The Problem of Preserving Database-Driven Content

Bill Anderson, Georgia Tech, Atlanta, GA and Denis Galvin, Rice University, Houston, TX

Abstract

DSpace and other digital repositories present unique challenges for preservation. They are typically driven by programming languages and don't fall into the usual folder/subfolder/file data structure. Materials can be difficult to extract, so individually-tailored approaches must be used. Both Georgia Tech and Rice belong to the Metaarchive cooperative, a group of schools and museums which have formed a private LOCKSS network (PLN) to do digital preservation. Both schools use DSpace for their digital repositories. For various reasons, the LOCKSS software does not easily harvest materials out of DSpace. Both schools were able to overcome these issues and deposit materials into the Metaarchive network. By collaborating and working through the cooperative, each school came up with a different approach.

LOCKSS and MetaArchive

LOCKSS stands for Lots of Copies Keep Stuff Safe. It is simple to explain and understand. Prior to the electronic age individual libraries would purchase copies of print journals. Those print journals would be distributed to multiple libraries or organizations. For example both Tulane and Rice might purchase a copy of the journal Information and Organization. If something happened to the copy at Tulane, Rice would still retain a copy of the item and perhaps hundreds of other institutions would also. Conversely, if Rice was purchasing copies of the item and the publisher folded, Rice would still own the collected journals.

In the current electronic climate institutions don't control journals they purchase. If a publisher suffers a catastrophic loss and has failed to safeguard the electronic information no copies of the item remain. If they go out of business the journals they have published go away. Institutions no longer control the items they have already purchased and there is no longer redundancy in the distribution process. LOCKSS is an attempt to solve this problem, but the idea has been extended to formats other then electronic journals by private LOCKSS networks (PLNs) like the Metaarchive network. LOCKSS essentially does three things; it harvests content by acting as a web crawler, it audits content to ensure data integrity, and it acts as a proxy component in case data becomes inaccessible. In the public version of LOCKSS, libraries around the world collect content directly from the publisher's website and then compare the collected content to what is available from the publisher in order to establish the content's authoritative version. [1] Once the content has been collected the LOCKSS software polls the data to ensure integrity. If for some reason anything is lost libraries can fall over to the content they have stored on their LOCKSS box by proxy. It is a "light" archive, meaning authorized users have access to the content.

PLNs are membership-based geographically distributed networks that are dedicated to the long term survival of digital archive. [2] Metaarchive is an example of a PLN as is the Closed Lots of Copies Keep Stuff Safe (CLOCKSS) project. PLNs enable like-minded institutions to shoulder the responsibility of preserving in perpetuity scholarly e-content of importance to the group. (Reich, V., & Rosenthal, D. 2009) They typically operate as "dark" archives meaning content is typically not accessible to outside parties

Metaarchive is a PLN, and it is operated outside of the LOCKSS network. It was started by a series of academic institutions in cooperation with the Library of Congress. In 2007 it transferred to an independent, unincorporated, international membership association, the MetaArchive Cooperative, with the purpose of supporting, promoting, and extending a collaborative approach to distributed digital preservation practices. [3] Metaarchive preserves materials that are thought to have value to each individual institution, but might not necessarily be of value to the group of institutions as a whole. [3]

DSpace doesn't play well with LOCKSS

Because the LOCKSS system was designed to crawl static websites, there are inherent problems in using it with dynamic, database-driven applications. In DSpace, content is retrieved from the database on the fly, in response to user search and browse requests. There is no simple, static collection page with a link to every item in the collection. Item handles and metadata are independent of the collections in which they are held, and the exact structure of the HTML pages depends largely on the details of the installation. The actual bitstream data for an item is held in a complex file structure, accessible only through the database. In theory, a DSpace installation could be almost exclusively search-driven, providing no visible links to material at all.

This is obviously a far greater level of complexity than the LOCKSS daemon was designed to handle. Two Metaarchive member institutions, Rice University and the Georgia Institute of Technology, had DSpace institutional repositories as primary components of the material they intended for preservation. Predictably, initial attempts to solve this problem within the LOCKSS framework resulted in clunky and unreliable plugins, redundant and/or incomplete harvests, and general confusion. Both Rice and Georgia Tech ended up with effective solutions, but their approaches were entirely different.

A LOCKSS Plugin

LOCKSS plugins are XML files which define how LOCKSS daemons fetch and preserve content. They are written using a utility called the plugin tool which provides a graphical interface for plugin creation and testing. It allows developers to define crawl rules, crawl schedules, filter rules and other relevant information needed for LOCKSS to harvest a collection.

The Rice Solution

Rice's solution to preserving data in DSpace was designed to produce a quick harvest. Rice joined Meataarchive in 2008, but had a lengthy process moving contracts through the legal department. It took a while to get a server, and then it took a while longer to get everything set up with the Metaarchive network.

After spending a long time getting everything going there was some sense of urgency to get the project moving. This influenced the decisions about how collections would be harvested.
The first sets of digital objects that Rice preserved were its electronic thesis and dissertations (ETDs). There are almost 7,000 ETDs which take up approximately 43GB of space. The maximum size for an archival unit (AU) at the time this harvest was done was 10GB so these collections encompass five AUs. As has been stated, the LOCKSS daemon struggles with the structure of DSpace's. There is no way to make it start under a specific collection and have it grab everything. Because of this a decision was made to create links to every item in the collection which we wanted to harvest. Manifest pages are very simple web pages which grant permission for a collection to be crawled. They are typically put up on the same server that will have its content preserved by the LOCKSS software. Manifest pages also act as a starting point for a collection. Because they act as starting points they can also contain a collection of links which can be crawled. We created one master manifest page which contained links to five "sub" manifest pages which contained links to every ETD in DSpace. The links were created by running a SQL statement in the DBMS which houses DSpace. They contain pointers to the full record which contain HTML metadata about the digital object which contain a link to the object which is to be preserved. The manifest page also contains a link to all the Dublin Core metadata for the collection which will be pulled in by the harvester.
The plugin was written specifically for the five sub-manifest pages. Plugins can contain configuration parameters along with crawl rules. In many plugins the only configuration parameter needed is the base URL which is typically the server name where the harvest will take place. For Rice's ETDs we added a second parameter called "part" which takes an integer as its argument. This parameter sets up a variable "part" for the crawl rules to work with.

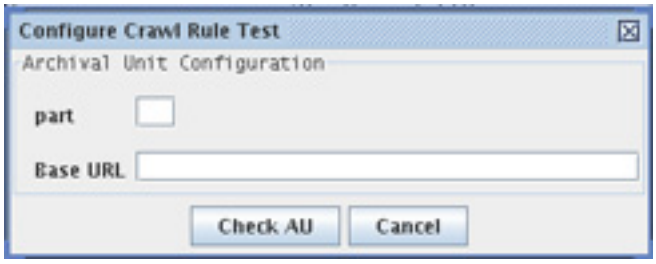

Figure 1. A configuration rule

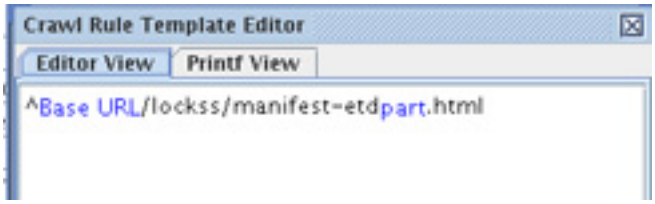

Figure 2. A crawl rule

Which the crawler then pulls from the manifest page:

manifest-etd1
manifest-etd2

When the crawl happens it moves through the manifest page to the sub-manifest pages and onto the DSpace server pulling in both the metadata and the digital object.
One of the problems with this approach is that it uses a static list of links. If anything new is added to the collection links must be added to manifest pages that also points to the content. For Rice's ETDs this would mean creating a new sub-manifest page with a link to it off the main manifest page. The new AU would then need to be ingesting by members of the Metaarchive network. This is not a lot of work, but it is not ideal.
Rice's solution to some extent represents a "good enough" approach. It produces a quick harvest of content, and as long as a due diligence is paid to collections which have been harvested it is sustainable. It is likely however that Rice will change its approach to harvesting DSpace as solutions and software evolves.

The Georgia Tech solution

Georgia Tech was an early member of the Metaarchive cooperative, and preserving our institutional repository (SMARTech) was an important goal from the beginning. We tried a number of solutions, but none gave us the reliability and flexibility we needed. We wanted the contents of a given harvest to be produced on the fly, in response to the LOCKSS daemon's request; we wanted to be certain that a collections full contents were accurately represented in LOCKKS; and we wanted something we could continue to rely on as SMARTech expanded. In attempting to solve the problem, we discussed using DSpace's native import/export functionality, and we also considered modifications to the LOCKSS plugin structure. The first wasn't well suited either to remote access or to LOCKSS' auditing functionality; and the second was deemed to be impractical and unsustainable. It became apparent that any good solution would need to rely on DSpace's OAI webapp, which was already designed to facilitate the harvesting of data.
OAI-PMH (Open Archives Institute Protocol for Metadata Harvesting), is a standard protocol for collecting and exchanging metadata among archives. DSpace comes with native OAI support, and SMARTech has implemented the DSpace OAI service and been registered as a data provider with Open Archives from the beginning; so we already knew the process was reliable and well-supported. To retrieve metadata from an OAI data provider, the requester attaches a series of parameters to the URL of the provider's OAI service, including a 'verb', which tells the service what supported actions to perform. This request would return all the metadata from the SMARTech collection with handle xxxx/xxxx, as METS-encoded xml:

http://smartech.gatech.edu/oai/request?verb=ListRecords&metadataPrefix=mets&set=hdl_xxxx_xxxx

This proved the key to our problem; but we ran into several challenges. An OAI request only returns metadata, it does not fetch the actual bitstream; and LOCKSS isn't fluent in XML, anyway. The handle attached to each item is included in the

header; but the LOCKSS daemon wouldn't be able to parse this information from the XML unaided. In addition, DSpace has a hard limit on the number of full records that can be retrieved at once, which we didn't want to override; and convincing the LOCKSS daemon to implement DSpace's continuation tokens would have taken a major rewrite of the code. Finally, many of our SMARTech collections were too big to fit in one archival unit, or too small to constitute a unit by themselves. It was obvious we weren't going to be able to just feed the LOCKSS daemon an OAI url and let it go to work. More was needed.

What we finally came up with was a plugin with only a few parameters. It looks for a manifest page which indicates whether the AU in question contains single, multiple, or partial SMARTech collections, and which contains the collections number(s). This pages calls a PHP script which constructs OAI urls for the indicated collections, using the "ListIdentifiers" verb, which returns only the item headers and is not limited by DSpace. The resulting XML is run through an XSLT stylesheet, which translates the header information for each item into a link to the full item record page in SMARTech; a page which contains all the items metadata, along with links to its associated bitstreams, and remarkably few links to DSpace functions and ancillary pages. These full-item links are returned to the LOCKSS daemon for its crawl.

We still have some issues to address. Our primary concern at the moment is formulating a recovery plan which will allow us to reassemble SMARTech from the harvested data; but we're confident that all the necessary information is there.

Conclusion

Both Rice and Georgia Tech were able to harvest dynamic content into the LOCKSS system by improvising, and inducing the technology to move beyond what it was designed to do. Many of the techniques employed -- SQL queries, adaptation of existing harvesting techniques like OAI, light-weight scripted frameworks -- could be adapted to facilitate other dynamic repository systems, such as Fedora or EPrints; but the ultimate solution should be a preservation harvesting system flexible enough to adapt to any web-based content.

References

[1] Reich, V., & Rosenthal, D. (2009). Distributed Digital Preservation: Private LOCKSS Networks as Business, Social, and Technical Frameworks. Library Trends, 57(3), 461-475.

[2] Skinner, K., & Schultz, M., (2010) Guide to Distributed Digital Preservation Atlanta, GA: Educopia Institute

[3] Katherine Skinner, & Martin Halbert. (2009). The MetaArchive Cooperative: A Collaborative Approach to Distributed Digital Preservation. Library Trends, 57(3), 371-392.

Author Biography

Denis Galvin received his BA in English from West Virginia University (1998) and is currently twelve credits shy of his MLIS from the University of North Texas. He worked previously for the library system at Carnegie Mellon University in Pittsburgh and has been in IT in libraries for over 12 years. His library career has mainly focused on the integrated library system, but he has worked extensively with LOCKSS while at Rice.

Bill Anderson is a Digital Library Developer at the Georgia Institute of Technology. He had been an information technology professional for twenty years, and has worked in libraries and academia for most of his career, including stints at Emory University and Duke University. In his current position, he has primary technical responsibility for the library's institutional repository, SMARTech, among other digital projects. He has worked on the Metaarchive project for two years.