Identifying and Implementing Modular Repository Services: Transfer and Inventory

Leslie Johnston
Library of Congress Office of Strategic Initiatives
101 Independence Ave., SE
Washington D.C., 20540-1340
707-202-2801
lesliej@loc.gov

ABSTRACT

In recent work at the Library of Congress, we have been identifying requirements for digital repositories for locally created collections and collections received from partner institutions. Our most basic needs are not surprising: How do we know what we have, where it is, and who it belongs to? How do we get files – new and legacy – from where they are to where they need to be? And how do we record and track events in the life cycle of our files? This paper describes current work at the Library in implementing tools to meet these needs as a set of modular services -- Transfer, Transport, and Inventory -- that will fit into a larger scheme of repository services to be developed.

1. INTRODUCTION

In examining the reasons why more institutions do not have trusted digital repositories for their collections, one always ends up with a long list of daunting requirements for building such a beast. Where do you begin? What are your most basic needs? What is the first step? In recent work at the Library of Congress, we have determined that our most basic needs are really basic: How do we know what we have, where it is, and who it belongs to? And how do we get files – new and legacy – from where they are to where they need to be?

The Library of Congress Office of Strategic Initiatives has been working on solutions for a category of activities that we refer to as "Transfer." At a high level, we define transfer as including the following human- and machine-performed tasks:

- Adding digital content to the collections, whether from an external partner or created at LC;
- Moving digital content between storage systems (external and internal);
- Review of digital files for fixity, quality and/or authoritativeness; and
- Inventorying and recording transfer life cycle events for digital files.

Transfer processes are not surprisingly linked with preservation, as the tasks performed during the transfer of files must follow a documented workflow and be recorded in order to mitigate preservation risks. Defining, implementing, and documenting appropriate transfer processes depends on the requirements of each collection building project, which can vary wildly. Best practices are still emerging.

2. WHERE IS THE CONTENT COMING FROM?

The Library of Congress has been digitizing its collections for over 15 years -- making collection materials available online since 1994 starting with the "American Memory" site[1] -- concentrating on its most rare collections and those unavailable anywhere else. The collections include photographs, manuscripts, maps, sound recordings, motion pictures, and books, as well as "born digital" materials.

The Library of Congress is involved in a number of collection-building activities with external partners. The National Digital Information Infrastructure and Preservation Program (NDIIPP)[2] (Anderson, 2008) provide funding through the National Science Foundation to over 130 institutional partners that send content to LC for stewardship and preservation. Content under stewardship by NDIIPP partners includes geospatial information, web sites, audio visual productions, images and text, and materials related to critical public policy issues. Each partner may deliver its content by a different transport mechanism, e.g., shipping hard drives or network transfer. Once acquired, the digital content must be validated and verified, inventoried, and placed in archival storage.

The National Digital Newspaper Program (NDNP)[3] (Littman, 2007) is a partnership between the NEH and the Library of Congress to provide enhanced access to United States newspapers. Over a period of approximately 20 years, NDNP will create a national, digital resource of newspapers from all the states and U.S. territories published between 1836 and 1922. In the NDNP transfer process, packages of digitized newspaper prepared by awardees are delivered on hard drives.

[1] The American Memory collections are available at: http://memory.loc.gov/ammem/index.html.
[2] For information on NDIIPP, please see: http://www.digitalpreservation.gov/.
[3] For information on NDNP, see: http://www.loc.gov/ndnp/.

Johnston, Leslie. "Identifying and Implementing Modular Repository Services: Transfer and Inventory." *Proceedings of DigCCurr2009: Digital Curation: Practice, Promise and Prospects;* April 1-3, 2009; Chapel Hill, North Carolina. Chapel Hill: School of Library and Information Science, University of North Carolina at Chapel Hill, 2009, pp 145-148.

In 2004, the Library's Office of Strategic Initiatives created a Web Capture team[4] to support the goal of managing and sustaining at-risk digital content. The team is charged with building a Library-wide understanding and technical infrastructure for capturing Web content. The team is identifying policy issues, establishing best practices and building tools to crawl, collect, and preserve Web content. As of fall 2008 the team has completed 17 Web archive collections and is working on building Web archives for four collections comprising approximately 3000 Web sites.

3. DEPOSIT AND TRANSFER TOOLS

Content transfers to the Library have largely consisted of small numbers of bulk transfers of content – typically, tens to hundreds of gigabytes at a time – using varied, primarily manual processes driven by Library staff to pull the content into the Library's environment. Through these experiences we have gained experience in transfer processes, and have been able to simplify the variety and complexity of the initial approaches to such bulk transfers. As the content packaging, transfer, and inventorying approaches have standardized, it has become possible to plan to grow and automate the number of transfers, in order to support new transfer scenarios in which transfers might involve many pushed deposits at smaller scale than the current bulk transfers.

For the purpose of transferring content to the Library, a package is a set of files stored in a file system, which may be a subset of a larger collection of content, to be transferred and managed as a unit. The set of files comprising a package may be transferred as a single file in a container format such as ZIP or tar to be unpacked upon receipt. Working with John Kunze of the California Digital Library, Andy Boyko, Justin Littman, Liz Madden, and Brian Vargas of the Library produced a generalized version of what had been initially referred to as the "LC Package Specification," now called "**BagIt**."[5]

The base directory of a Bag contains a **file manifest**, a **content directory**, and an optional **package information** directory. The **content directory** contains the contents of the package, as defined by its producer. The content directory may have any name and internal structure. There is no limit on the number of files or directories this directory may contain, but its size should make practical transfers easier, based on physical media limitations or expected network transfer rates. In the Library's experience, 500 Gb is the recommended maximum size, although Bags as large as 1.8 Tb have been transferred.

The file manifest lists the names and checksums of the content files and the package information files, excluding itself and any shipping files. The Library has been working primarily with md5deep[6], but any commonly recognized cryptographic checksum algorithm can be used to generate the manifest. Neither the file manifest nor the package manifest obviates the need for descriptive metadata being supplied by the package producer. The manifests assist in the transfer and archiving of the package as a unit, rather than supplying any description of the content.

Bags may be created before or after the act of transfer. The creation and communication of original checksums that accompany the Bag for verification after completion of the file transfer support a more easily audited process. In-house, content that is not received in Bags will have Bags generated as an aid to verifiable internal transport to archival systems and to aid in file-level preservation.

The Library has created a **Bag Validator script**, which checks that all files listed in manifest are in the data directory; there are no files in the data directory that are not listed in manifest; and there are no duplicate entries in the manifest. The **VerifyIt script** is used to verify the checksums of files in a Bag against its manifest. The Bag Validator is a Python script and VerifyIt is a shell script.

A client-side **Bagger** application has been developed to assist partners engaged in small-scale deposit transfers, automating the packaging and submission of locally-hosted content without requiring Library involvement, and ideally requiring no client-side IT support or infrastructure. This tool will be equally suited for packaging and transferring internal LC content, such as DVD or CD archives, to centralized transfer and storage environments. It is implemented as a Java Web-Start application for use across platforms, and supports the aggregation of files into Bag packages, including the creation of checksum manifests and Bag information files. This application was in part built on top of **BIL**—the BagIt Library—a Java library developed to support Bag services.

In order to support the expanding numbers and types of transfers, several software tools were needed to help automate transfers. The ***Deposit service*** is a web-hosted application for use by transfer partners in registering a new transfer; this application will support the registration and initiation of the transfer content via network transfer (rsync, ftp), and via fixed media, such as hard drives or DVDs. The web application is implemented using the Django web framework[7]. At the time of this writing, Deposit services are mid-way through the production implementation process, including review by representatives of the multiple digital content acquisition projects.

The Deposit tools are tied to a series of ***Transfer*** and ***Transport*** back-end tools used for retrieval, receiving and managing of content transfers. The **Parallel Retriever** implements a simple Python-based wrapper around wget and rsync, capturing files and producing a package that meets the BagIt specification when given a "file manifest" and a "fetch.txt" file. It has been used to transfer content from several transfer partners hosting rsync and HTTP servers, at rates exceeding 200Mbps over Internet2. It was initially built specifically for transfers from the Internet Archive to the Library via rsync, but has been extended to HTTP and FTP.

Underlying "Core Transfer" components support various transfer functions. The components are completely independent of any workflow, though they may of course be invoked by any designated workflow. "Core Transfer" services provide a

[4] For information on the Library's web capture activities, see: http://www.loc.gov/webcapture/.

[5] The current BagIt specification is available at: http://digitalpreservation.gov/library/resources/tools/docs/bagitspec.pdf.

[6] http://md5deep.sourceforge.net/.

[7] http://www.djangoproject.com/.

container for running transfer components so that they can be invoked though a Java Remote Service and respond with the Service Request Broker. It is implemented using Spring[8] and Hibernate[9].

4. INVENTORY TOOLS

The goal in developing Inventory Tools is to satisfy needs identified through the process of doing transfers manually and attempting to record their outcomes. These include keeping track of package transfers for a project, tracking individual packages and events associated with them, and a list of the files that make up each package and their locations. For legacy collections these tools can be pointed at existing directories to package, checksum, and record inventory events to bring the files under initial control.

The ***Inventory System*** has three parts: a package data model, a suite of command line inventory tools, and a reporting web application. The package data model instantiates a domain model for packages (Projects, Packages, Canonical Files, File Locations, and File Instances) which can be recorded in a persistent way, updated, and queried. The package data model is implemented using Java objects mapped to a PostgreSQL database using Hibernate for object-relational mapping. The Inventory tools inspect packages and update the database, and the reporting web application allows users to view reports on packages.

Since the Inventory System must also represent the history of a package, it must record events. There are events that occur on a Package level (Package Events) and on a file level (File Location Events). Examples of Package Events include "Package Received Events," which are recorded when a project receives a package; and "Package Accepted Events," which are recorded when a project accepts curatorial responsibility for a package. Examples of File Location Events include "File Copy Events," which are recorded when a package is copied from one File Location to another; and "Quality Review Events," which are recorded when quality review is performed.

The Transfer, Transport, and Inventory tools can be tied together into any of a number of project-based **Workflow systems**. The underlying workflow engine is jBPM[10], an open-source workflow system. The drivers of a workflow are process definitions, which represent the process steps. jBPM Process Definition Language (jPDL), the native process definition language of jBPM, is used to encode the workflow process steps as XML. The workflow can be designed using the visual editor Graphical Process Designer, a plug-in for the Eclipse platform.[11] A web user interface, called the **Transfer UI** in its first implementation for the NDNP, allows users to identify lists of tasks to be performed, initiate, monitor and administer processes; and notify the workflow engine of the outcome of manual tasks, including task completion. Workflow tasks instantiated through the system include transfer, validation by an NDNP-specific validation application (Littman 2006), manual quality review inspection, and file copying to archival storage and production storage. The Transfer UI was implemented using Spring MVC. Both the Inventory system and all workflows are built on top of the BIL Java Library.

5. CONCLUSIONS

At the time of this writing, the Transfer, Transport, Inventory and Workflow services have been put into production for NDNP (Littman, 2009). The Transfer and Transport tools have been put into production for NDIIPP, and production implementation is under way for the Inventory, Deposit, and Workflow services, as well as the Bagger application. By the end of 2008, many incoming collections will be processed using these tools. The expectation is that a retrospective inventory of the Library's digital collections will be undertaken in 2009 using these tools.

The first of the Transfer tools—the Parallel Retriever, the Bag Validator, and VerifyIt—have been released by the Library of Congress as open source on SourceForge[12]. The BIL Java Library is scheduled for release in April 2009. Additional tools and utilities will be released over time.

Why are such transfer tools and processes so important? While our initial interest in this problem space came from the need to better manage transfers from external partners to the Library, the transfer and transport of files within the organization for the purpose of archiving, transformation, and delivery is an increasingly large part of daily operations. The digitization of an item can create one or hundreds of files, each of which might have many derivative versions, and which might reside in multiple locations simultaneously to serve different purposes. Developing tools to manage such transfer tasks reduce the number of tasks performed and tracked by humans, and automatically provides for the validation and verification of files with each transfer event.

Why are we looking at close integration between transfer and inventory functions? Inventorying and audit functions have been identified as a vital aspect of data curation. One example initiative is the JISC Data Audit Framework project (Jones et al, 2008)[13], where work is proceeding on the development of a registry component intended for recording the results of data audits based on the framework, which will provide organizations "with the means to identify, locate and assess the current management of their research digital assets."

Inventory services can bring several benefits, including collection risk assessment and storage infrastructure audits. Realizing any benefits for effective data management relies on knowledge of data holdings. Knowledge of file-level holdings and recording of life cycle events related to those files from the moment that they enter the collection and in every future action reduces future risk by storing information that can be used in discovery, assessment, and recovery if and when a failure occurs.

Identifying needed services as modular rather than monolithic has allowed the Library of Congress to research and implement each of these functions in a more nimble way, all the while planning to fit those services into a larger scheme of repository services. The integration of modular transfer and inventory services as well as

[8] http://www.springframework.org/.
[9] http://www.hibernate.org/.
[10] http://www.jboss.com/products/jbpm.
[11] http://www.eclipse.org/.

[12] http://sourceforge.net/projects/loc-xferutils/
[13] http://www.jisc.ac.uk/whatwedo/programmes/digitalrepositories2007/dataauditframework.aspx

Johnston, Leslie. "Identifying and Implementing Modular Repository Services: Transfer and Inventory." *Proceedings of DigCCurr2009: Digital Curation: Practice, Promise and Prospects;* April 1-3, 2009; Chapel Hill, North Carolina. Chapel Hill: School of Library and Information Science, University of North Carolina at Chapel Hill, 2009, pp 145-148.

workflows allows for separation of tasks based on project or collection or format needs while supporting backend data integration where required. Modules can be idependently reimplemented in the future when the need arises. This also allows for extensions to services and functionality that we have not yet even considered, let alone planned for.

But do these services make up a repository? Looking at the *OAIS Reference Model* (CCSDS, 2002) and *Trustworthy Repositories Audit & Certification: Criteria and Checklist* (OCLC and CRL, 2007), we can consider the sections of the "Audit Checklist" at a very high level. Section A covers issues of administrative responsibility, organizational viability, financial sustainability, and procedural accountability. Section C covers criteria for a secure and trusted infrastructure. Section B covers the digital object management responsibilities of a repository. It is in this area that the modular work that the Library is undertaking in Deposit, Transfer, Transport, and Inventory can be categorized.

These modular services do not equate to everything needed to call a system a repository. There are only detached end-user discovery and delivery applications. Descriptive metadata is not yet tracked with the media files. There are currently no granular rights and access policies nor means to enforce them. Preservation monitoring is not yet in place. But there is a set of services that equate to many aspects of "ingest" and "archiving" – the registry of a deposit activity, the controlled transfer and transport of files, and an inventory system that can be used to track files, record events in those files' life cycles, and provide basic file-level discovery and auditing. Through the Inventory tools we expect to be able to provide persistent access at a file level. In other words, it may not yet be a full-blown repository, but is the first stage in the development of a suite of tools to help the Library ensure long-term stewardship of its digital assets.

6. REFERENCES

[1] Anderson, Martha. 2008. Evolving a Network of Networks: The Experience of Partnerships in the National Digital Information Infrastructure and Preservation Program. The International Journal of Digital Curation (July 2008: Volume 3, Issue 1). http://www.ijdc.net/ijdc/article/view/59/60.

[2] Jones, Sarah, Ball, Alex & Ekmekcioglu, Çuna. The Data Audit Framework: a first step in the data management challenge. The International Journal of Digital Curation (November 2008: Volume 3, Number 2, pp. 112-120). http://www.ijdc.net/ijdc/article/view/91/109.

[3] Littman, Justin. 2006. A Technical Approach and Distributed Model for Validation of Digital Objects. D-Lib Magazine (May 2006: Volume 12, Number 5). http://www.dlib.org/dlib/may06/littman/05littman.html.

[4] Littman, Justin. 2007. Actualized Preservation Threats: Practical Lessons from Chronicling America. D-Lib Magazine (July/August 2007: Volume 13, Number 7/8). http://www.dlib.org/dlib/july07/littman/07littman.html.

[5] Littman, Justin. 2009. A Set of Transfer-Related Services. D-Lib Magazine (January/February 2009: Volume 15, Number 1/2). http://dlib.org/dlib/january09/littman/01littman.html.

[6] Management Council of the Consultative Committee for Space Data Systems. 2002. Reference Model for an Open Archival Information System (OAIS). CCSDS 650.0-B-1, Blue Book, January 2002. http://public.ccsds.org/publications/archive/650x0b1.pdf.

[7] Online Computer Library Center, Inc. and the Center for Research Libraries. 2007. Trustworthy Repositories Audit & Certification: Criteria and Checklist. (2007). http://www.crl.edu/PDF/trac.pdf.

Johnston, Leslie. "Identifying and Implementing Modular Repository Services: Transfer and Inventory." *Proceedings of DigCCurr2009: Digital Curation: Practice, Promise and Prospects;* April 1-3, 2009; Chapel Hill, North Carolina. Chapel Hill: School of Library and Information Science, University of North Carolina at Chapel Hill, 2009, pp 145-148.