users (Nixon, 2003). Like all discussions of metadata, however, there are those who require both more and less information. Jones (2004) found the possible metadata as more than adequate for his uses while Horsman and Pompe (2005) found the metadata severely lacking in specificity for archive purposes. Furthermore they found the lack of multilevel description and authority control over vocabulary problematic (Horsman & Pompe, 2005). Browsing the University of Toronto's own "T-Space" repository list of subjects <<https://tspace.library.utoronto.ca/browse-subject>> without a controlled vocabulary and classification scheme proves to be daunting, and searching by subject is very difficult as well. It might be possible for individual communities to control their own vocabulary, but this is not a function of the software itself.

*Integration.* This standard metadata scheme does allow tight integration between DSpace and other digital repositories, through the implementation of the Open Archives Initiative protocol (Smith et al., 2003). This allows data submitted to DSpace to be "harvested" by other repositories. For instance, a community working in Library and Information Science, while submitting their papers to their local DSpace repository, might also concurrently submit their work to a OAI compliant pre-print repository such as the Digital Library of Information Science and Technology (DLIST) <<http://dlist.sir.arizona.edu>> without having to re-upload files or re-enter metadata a second time. This makes the connections between databases very easy and efficient, promoting scholarly interaction beyond the local department or faculty.

*Access.* Works are accessed by a unique identifier called a "handle," the goal being to have persistent citations to a particular document or object for as long as possible (Smith et al., 2003). Handles are organized by a special proxy server which keeps track of handles and their corresponding objects, allowing an item to move or change while retaining the same URL for web-browser access. As already noted, the user's web-browser will open any formats it recognizes, and any other formats will be downloaded for viewing by the appropriate software. Not only does this allow for secure archiving

and cataloguing of materials, but also gives researchers direct links to previously read materials and long lasting citations within their own publications for others to follow what they had read. These permanent URLs also facilitate long-term archivization: as file formats and technologies change, those archives which can be translated between formats can retain the same URL, allowing transparent access to users in the distant future (Smith et al., 2003).

### Summary of Issues and Benefits

*Issues.* As has been addressed, there are some problems with DSpace. In the first place, the software is open source. While this does come with its own benefits, it also comes with its own problems. Commercial support for the software does not exist at this time, neither for installation nor for later technical issues. Libraries used to working with commercial software or ILS vendors may find implementation difficult. Furthermore, some who have previously implemented the software have had problems with performance while updating files and with the structure of the communities, although these may have been fixed in successive releases of the software.

The major difficulty we have found is with DSpace's handling of metadata. While we feel that the number of fields in Dublin Core is adequate for most if not all uses (DCMI Usage Board, 2006), we are troubled by the lack of authority control when completing its fields. Without some control over uniform titles, authors and subjects accessing the items in the future will very problematic. However, this could be solved at an institutional policy level, with guidelines for submission and librarians or faculty having roles in the "workflow" overseeing metadata. While there is no scope in this paper for a discussion of necessity of controlled vocabulary, we will stress that this necessity does not just apply to paper documents, but to digital ones as well.

*Benefits.* Despite this fault, we do find that DSpace has many positive aspects. We find it to be an amazingly flexible and robust system which would be ready to handle almost any university's needs right out of the box. It has the flexibility to handle all types of documents and methods of research, as well as the simplicity to encourage non-technical users towards the Open Access (OA) of scholarly research. We also feel that, given Smith's intentions as cited above, the system would be an ready for a university to experiment in self-publishing even a part of its faculty's research. Furthermore, while open

source can have its drawbacks, it has some definite benefits. The software itself is customizable from the ground up, and any perceived problems with the system could be fixed by an institution if they so desired. If this were beyond the abilities of the institution, the software is free, has little hardware requirements, and would require little administration for a simple, uncustomized installation.

### Conclusions

It is the goal of the developer's of DSpace to make the collection, preservation, indexing and distribution of digital research objects simple (Smith et al., 2003), to the extent that it encourages researches to self-archive their own work. Despite a few drawbacks that we have noted, particularly with the lack of control over metadata, DSpace is an excellent digital repository system supported by an active community of both users and developers. Given DSpace's flexibility to archive any type of digital object and deal with any model of research within a department or other research community, it is a highly recommended system which can only improve with further development. This flexibility is increased by the fact that DSpace is open source, and any modifications or improvements can be implemented by the institutions themselves, and those improvements can be shared with the wider research community.

## References

- DCMI Usage Board. (2006). DCMI metadata terms. Retrieved November 8 2006 from the Dublin Core Metadata Initiative website: <http://dublincore.org/documents/dcmi-terms/>.
- DSpace. (2006). Retrieved November 8 2006 from Sourceforge website: <http://sourceforge.net/projects/dspace/>.
- DSpaceInstances. (2006). Retrived November 8 2006 from DSpace Wiki: <http://wiki.dspace.org/index.php/DSpaceInstances>.
- DSpace System Manager: Impliment DSpace. (2006). Retrieved November 8 2006 from DSpace Federation website: <http://dspace.org/implement/sys-man.html>.
- EndUserFaq. (2006). Retrived November 8 2006 from DSpace Wiki: <http://wiki.dspace.org/index.php//EndUserFaq>.
- Horsman, P., & Pompe, K. (2005). Building a digital archive: A dutch experience. *RLG DigiNews*, 9(6). Retrieved November 8 2006 from RLG website: [http://www.rlg.org/en/page.php?Page\\_ID=20865#article2](http://www.rlg.org/en/page.php?Page_ID=20865#article2).
- Interview: A journey into DSpace. (2003, October 20). *Open Access Now*. Retrieved November 8 2006 from: <http://www.biomedcentral.com/openaccess/archive/?page=features&issue=7>.
- Introducing DSpace. (2006). Retrieved November 8 2006 from DSpace Federation website: <http://dspace.org/introduction/index.html>.
- Jones, R. (2004). DSpace vs. ETD-db: Choosing software to manage electronic theses and dissertations. *Ariadne*(38). Retrieved November 8 2006 from: <http://www.ariadne.ac.uk/issue38/jones/>.
- Nixon, W. (2003). DAEDALUS: initial experiences with EPrints and DSpace at the University of Glasgow. *Ariadne*(37). Retrived November 8 2006 from: <http://www.ariadne.ac.uk/issue37/nixon/>.
- Smith, M., Bass, M., McClellan, G., Tansley, R., Barton, M., Branschofsky, M., et al. (2003). DSpace: an open source dynamic digital repository. *D-Lib Magazine*, 9(1). Retrieved November 8 2006 from: <http://www.dlib.org/dlib/january03/smith/01smith.html>.
- TechnicalFaq. (2006). Retrived November 8 2006 from DSpace Wiki:

<http://wiki.dspace.org/index.php//TechnicalFaq>.