2015

Archive - A Data Management Program

James H. Devilbiss
James Madison University

C. Steven Whisnant
James Madison University

Yasmeen Shorish
James Madison University

Archive – A Data Management Program

J.H. Devilbiss, C.S. Whisnant, and Y. Shorish

April 15, 2015

Abstract

To meet funding agency requirements, a portable data management solution is presented for small research groups. The database created is simple, searchable, robust, and can reside across multiple hard drives. Employing a standard metadata schema for all data, the database ensures a high level of standardization, findability, and organization. The software is written in Perl, runs on UNIX, and presents a web-based user interface. It uses a fast, portable log-in scheme, making it easy to export to other locations. As research continues to move towards more open data sharing and reproducibility, this database solution is agile enough to accommodate external participants, while satisfying the unique needs of the internal research group.

Introduction

The National Science Foundation and other federal funding agencies have begun requiring that all products of research be collected for storage and made available to others in a reasonable way (National Science Foundation, 2015). One reason for this regulation, beyond the obvious benefits of compiling information, is to prevent the loss of data that may be hidden away or forgotten by researchers. Heidorn (2008) refers to this as 'dark data' and claims that it alone could provide a wealth of previously untapped knowledge. In a study conducted by Tenopir et al. (2011) 53.6% of respondents claimed that insufficient time was the foremost reason for not making data electronically available. To satisfy both the regulations and time constraints of researchers, any data storage system must work quickly and efficiently. Large data repositories for specific areas of research already exist and generally involve unmediated sharing permissions. The data owner has released the data freely, perhaps with varying license permissions (Biosharing, n.d.; Scientific Data Sharing Project, n.d.; Data.gov, n.d.; Dryad, n.d.).

Ideally, researchers could fulfill federal funding requirements by having the ability to distribute their data while maintaining some level of control over the actual sharing mechanism.

There are many fields of research for which such existing large data repositories do not exist. This leads each researcher to develop his or her own solution. And, as noted above, the solution may be to simply ignore the issue. Without the time and interest to develop a robust solution, individuals and small groups are especially likely to ignore the problem. Evolving federal mandates mean that this head-in-the-sand approach is no longer possible.

Our solution to comply with regulations and prevent future loss of research data is *Archive*. This program provides a web interface to a data store that can maintain the collected output of a small to medium-sized research group. The data entries are stored as individual files on a networked hard drive and have associated XML descriptor files. Any file type may be uploaded and subsequently retrieved by any registered user in the system. Unregistered users may request data and can be granted temporary access at the discretion of the owner or custodian. This mediated sharing mechanism, whereby an unregistered user makes contact with the data custodian to request access, may alleviate the anxiety that some researchers feel when asked to share data.

Overview

Archive is written in Perl for portability and ease of software development. It has been developed on an open-source Linux platform (*Redhat*) but it should be possible to port it to a Windows or Mac OS system with a minimum of effort.

Each entry in the archive consists of two files: the data file and its associated XML metadata file. This permits storage of data in any format without the associated software needed to read it, as well as the ability to keep an arbitrary amount to metadata describing the data. The metadata files use selected elements from the Qualified Dublin Core schema (`dublincore.org/documents/2000/07/11/dcmes-qualifiers/`) to describe the data. This list of elements includes the Replaces/isReplacedBy, Referenced/isReferencedBy, and Requires/isRequiredBy fields to permit the association of data sets. Thus, the archive is not simply a collection of unrelated files, it is a carefully described, threaded repository of the results of the research program. In addition, the "relation" field is used to store the URL for the electronic log book entry describing the creation of the resource. If paper log books are used, this can store the volume and page numbers.

The inclusion of the Replaces/isReplacedBy fields allows the linking of files (described below) that supersede earlier versions so that no data

need be removed from the archive. The makes possible the creation of a full record of activities even if some entries are later found to be in error or superfluous. Moreover, should a subsequent correction be found to be in error, the original work is still intact and can be relinked to the newest work.

Ideally, data should be stored in plain text (ASCII) for universal access. However, any format is permissible. Other commonly used formats include PDF, various image formats (JPEG, PICT, GIF, etc.), ROOT (`root.cern.ch/drupal/`), or MAESTRO (`www.ortec-online.com/Solutions/applications-software.aspx`) files. Storage in a format readable by Microsoft Office programs (spreadsheets, documents, presentations) are also commonly used, but may be problematic in the long-term due to the proprietary nature of the platform. Nevertheless, any format is acceptable for upload to the archive. Some research products such as computer code routinely involve multiple files. Rather than upload these as many separate, linked entries, such files sets can be grouped into a *.tar* file and stored as a single entry containing all files.

The metadata for all entries in the archive are accessible to the public via the web for browsing and searching. The data itself may be downloaded by the known members of the research group by entering a user-specific password. Visitors may request a data file *via* the web using the request form. This generates an email request to the Principle Investigator (PI). Approval is granted *via* email and the visitor is given a temporary link for copying the data file. Some data sets will also require information in the log book in order to be fully understood. In such cases, the information in the appropriate log book entry can be forwarded by e-mail. In any event, the distribution of the data and the log book entry describing it is at the discretion of the PI and can be granted or refused as appropriate. This method of requiring human intervention for each outside request of the data provides for a secure system in which the producers of the data are aware of and in control of all data distribution. For a small research group where the data request rate is modest, this should not be an undue burden.

While for some groups, entering the data in the archive as it is collected is a good model, others may find that periodic updates are better suited to the usual laboratory workflow. In addition, the periodic entry method allows naturally for an embargo period during which even the metadata is not available to others. This allows for a simple way to control who can see the data (or metadata) and when. For data that is to be published, the embargo period will need to be longer, generally extending after the data is archived until the paper is accepted. Those who request such data may be informed of this increased delay during the data request process. Of course, external factors such as patent filings or federal data sharing requirements will

affect any potential embargo.

All archive activity is logged making it possible to monitor software performance and track activity. The log is saved as a text file in the directory that stores *Archive*.

An example of this Perl-based archive software system is found at `acadine.physics.jmu.edu/principium/combined`. Each research group using the software will require a separate installation of the archive. While running these on separate computers is recommended, it is not required. It is certainly possible to create several archives on one computer, provided that each instance uses separate disks for data storage and uses separately configured copies of the software.

Archive has provisions for linking data in various ways to create a rich structure rather than simple directories of unrelated files. Data are stored in directories that sort the data by year and source laboratory to allow for browsing in a reasonable way. If a physical laboratory houses multiple projects, the data entry in the archive can be sorted by project or virtual laboratory to keep data streams well-defined. Files are named with upload times (Linux timestamp) for uniqueness. In addition to browsing year and laboratory directories, a search capability is provided to select items based on metadata selected at the time of creation of the archive. By default, the searchable fields include the date, abstract, format, subject, title, and location. Other fields may be easily added to the search list when the archive is created. Searching also permits use of regular expressions (e.g. regex) to create sophisticated searches. In our testing, search times never exceeded 10 seconds for archive sizes up to 10^4 items.

The user login system includes an attempt-rate limit to deter password guessing. Users may attempt a maximum of five logins every hour. Testing shows that groups of fewer than 1000 users will have negligible login times.

The archive is capable of spanning multiple hard drives to accommodate growth. Limits on the maximum usable space on each hard drive can be defined to prevent disk writing errors. As the used space approaches the preset limit, the archive will move on to the next disk and the owner will be notified by email of the current status of each hard drive. Because all information is stored as files on hard drives, a simple backup scheme is possible. Daily backups of each drive in the archive may be made both to a local disk and/or to central servers. These are stored as *.tar* files, one per hard drive, simplifying restoration in the event of a failure. Integrity of the archive is ensured by a daily consistency check scheduled each evening. This forces the XML index files to correctly reflect the folder contents and data file linkages, ensuring that searches are accurate.

Web Interface

The *Archive* main page provides primary access to the archive. It is where a user may browse the archive, upload data sets, create links between data sets, and search and review definitions of metadata.

Figure 1: The home page with one data set, "flash", uploaded into the archive during the year 2013 that was generated at the location HHS_0111.

Browsing and Linking

Under the *Browse the Archive* heading (see figure **??**) user may explore the archive by clicking on the header for a given year. This drops down additional headers for all of the research locations with associated data sets during that year. Clicking on a location displays the data set titles as hyperlinks. These hyperlinks open a separate examination page for the associated metadata where any user may view the full metadata on for the data entry. On the browse page, to the right of each title is a radio button and a check box used for linking entries. Linking allows users to connect data together unquestionably, a known problem in research groups (Akmon, Zimmerman, Daniels, & Hedstrom, 2011). A registered user may click on a radio button to choose a metadata element to link with other data. The user may then select check boxes for other data that are associated with this data set.

Once the proper radio buttons and checkboxes have been chosen, the user must click the "Link" hyperlink in the navigation bar to reveal the options for relating metadata. The user may choose to either *require*, *reference*, or *replace* the checkboxes with the radio button. For example, if *require* is chosen then a change will be made in the respective metadata files noting that the radio button choice will require the selected checkboxes and the items indicated by the checkboxes will be required by the item whose radio button is clicked. To make the final

link, the user must supply a username and password. If the username is recognized and registered with the system, then the link between the data sets is made. If the username is not recognized, then the user receives an error message and the link is not created.

Uploading

Included in the navigation bar on the home page is a hyperlink to the upload form (see fig. **??**). Clicking on "Upload" will reveal a drop down upload form. The user must fill in the form with information about the data to generate the metadata. The entire form does not need to be filled in each time, but a few queries are required to distinguish this data from similar data and to organize it properly in the archive. The required fields are *abstract*, *location*, *subject* and *title* but can be customized in the program initialization file. A username and password must be supplied to the upload the data. Lastly, there is a checkbox at the bottom that must be checked to attempt a data upload. If any of the required fields are not filled out, or if that check box is unchecked, then the page will refresh, give the user an error message and no upload takes place. If a data set is properly uploaded a message appears on the screen confirming that the data was properly received.

Some fields are expected to be standard for a given installation and may be supplied by a template defined in an initiation file. In this way, the user has the option to edit these fields at upload time, as appropriate. The administrator may also elect to include or omit selected fields to further tailor the installation.

Searching

Included in the navigation bar on the home page is a hyperlink to the search form (see fig. **??**). The search form queries the user for the categories he or she would like to search through. These categories are limited to what has been captured in the metadata. The categories that users are allowed to search by can be manually added to or deleted from the initialization file. If *location* is a search criteria, a drop down menu is added to let users choose which of the locations to search by. The possible locations are stored in the initialization file. Additionally, there is one extra search criteria, *regex*, that allows users to search through metadata using regular expressions. The form also asks the user to provide dates to search through. If a user does not specify a date range, the search engine uses the widest possible date range. The form also includes a text box for the user to enter what text or regular expression to search for and the search button. Once the server has finished searching the metadata in the archive the results are displayed in a scrollable box below the search form.

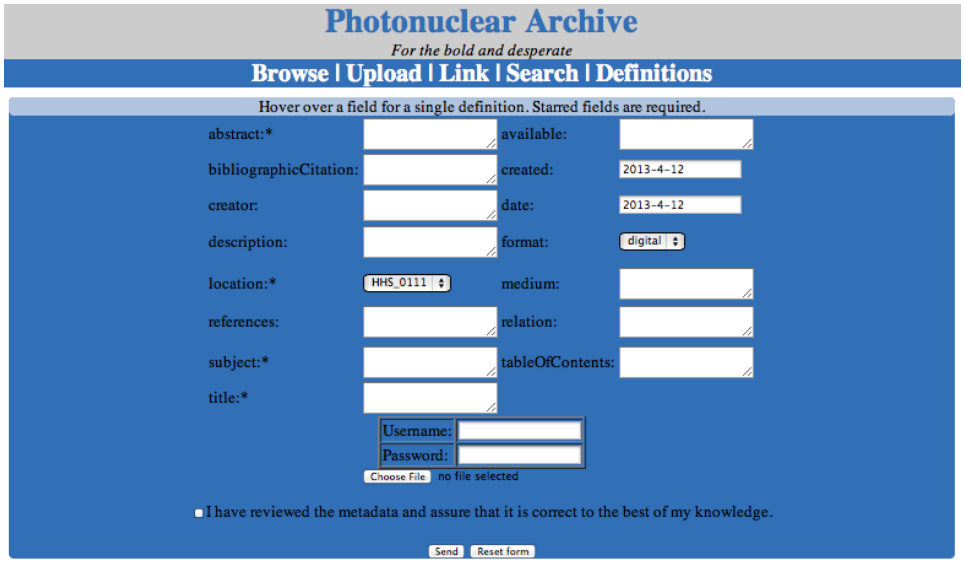

Figure 2: The upload form on the home page.

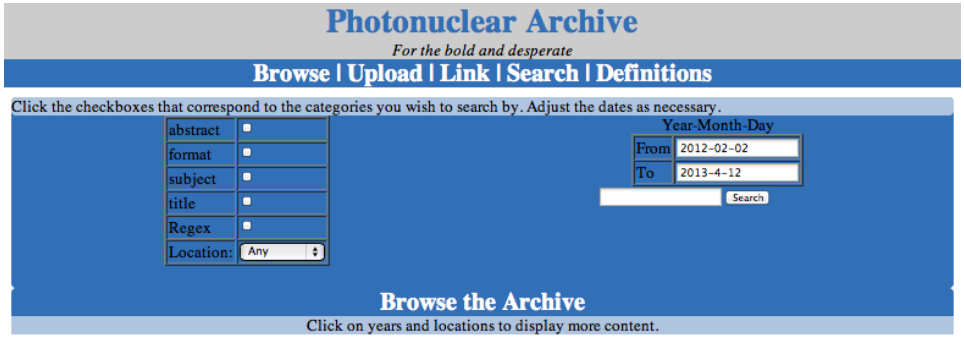

Figure 3: The search engine for the archive. *Regex* indicates that regular expression may be entered to allow for more advanced searches.

Definitions

Included in the navigation bar on the home page is a hyperlink to the definitions of each metadata element. The hyperlink brings up a column of the controlled vocabulary used to describe notitia. Clicking on each word toggles whether the definition is seen or not.

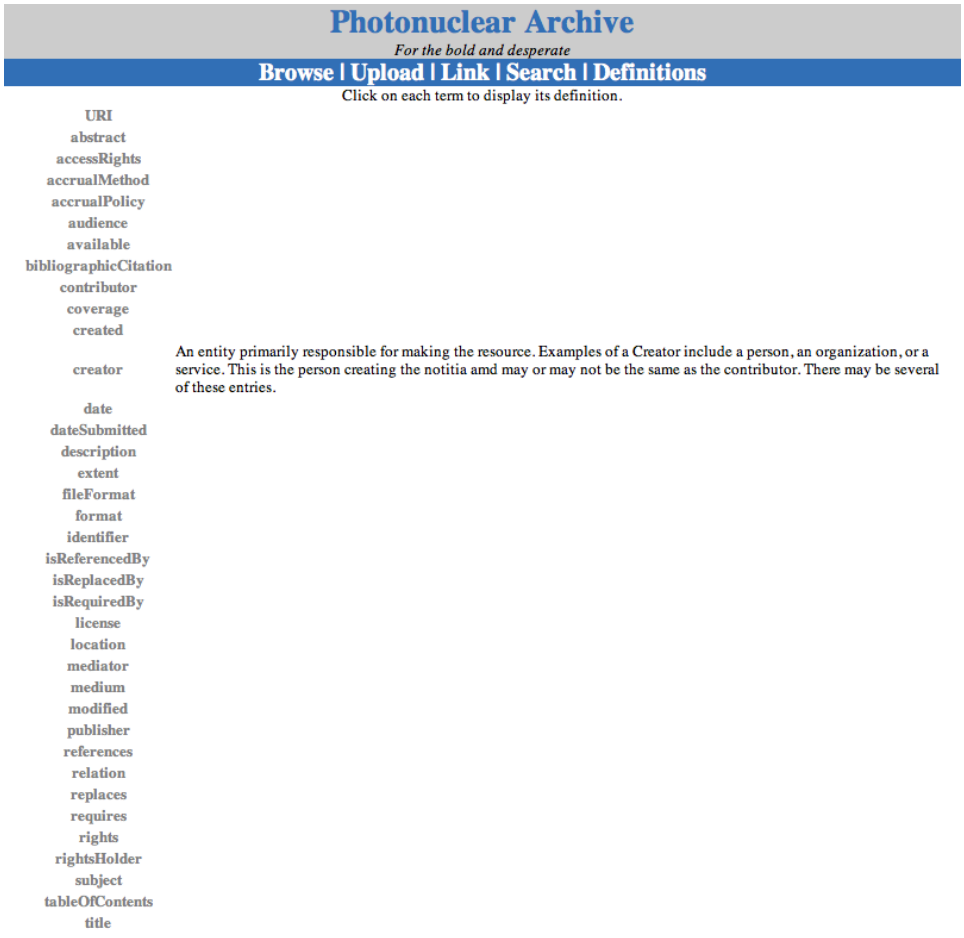

Figure 4: The definitions area of the web site with the *creator* field entry toggled to show the definition.

This controlled vocabulary came from the Dublin Core metadata schema (Dublin Core Metadata Initiative, n.d.). Dublin Core was chosen as the model because it offers a set of criteria that do not overlap, is comprehensive for most purposes, and is an ANSI/NISO accepted standard (ANSI/NISO Z39.85-2012, 2013).Controlled vocabulary al-

lows for consistent naming of criteria in research labs, yet is only utilized in about 56% of labs, according to Tenopir et al. (2011). The following are the elements that were chosen for use in the archive. The general metadata definitions given by the Dublin Core are specialized to particular needs of archiving data sets. The items included in the database are listed here with brief definitions included. If a particular implementation of the archive needs fewer, these can be ignored or set to a default value for all entries.

1. **Abstract** A summary of the resource. This is a required element.
2. **Access Rights** Information about who can access the resource or an indication of its security status. Access Rights may include information regarding access or restrictions based on privacy, security, or other policies. This is normally set once for the entire archive so that any distributed data will have this included in the metadata. These items and others can be set in the initialization file (init.xml) to create defaults for the archive. This allows many items, even ones that frequently change to appear with a default setting, simplifying the input of metadata.
3. **Accrual Method** The method by which items are added to a collection. This applies to the archive itself and is normally set as a default in the initialization file.
4. **Accrual Policy** The policy governing the addition of items to a collection. This applies to the archive itself and is normally set as a default in the initialization file.
5. **Audience** A class or entity for whom the resource is intended or useful. Usually this will be the group or collaboration collecting the data.
6. **Availability** Date that the resource became or will become available. On the date listed here, the data will be available for export to users outside the collaboration.
7. **bibliographicCitation** A bibliographic reference for the resource. Recommended practice is to include sufficient bibliographic detail to identify the resource as unambiguously as possible. In the archive, this provides the URL to the metadata file (a self-reference to the metadata file).
8. **Contributor** An entity responsible for making contributions to the resource. Examples of a contributor include a person, an organization, or a service. For the archive as a whole and for a location, this is the organization; for a specific data entry, it is the person making the entry.
9. **Coverage** The spatial or temporal topic of the resource, the spatial applicability of the resource, or the jurisdiction under which

the resource is relevant. Spatial topic and spatial applicability may be a named place or a location specified by its geographic coordinates. This is usually the city/locale where the data were collected. Temporal topic may be a named period (i.e. Jurassic), date, or date range.

10. **Created** Date of creation of the resource. This normally corresponds to the information in the logbook.
11. **Creator** An entity primarily responsible for making the resource. This is the person creating the archive entry and may or may not be the same as the contributor.
12. **Date** This is used to specify the date on which the data set was exported from the archive for use by a collaborator or an outside user. Preferably the ISO 8601 format: YYYY-MM-DD.
13. **Description** An account of the resource. Description may include, but is not limited to an abstract, a table of contents, a graphical representation, or a free-text account of the resource. This may be used to expand on the abstract in cases where a separate, more detailed account is appropriate. This is different from the abstract in that it gives a more complete account of the conditions under which the archive entry was created.
14. **Extent** The size or duration of the resource. This is the file size. It is determined by the software and is added to the XML when the archive entry is created.
15. **File Format** A digital resource format. This describes the file format and names the software required to read it. If the software is locally produced, it should be also in the archive and referenced by this entry.
16. **Format** The file format, physical medium, or dimensions of the resource. Examples of dimensions include size and duration. Most useful if archiving the description of a physical object.
17. **Identifier** An unambiguous reference to the resource within a given context. The identifier is the system epoch timestamp combined with the year and location. It is unique and added by the system when the *XML* is created. The identifier has the format:

    <year>_<location>_<timestamp>.<ext>

    This is parsed by the program to locate the file without having to search the entire directory structure.
18. **isReferencedBy** A related resource that references, cites, or otherwise points to the described resource. This points to a referencing data entry in the archive. This element may be repeated for multiple entries. Online this is displayed as a hyperlink to referenced entry.

19. **isReplacedBy** A related resource that supplants, displaces, or supersedes the described resource. This points to the data that supersedes this one. This element may be repeated for multiple entries. Online this is displayed as a hyperlink to referenced data set.

    The use of isReferencedBy and isReplacedBy make deletion of entries from the archive unnecessary and permit the archive to be treated as you would treat a log book written in ink: superseded entries are figuratively crossed out and a note is made indicating where to look to find the updated information.

20. **isRequiredBy** A related resource that requires the described resource to support its function, delivery, or coherence. This points to another data set that is required by this the current one. This element may be repeated for multiple entries. Online this is displayed as a hyperlink to referenced entries. An example would be the software that is required to read an uploaded file.

21. **License** A legal document giving official permission to do something with the resource. This usually applies to the archive only.

22. **Location** A spatial region or named place. This is the location in which the data was produced. It is used to determine the location name for storing the data entry. The list of locations are determined from the list in the XML file. This is a required element.

23. **Mediator** An entity that mediates access to the resource and for whom the resource is intended or useful. In most cases, this is the PI for the research group, or other assigned custodian.

24. **Medium** The material or physical carrier of the resource. This is usually "digital" for all computer files. In the case of the item being a physical item, not a data file, this will describe the nature of the item (gas, computer card, device, etc.). This makes it possible to archive items that are not digital and keep a record of their creation and storage location.

25. **Modified** Date on which the resource was changed. This is the date that the XML was modified.

26. **References** A related resource that is referenced, cited, or otherwise pointed to by the described resource. These can be URL or citation to relevant resource material for using or understanding the data entry. If it begines with "http:" then it is assumed to be a URL and is displayed as such on the web page. If not, the text is simply displayed. Multiple items are allowed, each separated by a semi-colon (;).

27. **Relation** A related resource. This is the URL for the log book entry that describes the creation of the information in the data

entry. The URL for the metadata (bibliographicCitation) is the one given to external users and the one used in the log book for cross-reference. Multiple items are allowed, each separated by a semi-colon (;).

28. **Replaces** A related resource that is supplanted, displaced, or superseded by the described resource. This refers to the entry that is replaced by the current one. This element may be repeated for multiple entries. Online this is displayed as a hyperlink to referenced entry.
29. **Requires** A related resource that is required by the described resource to support its function, delivery, or coherence. This refers to the entry that require this item. This element may be repeated for multiple entries. Online this is displayed as a hyperlink to referenced entry.
30. **Rights** Information about rights held in and over the resource. Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights. This applies to the archive only and should conform to the institution's standard.
31. **Rights Holder** A person or organization owning or managing rights over the resource. This is typically the project PI and/or the institution. This applies to the archive only.
32. **Subject** The topic of the resource. Typically, the subject will be represented using keywords, key phrases, or classification codes. Recommended best practice is to use a controlled vocabulary. A list of locally useful keywords are expected to be different for each archive. This is a required element.
33. **Table Of Contents** A list of subunits of the resource. If the entry is a single file, this is not required. However, it is possible in the case of software packages added to the archive, that the entry will be a *.tar* file containing many component files. In this case, this element will list the component filenames.

    This is the preferred method for storing files that only are useful or make sense in the context of the other files like a program or a multi-document LaTeX file (which may include style sheets, etc.). For acquired data or program outputs, they should be uploaded individually and cross-linked to other, related items.
34. **Title** A name given to the resource. This is a required element.

Storage

The archive is maintained in such a way that it may span an arbitrary number of disk drives. This allows the archive to grow without regard

to the current space available by simply adding new hard drives to the system. Given that currently available hard drives are readily found in the 1-4Tb size range, unless the research group is producing very large quantities of large data sets, the number of hard drives should be expected to grow slowly.

The drives allocated to the storage system are known to the archive via an XML file at the top of the directory on the first hard drive used. With this list of hard drives to use and the amount of space available on each, the program will automatically fill one disk after the other and maintain the archive transparently.

The Primary Hard Drive

The primary hard drive is the most important. At the top level it contains all of the necessary programs and settings files. The index file at the top of this first hard drive contains all of the information that the computer needs to find the directories for each year.

Other Hard Drives

The first criteria checked when storing data is the amount of free space on the hard drive currently being used. If there is not enough space to hold the uploaded data set, the program moves on to the next drive on the list. The list of drives and their paths is kept in the initialization file.

The Examination Page

Upon clicking the title of a data set under the browsing area on the home page, the user is redirected to the examination page. The examination page displays the metadata information of a data set for the user. If any parts of the metadata are titles of other metadata, then those titles are made hyperlinks to their respective metadata files. Because a login is not required to browse the archive, not all of the metadata is diplayed on the examination page. Sensitive data, such as the URL, is withheld.

On the right side of the page is a form allows users not in the system to request data from the owners. The request form asks for the user's name, institution, email, phone number and reason for requesting the data. Upon submission of the form, the email and phone number are checked for formatting. If either are not in a reasonable format, the user receives an error message on the next webpage. If the email and phone number pass inspection then a soft link is made on the serving computer for the data requested. This link is formatted into a URL and emailed with the form to the archive owner (or custodian) for

his/her examination. If the archive owner decides that the requesting user may have temporary access to the data set, then the owner may forward the email to the user. A scheduled background task (cued by the UNIX Cron utility) deletes the link in 72 hours. This allows the user to access the data for a limited time.

If the user requesting data is registered, then she or he may fill out the form below the data request form to immediately download the data set. Once the username and password are typed correctly, the next page provides a hyperlink giving the user access to the data set.

Programs

Pass_Gen

The Pass_Gen program is run by the owner when managing users, usernames and passwords. The program can add and remove users from the system by modifying the initialization file. The program allows users to change passwords. Users are not allowed to change passwords directly from the website to enhance security. All passwords are encrypted using a salted hash and stored locally.

Crawl

To prevent errors from building up in the metadata from failed linking, a program was written to check the reciprocality of links. The *Crawl* program goes through every metadata file in the archive and examines the linking categories, *references*, *requires*, and *replaces* for each metadata element. Each of those elements can potentially reference another element. Each referenced metadata element is then checked for the original metadata in the conjugate category, *isReferencedBy*, *isRequiredBy*, and *isReplacedBy*.

Special Files

Initialization File

An initialization file is included to control broad aspects of the archive. The first elements called "disk" in *init.xml* are the hard drives the archive uses. For each disk, a directory and minimum hard drive space in Mb must be given. The minimum hard drive space acts as a threshold for deciding when the archive is required to move on to the next hard drive. If an uploaded file is larger than the minimum hard drive space but the hard drive has yet to reach its fill limit, the archive will move on to the next hard drive. As drives fill up, the administrator

will receive an email with a warning detailing which hard drives have filled up and how many free hard drives are left.

The next tag is *download* which requires the administrator's email. This is the address that will be emailed when the archive must make the administrator aware of some condition, e.g. an unregistered user is requesting data, or a hard drive has no free space. The next tag is *main*. It holds the base web address to which the address for all other pages will be appended. Within *main* is the tag *fields* which contains the controlled vocabulary and their definitions.

Log File

A text file is kept for the archive to events. If data is uploaded or metadata modified, the log file records what happens with a local date and time. Should an error occur with the system, the log can be used to assist in tracing the source of the error.

Discussion

This web based, generic archive system has been written for small to medium research groups for relational storage of data. Data are stored with associated metadata files describing the contents of the data, and any relations the data have with other data. The system allows for the distribution of data, as mandated by the National Science Foundation and other funding agencies, at the discretion of the PI. It is conceivable that research groups could, with minimal customization, create an open set of permissions that would allow for unmediated data discovery and download.

Institutions that lack formalized data repositories or institutional data management infrastructure may find this solution appealing. Testing has shown that the system is portable, requiring only a UNIX operating system and sufficient disk space to run. Multiple archives can be housed on a single computer, although this may lead to more complicated recoveries in the event of a hard drive failure. An instance of the system could be launched for a department or an entire college, depending on the size and scope of the local conditions. Above all, the system is flexible. Minimal programming in a stable, freely available language, Perl, is required for customization.

The future of data storage and retrieval options is impossible to predict in any detail. Some researchers may find that in a few years their institution adopts an enterprise solution, or best practices in a particular field will evolve solutions that can be implemented across institutions. In the meantime, for small groups where resources are limited and output is modest, the software described here offers an inexpensive

solution. Additionally, should discipline-based or institution-based solutions arise, the data is stored in a transparent fashion that can allow straightforward (if a bit tedious) transfer into the larger system.

Experience with the program shows that minimal training of users is required to achieve reliable, uniform metadata when adding data. This makes it a good solution for groups including undergraduates where the turnover rate is higher than for graduate students and postdocs.

Since the computing demands are modest, the software can be implemented on older computers that have been superseded in the laboratory. The price of adding USB hard drives for data storage is similarly low. Storing the data on external drives makes it is easy to move the entire system to another platform should the computer fail. The ability to easily store backups as *.tar* files means that extra security can be obtained by periodic backups to networked drives that are themselves backed up regularly by university computer support.

The basic access to the data by visitors to the web site is all funneled through the PI or custodian. This gives the PI or custodian full control over when and how the data are available to others. This mediated form of data sharing not only allows each group to choose it's own policy, but allows the policy to be different for data sets of different sensitivity.

References

Akmon, D., Zimmerman, A., Daniels, M., Hedstrom, M. (2011). The application of archival concepts to a data-intensive environment: working with scientists to understand data management and preservation needs. *Archival Science, 11*(3-4), 329-348. doi: 10.1007/s10502-011-9151-4

ANSI/NISO Z39.85-2012 The Dublin Core Metadata Element Set. (2013) Retrieved from `www.niso.org/standards/z39-85-2012`

Biosharing. Retrieved April 14, 2015 from `biosharing.org/`.

Data.gov. Retrieved April 14, 2015 from `www.data.gov/`.

Dryad Digital Repository. Retrieved April 14, 2015 from `datadryad.org/`.

Dublin Core Metadata Initiative. (n.d.) Retrieved April 14, 2015 from `www.dublincore.org`

Heidorn, Bryan P. (2008). Shedding light on the dark data in the long tail of science. *Library Trends, 57*(2), 280-299. doi: 10.1353/lib.0.0036

National Science Foundation. (2015) Today's Data, Tomorrow's Discoveries. Retrieved from `http://www.nsf.gov/pubs/2015/nsf15052/nsf15052.pdf`

Scientific Data Sharing Project. Retrieved April 14, 2015 from `scientificdatasharing.com/`.

Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A., Wu, L., Read, E., & ... Frame, M. (2011). Data sharing by scientists: Practices and perceptions. *PLoS ONE, 6*(6). doi:10.1371/journal.pone.0021101