Developing an Institutional Repository Using DigiTool

Journal:	*The Electronic Library*
Manuscript ID:	TEL-Feb-2010-0015.R1
Manuscript Type:	Article
Keywords:	DigiTool, digital repository, Colorado State University Libraries, institutional repository, technical aspects

Developing an Institutional Repository Using DigiTool

Abstract

Purpose - This article aims to inform library professionals on technical issues relating to implementing and using DigiTool, proprietary software by Ex Libris, to develop an institutional repository (IR).

Design/methodology/approach - This article describes Colorado State University Libraries' experience to date in developing an IR using DigiTool. Topics discussed are based on our processes and workflows, and include local customization; metadata and object ingest; implementation of handles; incorporation with web discovery; and management of statistical data.

Findings - We consider DigiTool a powerful, complex, and relatively mature out-of-box IR platform that fulfills our needs to establish and maintain an IR.

Originality/Value - The experiential information and technical details on implementing and using DigiTool will be valuable to institutions who are interested in adopting this product for a similar purpose.

Keywords Colorado State University Libraries, digital repository, DigiTool, institutional repository, technical aspects

Paper type Technical paper/Case Study

1. Introduction

Institutional repositories (IR) have recently become a fast-growing area of academic institutions' information landscape. IR provides open access to valuable research and historical materials worldwide, and is a useful promotional tool for universities. In late 2006, Colorado State University (CSU) identified building an IR as one of the University's strategic directions, in response to the open-access movement in scholarly communication and the need of a central place for CSU scholarship and history. Leadership of this effort was assigned to the University Libraries (CSUL).

Goals of the CSU IR include:

- Highlight CSU faculty and student research
- Preserve and make accessible CSU intellectual assets and institutional memory
- Increase CSU's visibility to the world

Assuming this important task, CSUL evaluated several software options, including DSpace, Digital Commons, DigiTool, Fedora, Hive, and Symposia. As a result, DigiTool, a digital asset management system by Ex Libris [1], was purchased in May 2007 to serve as CSU's IR platform. The reasons for choosing DigiTool include:

- It is a relatively mature product supported by a vendor, which requires less investment from local IT.
- It supports established file formats and metadata standards, and provides a pleasant user interface with desired features, such as JPEG2000 viewer, METS viewer, full-text searching, and user deposit.
- CSUL was already using other Ex Libris products, such as SFX and MetaLib. Integration with these products will be easy if needed.

From May 2007 to March 2008, CSUL's Digital Repositories Services (DRS) staff and Research and Development Services (R&D) staff worked closely together to implement DigiTool. Tasks included installing and configuring the system, customizing the repository web interface, implementing handles [2], testing metadata and object ingest workflows, testing user deposit modules, setting up a local test server, and others. Since the repository's official launch in March 2008 to the completion of this article in December 2009, nearly 15,000 digital objects have been made accessible online [3], including electronic theses and dissertations (ETD), faculty publications, student research posters, conference proceedings, journal publications, publications of CSU-affiliated centers and research institutes, and archival documents and images.

Prior to this endeavor, CSUL had adopted and used CONTENTdm [4] as its digital asset management system since 2001. From 2001 to 2007, CSUL created and made accessible online nearly 4,000 digital objects using CONTENTdm. These include a wide range of materials, such as the International Poster Collection, Garst Wildlife Photographic Collection, Germans from Russia Collection, Colorado's Waters Digital Archive, and Celebrate Undergraduate Research and Creativity Digital Showcase [5]. Two collections of a small number of items from these legacy collections have been migrated into the DigiTool and we anticipate more items to be migrated in the future.

There is a small amount of literature describing or mentioning tests and evaluation of DigiTool in developing institutional repositories or digital initiatives. As the first implementer in the United Kingdom, Liverpool John Moores University (LJMU) (Stevenson and Hodges, 2008) explored use of DigiTool in 2005 to create a university digital repository. They carried out several trial projects to test the system capabilities of DigiTool, exchanged information with Ex Libris, and found the product suitable for their purpose. They noted that the advantages of choosing DigiTool over open source software were guaranteed availability of customer support from Ex Libris and the purchase of one single platform for managing the full range of digital collections at LJMU including documents, images, audios, and videos. McGill University (Park *et al.*, 2006) tested DigiTool as the system platform for creating "an Open Archives Initiative (OAI) compliant electronic thesis model that will be linked to the university's institutional repository" in its "storage, cataloguing, and dissemination capability". However, this paper focused on describing workflows of their ETD initiative and did not report the test results with DigiTool. In addition, two articles (Lynch and Lippincott, 2005; Kennan and Kingsley, 2009) listed DigiTool as one of the less common IR systems in the United States and Australia, respectively.

This paper describes CSUL's experience of using DigiTool to develop an IR. The discussions are divided into the following sections: DigiTool overview, local customization, implementation of handles, metadata and object ingest, usage statistics generation, integration with CSU Discovery, and conclusions.

2. DigiTool Overview

DigiTool is a complex digital asset management system which allows institutions to create, manage, and preserve online-accessible digital collections. DigiTool's system architecture includes back-end databases, web services/components, and a client-server module that works on the Windows system.

The majority of DigiTool functions can be performed in DigiTool's six primary web-based modules: Resource Discovery, Management, Approver, Collection Management, Deposit, and Web Ingest.

- The Resource Discovery module is the public web interface where end users access digital collections and objects.
- The Management module consists of tasks for repository management, such as system maintenance and cleanup, editing configuration files, creating user deposit forms, adding/editing Dublin Core (DC) metadata fields, generating/publishing handles, and creating reports.
- The Approver module allows staff to review user deposits and approve or decline submitted materials.
- The Collection Management module allows staff to create and organize virtual collections for end user access.
- The Deposit module is intended for use by end users to submit their materials.
- The Web Ingest module is used by staff to upload digital objects and metadata into DigiTool.

In addition, DigiTool provides a Windows-based application called "Meditor" (Metadata Editor). Meditor is designed for performing back-end staff functions on individual PCs and includes all the above-mentioned functions with additional functionalities in metadata and object management.

Depending on whether user deposit is involved, workflows in DigiTool can be either:

- User deposit > staff approval > staff ingest > repository index > user access; or
- Staff ingest > repository index > user access

The most common scenario of user deposit for us is an ETD collection in which graduating students directly submit their works into the repository. Because the ETD submission and approval process is still in its beginning phase, we haven't yet involved DigiTool user deposit module in our workflow. Our current procedure is that the DRS staff members receive student submissions in electronic format via our ETD web site [6], create metadata, and directly ingest into the repository. However, we anticipate involving the DigiTool user deposit module in the near future.

"Digital entity" is a key term in the DigiTool language. A digital entity is stored in the DigiTool back-end repository and is comprised of several components: a persistent identifier called "PID", a control section with various attributes (label, note, entity type, usage type, etc.), metadata of various types (administrative, descriptive, technical, preservation), data stream(s), and relations with other digital entities ("Manifestations", "Includes", "Part of"). Digital Entities in DigiTool may be one of the following usage types: ARCHIVE, VIEW, THUMBNAIL, or INDEX. A digital object may consist of one or more digital entities. For example, a PDF document normally consists of the following three manifestations, each a digital entity itself: the PDF (VIEW), an image in JPG format (THUMBNAIL), and a full-text document in HTML format (INDEX).

DigiTool can generate both simple and complex objects. A simple object has no internal structures, such as a student's thesis in a single PDF document (example: http://hdl.handle.net/10217/16266). A complex object has internal structures, such as an issue of a journal that consists of a front cover, preliminary pages, articles, and a back cover (example: http://hdl.handle.net/10217/22042). There are multiple ways to create single and complex objects in DigiTool, which are described in the Metadata and Object Ingest section of this article.

Metadata is a crucial component of a digital repository in terms of facilitating resource discovery. DigiTool supports various established descriptive, technical, and preservation metadata standards, including Dublin Core (DC), Machine-Readable Catalog Records (MARC), Metadata Object Description Schema (MODS), Metadata Encoding and Transmission Standard (METS), NISO Technical Metadata for Digital Still Images (NISO), and PREMIS Preservation Metadata (PREMIS). In addition to a full set of basic and qualified DC fields, DigiTool allows the definition of local DC fields for specific collection needs. For example, we have added Degree Name, Degree Grantor, Department, Advisor, Committee Members, and others for the ETD collection in order to conform to the Networked Digital Library of Theses and Dissertations (NDLTD)'s ETD-MS [7]. DigiTool also supports adding access rights metadata governing digital objects' view, edit, and ownership rights. DigiTool stores all metadata in XML format, regardless of its original formats.

Ingest, meaning the uploading of digital files and metadata into the repository, is a crucial process executed by the DRS staff. DigiTool supports various ingest approaches, based on object and metadata types. Manifestations and technical metadata may be generated during the ingest process by DigiTool. Post ingest, digital objects and metadata are searchable via the Resource Discovery module the next day after system update. In the end, a digital object and its metadata are delivered to end users via a DigiTool web dispatcher called "object viewer".

DigiTool supports both a Google-like simple keyword search and advanced search options. Digital objects and metadata in DigiTool are discoverable by common search engines such as Google, Yahoo, and Bing. Collections in DigiTool are organized in a tree structure for browsing and can be created in one of the following three types: node (which contains sub-collections), itemized (to which individual items are linked manually), and logical (which is updated by the system automatically based on a predefined search). All of our collections are node or logical. We have implemented logical identifiers that are manually assigned by DRS staff to all DigiTool items in order to fully utilize DigiTool's logical collection feature. For example, an item with a logical identifier "ETDF20091000001FRWS" will automatically appear in a collection with a predefined search, "ETDF2009******FRWS" once it is indexed. Details of logical identifiers are discussed in the CSU Core Data Dictionary (version 1.1) [8].

Working with DigiTool requires close collaboration between librarians and IT staff. A system administrator is required to maintain all system functions, such as configurations, updates, and creating/updating staff accounts. The DigiTool system is periodically updated with service pack releases.

3. DigiTool Customization

Various customizations had to be completed during our local implementation of DigiTool. Among them, two of the most important customizations are for the repository web interface and the metadata full view.

Repository Web Interface

Customizing the DigiTool web interface is simple, but we intentionally minimized customization to avoid repeated work. This is because when new DigiTool service packs are installed, some customized settings may be overwritten. Most institutions implementing DigiTool take a similar approach, which result in similar repository interfaces. Figure 1 shows a comparison of the repository home pages of the CSU Digital Repository, Center for Jewish History Digital Collections, Boston College eScholarship, and Publication of Archival Library and Museum Materials by State University Libraries of Florida.

TAKE IN FIGURE 1

From this example, we can see that all sites have customized color schemes, menus, search buttons, headers, and footers; however, the main sections that provide search and collection browsing functionalities are similar.

Web interface customization in DigiTool is mainly controlled by a CSS file. When we made the customization, we followed rules such as no changes of file names, no changes for template structure, no removal of placeholders across sections, and no changes of section names. The following two examples are CSU Digital Repository's custom styles for hyperlinks and page titles.

Example 1: Define Hyperlinks

```
a.TB:link, a.TB:visited, a.TB:hover {
color:#13694E;
font-family:'Arial Unicode MS',TAHOMA,ARIAL,VERDANA,sans-serif;
font-size:80%;
height:18px;
padding-bottom:2px;
padding-left:6px;
padding-right:6px;   }
```

Example 2: Define Font style for Page Titles

```
.PageTitle {
border-bottom:1px solid #66AA55;
font-family:'Arial Unicode MS',TAHOMA,ARIAL,VERDANA,sans-serif;
.PageTitle #browsetab {
 background-color:#66AA55;
 color:white;
 font-size:80%;
 font-weight:bold;
 padding:0 15px;  }
```

Repository Metadata Full View

DigiTool has four approaches to display descriptive metadata: brief view, table view, full view, and object viewer. Metadata displayed in the object viewer is a complete list of descriptive metadata associated with an object and is displayed in its original format (DC, MARC, or MODS). The metadata fields chosen to display in the other three views can be locally configured, especially the metadata full view. The metadata full view is an intermediary display of fuller metadata that facilitates user assessment of the content of a resource prior to actual access. Links to our repository items retrieved by common search engines direct to the metadata full view in DigiTool. Configuration of the DigiTool metadata full view applies to all items across collections. Thus, we took careful considerations when deciding what to display in our metadata full view and the process involved discussions with the librarians from the College Liaison Services and Metadata and Preservation Services (MPS). As a result, only metadata fields that we consider crucial to user assessment of resources are included (see http://hdl.handle.net/10217/28672 for an example of CSU ETD metadata full view). In addition, the DigiTool metadata full view can only appear in the DC format; therefore, other metadata formats must be mapped to DC in order to display correctly in the full view. DigiTool provides default mappings of other metadata formats to DC in its backend databases, although we also referred to the Library of Congress MARC to Dublin Core Crosswalk [9] and made some additional modifications in our local MACR to DC mapping for accuracy and local needs (see http://hdl.handle.net/10217/4200 for an example of MARC metadata full view).

4. Implementation of Handles

"The Handle System is a technology specification for assigning, managing, and resolving persistent identifiers (PI) for digital objects and other resources on the Internet." [10] In DigiTool, a handle is referred to as PI or handle PI but we normally call it "handle". In our repository, a handle is presented as a URL that links to a digital object of VIEW type. A handle is designated to only one digital object and it is permanent. This means one object will have the same URL regardless of any change of hosting server or operating system. A handle, for example, http://hdl.handle.net/10217/1553, consists of three parts, a prefix (i.e. hdl.handle.net), an institution ID (e.g. 10217), and an object ID (e.g. 1153). Prior to DigiTool implementation, we requested an institutional ID from HANDLE.NET and installed the handle system outside the DigiTool framework.

Implementing handles is more complicated than web interface customization. We were one of the first institutions to implement handles in DigiTool, and during the process we worked closely with Ex Libris technical support staff. In this article, we list the main steps we have taken. For security purposes, we replaced actual paths and file names in the examples with ***.

General Configurations

(1) pi_profiles_rules.xml

The first file we modified is pi_profiles_rules.xml which specifies rules of generating handles. This file has two main sections, profile settings and rule settings. The profile settings section describes how handles should be built and the rule settings section determines which profile to use. For most DigiTool users, they only need to change pi_prefix from the default to their own handle prefix. For example, we changed ours from the DigiTool demo prefix "12345" to "10217" on line 3.

Line 1: <pi_profile name="handle_profile">
Line 2: <pi_type>handle</pi_type>
Line 3: <pi_preffix required="true">10217</pi_preffix> //omitted code

The rule section defines parameters for generating handles. By default, the <rl:de_rule> section specifies that only digital entities of the VIEW or VIEW (PRIMARY) types can have handles and their associated manifestations, such as thumbnails and full text, have no handles. At CSU, we kept the default rules.

(2) pi_publisher_rules.xml

Next we modified pi_publisher_rules.xml which also has both profile and rule sections. For the rule section, we only changed the default <pi_prefix> to CSU's prefix (as shown in line 4). For the profile section, we changed the <parameter name="authentication.public.handleName"> to CSU's prefix and administrator password (as shown in line 7 and 9).

```
<!-- CSU  Digital Repository's rules section example -->
Line 1: <rl:de_rule name="view_handle_publisher">
Line 2: <usage_type>VIEW</usage_type>
Line 3: <entity_type/><file_extension/><relation_type></relation_type><preservation_level/>
Line 4: <pi_prefix>10217</pi_prefix>  //omitted code
```

```
<!-- CSU Digital Repository's profile section -->
Line 5: <pi_publisher><!-- No changes in this section -->
Line 6: <params>
Line 7: <parameter name="authentication.public.handleName">0.NA/10217</parameter>
Line 8: <parameter name="authentication.public.privateKeyFile">/***/admin***.bin</parameter>
Line 9: <parameter name="authentication.public.passphrase">xxxxx</parameter> //omitted code
```

(3) hdltool.ini

The hdltool.ini provides default authorization information. There are two configurations for a public and a private key authorization. At CSU, we use a private key.

```
Line 1: {"SecIndex" = "300"
Line 2: "PrivKey" = "/***/adm***.bin"
Line 3: "PubHandle" = "0.NA/10217"  }
```

(4) handleConfig.properties

The handleConfig.properties is DigiTool's key for locating and communicating with the appropriate handle server. We replaced the default handle prefix to CSU's handle prefix and reset the default authorization values.

```
Line 1: authentication.public.handleName=0.NA/10217
Line 2: authentication.public.privateKeyFile=/***/adm***.bin
Line 3: authentication.secret.handleName=10217/ADMIN
Line 4: authentication.secret.pass=***
```

Local Configurations

For our repository, we decided to display a handle in DigiTool's metadata full view along with other DC fields and label it "Bookmarkable URL". The www_r_silo_conf.xml file configures metadata display in the brief view, table view, and full view. It specifies which field to display and its label. We modified one line in <results_full> section, <name lng="ENG">Bookmarkable URL</name>.
By default, the Bookmarkable URL and other URLs in metadata are strings of text. In order to make them hyperlinks, we added two functions in script.js. One function specifically converts

handles into hyperlinks, and the other converts text strings starting with http or https in other metadata fields into hyperlinks. Other modified files include results-full-body and digital_entity_urls_template.xml. The source code can be requested from CSUL.

Implementation

A handle is not automatically generated post-ingest. There are three processes to make handles available in DigiTool: generate handles, publish handles, and repository harvest. There are two options to generate handles, global handle generation via the management module and create URN by PID via Meditor. At CSU, we use global handle generation via the management module because handles can be generated for a specified range of PIDs. After handle generation, we run a process to publish handles in the Management module. This process links a published handle on the handle server, which is outside of the DigiTool framework, to the corresponding digital entity on the DigiTool server. Last, the DigiTool repository must be reharvested before handles are available in the Resource Discovery. At CSU, we have set up an automated process in which the repository is harvested daily during low-traffic times.

5. Metadata and Object Ingest

DigiTool supports multiple approaches for ingesting metadata and digital objects, based on metadata (DC, MARC, MODS, or METS) and object types (simple or complex object). We currently have two types of descriptive metadata in the repository, DC and MARC. DC is flexible, easy to create, and works for all common types of electronic resources. We use DC when new metadata needs to be created. MARC is a standard metadata format for print library resources. In order to reduce metadata creation cost, we repurpose MARC when the electronic resources have corresponding prints that have been previously cataloged locally. Staff involved in metadata creation includes the members from DRS, MPS, and Archives and Special Collections (Archives), while ingest is exclusively executed by the DRS staff. The major file formats in our repository at present are documents in PDF and images in JPEG2000, although we do have seventeen videos and audios in various formats. Based on our experience to date, we have identified the following common types of metadata creation and ingest approaches to be discussed in detail.

DC XML and Associated File Streams

DC XML is Dublin Core metadata created in Extensible Markup Language (XML) [11] format. DC XML is easy to create and can be used to ingest one or multiple files at a time. DC XML templates can be created for materials of a similar nature (for example, ETDs, faculty publications, conference proceedings, journals, or student posters), which facilitate metadata creation and staff training. We create new metadata in DC XML when we have relatively small to medium-scale collections. In addition, DC XML can be used in conjunction with METS XML as descriptive metadata (see METS XML section). XML files are essentially TXT files that can be created and edited using any text editor; however, we recommend using a professional XML editor. This is because professional XML editors provide much friendlier working environment and offer features such as XML validation and transformations among different metadata schemas. At CSUL, we have chosen to use <oXygen/> XML Editor [12]. The academic version of <oXygen/> can be acquired at a very low cost and it offers an excellent set of functionalities.

MARC XML and Associated File Streams

We repurpose MARC when we have corresponding catalog records for digitized materials. Examples include Atmospheric Science Papers (see http://hdl.handle.net/10217/27), Colorado Agriculture Bibliography publications (see http://hdl.handle.net/10217/4458), and Colorado Water Institute publications (see http://hdl.handle.net/10217/3167). When repurposing MARC, we follow these procedures:

1. Catalogers in MPS extract MARC records from our cataloging system, the Millennium ILS powered by Innovative Interfaces.
2. Catalogers use MarcEdit [13], open-source software, to edit the exported MARC data. This includes adding filenames and format (in MARC 856 field subfield u and q, e.g. "856 00 $u 0211_Bluebook.pdf $q application/pdf"), logical identifiers (in MARC 592 field subfield a, e.g. "592 $a FACFATMS100042BLUE") and copyright information (in MARC 506 or 540 field subfield a, e.g. "506 $a http://www.acns.colostate.edu/?page=copyright"). Adding filenames in the MARC 856 field is required for linking individual MARC records to their respective file streams for this type of ingest.
3. Catalogers use MarcEdit to transform the edited MARC data into MARC XML.
4. Catalogers notify DRS staff available MARC XML files.
5. DRS staff members validate MARCXML files in <oXygen/>XML editor and ingest them into DigiTool with associated file streams.

Similar to DC XML, MARC XML can be used to ingest either one or multiple files at a time and be used in conjunction with METS XML as descriptive metadata (see METS XML section). Repurposing MARC reduces our cost of metadata creation. It is also an efficient way to make digitized materials available via the repository as quickly as possible.

CSV File

In large-scale digital projects when many staff members are involved in new metadata creation and in migration projects when we move metadata and digital objects from CONTENTdm to DigiTool, we use CSV files for ingest. Examples of large-scale digital projects include the University Historic Photograph Collection (see http://hdl.handle.net/10217/35514) and the Water Resources Archive (see http://hdl.handle.net/10217/31642). Examples of migration projects include the Celebrate Undergraduate Research and Creativity (CURC) Showcase (see http://hdl.handle.net/10217/24128) and the Information Science and Technology Center (ISTeC) Student Research Posters (see http://hdl.handle.net/10217/530).

A comma-separated value (CSV) file is a common and simple text file that stores tabular data. Each line in a CSV file corresponds to a row in a table and each column corresponds to a field that is separated by commas. "CSV files are often used for moving tabular data between two different computer programs, for example, between a database program and a spreadsheet program" [14]. CSV files storing metadata, used in conjunction with their mapping XML according to DigiTool specifications (see Appendix A), are an option of batch ingest that DigiTool provides. Procedures of CSV ingest for large-scale projects and migration projects are slightly different.

In large-scale digital projects, we follow these procedures:

1. MPS staff members create metadata in MS Excel. The first row in the Excel file contains names of metadata fields outlined by a metadata librarian, which varies by project. The following rows contain metadata records. The reasons we create metadata in MS Excel are that Excel provides many efficient text editing features and most of our metadata staff members are familiar with the application.

2. DRS staff members create mapping XML files based on the order of the metadata fields outlined in the Excel files. The purpose of mapping XML files is to interpret the tabular data stored in CSV for DigiTool processing.

3. MPS or DRS staff members convert the Excel files into CSV files using the "Save As > CSV (Comma delimited) (*.csv)" function in Excel.

4. DRS staff members ingest the CSV files, mapping XML files, and their associated file streams into DigiTool.

In migration projects, we follow these procedures:

1. MPS staff members export metadata from CONTENTdm in tab-delimited TXT files.

2. DRS staff members create mapping XML files based on the order of the metadata fields in the TXT files.

3. DRS staff members convert the tab-delimited TXT files into CSV files using a local program.

4. DRS staff members ingest the CSV files, mapping XML files, and their associated file streams into DigiTool.

CSV ingest is a very efficient way to batch upload metadata and file streams into the repository. Using such an approach, we have made available online nearly 10,000 images in the University Historic Photograph Collection and nearly 600 images and documents in the Water Resources Archive within a year.

Complex METS Objects

The Metadata Encoding and Transmission Standard (METS) schema "is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed using the XML schema language…" [15]. METS XML supports creation of complex digital objects with internal structures, such as books, journals, and archival aggregations. A few examples of our complex METS objects are the CSU Board of Governors meeting materials (see http://hdl.handle.net/10217/18499), Journal of Student Affairs (see http://hdl.handle.net/10217/22042), and Rocky Mountain Farmers Union historical photographs (see http://hdl.handle.net/10217/3691). At this time, we have 366 METS objects in our repository. Complex METS objects can be generated either by DigiTool internally using ingest types specified below or by creating METS XML files offline and ingesting them with associated file streams.

(1) Generate METS Objects Internally by DigiTool

DigiTool can create complex METS objects internally by ingest type "File stream(s) that will become part of one parent record" or "File stream(s) utilizing DigiTool file naming convention". Using the "parent record" approach can only generate METS objects with physical structural maps, while using the "file naming convention" approach can generate METS objects with either physical or logical structural maps. The difference of a physical and a logical structural map lies in the navigation structure that they provide. A physical structural map provides flat navigation and a logical structural map provides hierarchical navigation within the METS object viewer (see Figure 2).

TAKE IN FIGURE 2

Using these approaches, descriptive metadata will need to be added and labels in structural maps will need to be edited later in Meditor. These ingest approaches are not limited by file format; however, they require more time in structural map editing, especially when a METS object consists of a large number of files. We use these approaches when we need to create METS objects of mixed file formats, for example, when a METS object consists of both PDF files and JPEG2000 images.

(2) Create METS XML Files Offline and Ingest with Associated File Streams

One of our R&D staff members developed a utility based on the METS object generation procedures that we acquired from the Center for Jewish History, a fellow institution that uses DigiTool. The utility works specifically with DigiTool and helps the DRS staff automatically generate raw METS XML files offline when a METS object consists of either all PDF files or all JPEG2000 images. The utility is available upon request to readers of interest.

The procedures of creating METS objects using this utility are:

1. Collect all files that will be included in the METS object and group them into one folder.
2. Number the files so that all files are listed in a desired sequence for end user navigation. For example:

   101_Cover.pdf
   102_Article1.pdf

103_Article2.pdf
104_Article3.pdf

3. Ingest these files into DigiTool as file streams with no relationships. During this ingest, the files are uploaded to the DigiTool server and manifestations of the files (e.g. thumbnail, full-text) may be generated by DigiTool as needed.
4. Run the METS utility. Select corresponding options according to the file and metadata format. Based on local needs, technical metadata may be extracted. The utility generates a raw METS XML file from the previous ingest.
5. Edits are made to the raw METS XML file, including providing correct information in the METS header, adding descriptive metadata in either DC or MARC XML, and completing the structural map. Structural maps can be either physical or logical (See Appendix B).
6. Ingest the completed METS XML file and the associated file streams generated from the first ingest.

The advantage of creating METS XML files offline is the flexibility and efficiency it provides to structural map creation, especially with logical structural maps.

6. Integration of DigiTool with CSU Discovery

We have several additional custom interfaces for searching items in DigiTool utilizing the Ex Libris DigiTool API [16], such as the University Historic Photograph Collection web site (http://lib.colostate.edu/archives/uhpc/) and CSU Discovery (http://discovery.library.colostate.edu/). CSU Discovery is an integrated search tool that allows end users to search our libraries' online catalog, the digital repository, and CSU web sites at one location (see Figure 3).

TAKE IN FIGURE 3

This search tool is built on VuFind [17], an open-source library OPAC project using PHP as a web component and Solr [18] as the backend database. To incorporate DigiTool data into CSU Discovery search, metadata is harvested via the Ex Libris DigiTool API and the output is in XML format. The harvest is conducted five days a week and the results are transformed for insertion into the Solr database. With CSU Discovery, end users can search items in DigiTool directly via the library home page

7. Managing Statistical Data

Statistical Data at the Item Level

In the DigiTool management module, there are two report functions. The Collection Distribution Summary report lists the top-level collections available in the repository and their distributions by the number of files in collections and the size of collections in MB. The Delivery Usage Statistics report shows the number of user requests and delivered requests on items for each top-level collection. For example, the report tells us that from June 10, 2009 to Nov 13, 2009, users made 219 visits to our Theses and Dissertations collection and 71 documents in this collection have been viewed. The report can also list what titles have been viewed and the number of requests and delivered requests on each title.

Besides the DigiTool report functions, we developed our own statistics page at the item level for additional monitoring of repository usage, which is written in PHP. It displays all objects of VIEW type in our repository. Information on each item includes its PID, handle, number of delivered metadata full views, number of delivered object views, title, creator, subject, date, identifier, and so on. The display can be sorted and searched by these attributes (see Figure 4).

TAKE IN FIGURE 4

In Figure 4, the handle number, 556, corresponds to the actual handle, http://hdl.handle.net/10217/556. For staff reference, we also have a column, Open Archives Initiative (OAI), linking to an item's OAI record in XML format that is created for external harvesting. Our statistics only include hits from non-library-staff IP addresses in order to help us monitor usage exclusively by outside end users. We also use this page to keep track of collection usage. By searching the logical identifiers, for example, by searching "FACFATMS" in the Identifier field, we can gather all faculty publications from the Atmospheric Science department and obtain a summary of usage statistics on this collection.

To create this page, our system administrator created a Perl script file to collect data from an Apache server log file which excludes hits from staff and web crawlers based on IP addresses, and output this data into a CSV file. A PHP file was then created to read the CSV file and display the results on the web. The code for creating this page can be made available to the audience of this article upon request.

Statistical Data at Site Level

At CSUL, we use Google Analytics [19] to track and analyze our main components' traffic on the library web site. These main components include the online catalog, the Digital Collections page, and the CSU Digital Repository. An individual profile was created for each component in Google Analytics.

Figure 5 is a screenshot of the Google Analytics administrator interface. This interface can show the total number of site visits, average visiting time, and traffic trends for each profile by day, week, month, or year.

TAKE IN FIGURE 5

When clicking a "View report" link in the administrator interface, we can view a site's traffic trends, site usage, visitors' overview, traffic sources overview, and other statistical information. Figure 6 shows that from May 26, 2009 to June 25, 2009, there were a total of 5,389 non-staff visits to the CSU Digital Repository and 75.80% of the visits were redirected from search engines.

TAKE IN FIGURE 6

When clicking on the "View report link" in the Traffic Sources Overview section, we can view a breakdown of repository usage by different sources. For example, we can further tell from Figure 7 that out of the 4,085 visits redirected from search engines from May 26, 2009 to June 25, 2009, 3,715 were from Google.

TAKE IN FIGURE 7

8. Conclusions

From March 2008 to the end of 2009, CSUL has been able to make over 15,000 digital objects available online with DigiTool. Working with DigiTool requires close collaboration between librarians and IT professionals because of the complexity of its system architecture. We are most satisfied with DigiTool in its completeness of IR functions, support of established metadata standards and file formats, multiple ingest approaches that offer flexibility and efficiency, and the capability to integrate with our Discovery Tool and external search engines and harvesters. We found it beneficial to set up a test server that has helped us to investigate DigiTool system functionalities and ensure the accuracy and soundness of each system update. DigiTool users form a small yet close community, and we often find our questions answered via the DigiTool listserv by helpful colleagues. In conclusion, we consider DigiTool a powerful, complex, and relatively mature out-of-box IR platform that fulfills our needs to establish and maintain an IR.

Appendices

Appendix A: Examples of Metadata and Mapping XML for CSV Ingest

(1) A simple CSV file storing metadata:

File Name,Title,Author,Subject
Image1.jp2,Ocean,"James, Thompson",Ocean
Image2.jp2,Flower,"James, Thompson",Flowers
Image3.jp2,Sunset,"James, Thompson",Sun -- Rising and setting

(2) Mapping XML used in conjunction with the CSV file storing metadata:

```
<?xml version="1.0" encoding="utf-8"?>
<tm:x_mapping xmlns:tm=http://com/exlibris/DigiTool/repository/transMap/xmlbeans
start_from_line="2">
        <x_map>
                <x_source position="2"/>
                <x_target md_name="descriptive" md_type="dc">dc:title</x_target></x_map>
        <x_map>
                <x_source position="3"/>
                <x_target md_name="descriptive" md_type="dc">dc:creator</x_target></x_map>
        <x_map>
                <x_source position="4"/>
                <x_target md_name="descriptive" md_type="dc">dc:subject</x_target></x_map>
        <x_map>
                <x_source position=""/>
                <x_target>control/usage_type</x_target>
                <x_default>VIEW</x_default></x_map>
        <x_map>
                <x_source position="1"/>
                <x_target>stream_ref/file_name</x_target></x_map>
        <x_map>
                <x_source position=""/>
                <x_target>stream_ref/directory_path</x_target>
                <x_default>/</x_default></x_map>
        <x_map>
                <x_target>stream_ref</x_target>
                <x_attr>store_command</x_attr>
                <x_default>copy</x_default></x_map>
</tm:x_mapping>
```

Appendix B: Examples of METS Physical and Logical Structural Maps

(1) A physical structural map for flat navigation

```
<structMap TYPE="PHYSICAL" ID="smd001">
   <div ORDER="1" ID="Auto_Generated_Qualifier_STRUCTMAP" TYPE="image folder"
LABEL="Colorado - Brighton - Agricultural Machinery" DMDID="dmd001"
xmlns:mets="http://www.loc.gov/METS/"  xmlns:mods="http://www.loc.gov/mods/"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://www.loc.gov/METS/" >
     <div TYPE="image" LABEL="Photo 1" ORDER="1">
      <fptr FILEID="s001" /><fptr FILEID="t001" /></div>
     <div TYPE="image" LABEL="Photo 2" ORDER="2">
      <fptr FILEID="s002" /><fptr FILEID="t002" /></div>
     <div TYPE="image" LABEL="Photo 3" ORDER="3">
      <fptr FILEID="s003" /><fptr FILEID="t003" /></div></div></structMap>
```

(2) A logical structural map for hierarchical navigation

```
<structMap TYPE="Logical" ID="smd001">
  <div ORDER="1" ID="MANUAL_GENERATED_STRUCTMAP" TYPE="document" LABEL="CSU Board
of Governors 2004 meeting agendas and minutes" DMDID="dmd001"
   xmlns:mets="http://www.loc.gov/METS/"
   xmlns:mods="http://www.loc.gov/mods/"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.loc.gov/METS/" >
   <div ID="div1.1" LABEL="Chapter 1" TYPE="document" ORDER="1">
     <div LABEL="Section 1" TYPE="document" ORDER="1">
      <fptr FILEID="s001" /><fptr FILEID="t001" /><fptr FILEID="i001" /></div>
     <div LABEL="Section 2" TYPE="document" ORDER="2">
      <fptr FILEID="s002" /><fptr FILEID="t002" /><fptr FILEID="i002" /></div></div>
    <div ID="div1.2" LABEL="Chapter 2" ORDER="2">
     <div LABEL="Section 1" TYPE="document" ORDER="1">
      <fptr FILEID="s003" /><fptr FILEID="t003" /><fptr FILEID="i003" /></div>
     <div LABEL="Section 2" TYPE="document" ORDER="2">
      <fptr FILEID="s004" /><fptr FILEID="t004" /><fptr FILEID="i004" /></div></div>
   </div></structMap>
```

Notes

[1] http://www.exlibrisgroup.com/category/DigiToolOverview [accessed March 12, 2010]
[2] http://www.handle.net/ [accessed March 12, 2010]
[3] http://DigiTool.library.colostate.edu/R [accessed March 12, 2010]
[4] http://www.contentdm.org/ [accessed March 12, 2010]
[5] http://digital.library.colostate.edu/ [accessed March 12, 2010]
[6] http://lib.colostate.edu/repository/etd/ [accessed March 18, 2010]
[7] http://www.ndltd.org/standards/metadata/etd-ms-v1.00-rev2.html [accessed March 12, 2010]
[8] http://hdl.handle.net/10217/3147 [accessed March 12, 2010]
[9] http://www.loc.gov/marc/marc2dc.html [accessed March 12, 2010]
[10] http://en.wikipedia.org/wiki/Handle_System [accessed March 12, 2010]
[11] http://www.w3.org/XML/ [accessed March 12, 2010]
[12] http://www.oxygenxml.com/ [accessed March 12, 2010]
[13] http://people.oregonstate.edu/~reeset/marcedit/html/index.php [accessed March 12, 2010]
[14] http://en.wikipedia.org/wiki/Comma-separated_values [accessed March 12, 2010]
[15] http://www.loc.gov/standards/mets/ [accessed March 12, 2010]
[16] http://en.wikipedia.org/wiki/Application_programming_interface [accessed March 12, 2010]
[17] http://vufind.org/ [accessed March 12, 2010]
[18] http://lucene.apache.org/solr/ [accessed March 12, 2010]
[19] http://www.google.com/analytics/ [accessed March 12, 2010]

References

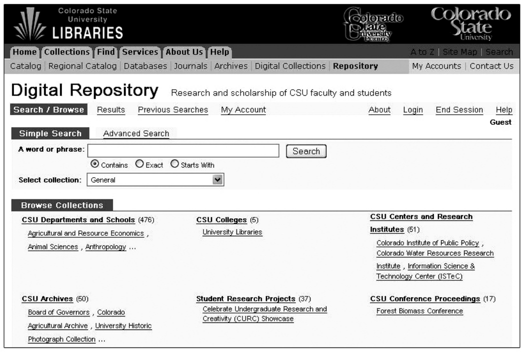
Valerie, Stevenson and Hodges, Sue (2008), "Setting up a university digital repository: experience with DigiTool", *OCLC Systems & Services*, Vol. 24 No. 1, pp. 48-50.

Park, Eun G. et al. (2007), "Electronic thesis initiative: pilot project of McGill University, Montreal", *Program: Electronic Library & Information Systems*, Vol. 41 No. 1, pp. 81-91.

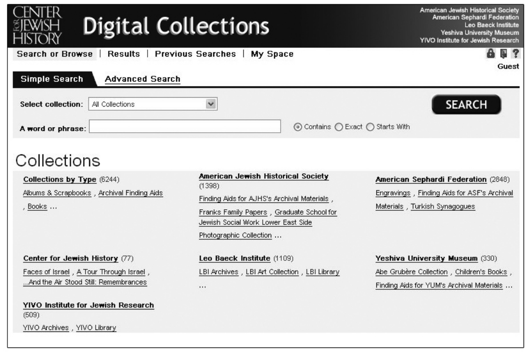
Lynch, Clifford A. and Lippincott, Joan K (2005), "Institutional repository deployment in the United States as of early 2005", *D-Lib Magazine*, Vol. 11 No. 9, http://webdoc.sub.gwdg.de/edoc/aw/d-lib/dlib/september05/lynch/09lynch.html [accessed March 08, 2010].

Kennan, Mary Anne and Kingsley, Danny A. (2009), "The state of the nation: a snapshot of Australian institutional repositories", *First Monday*, Vol. 14 No. 2, http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/2282/2092 [accessed March 08, 2010].

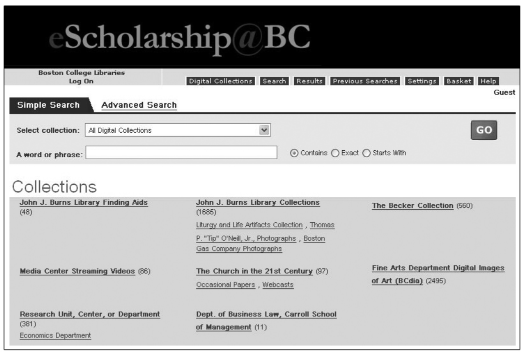
1.



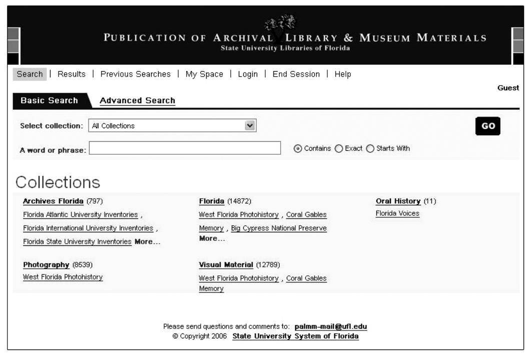
Colorado State University Digital Repository



Center for Jewish History Digital Collections

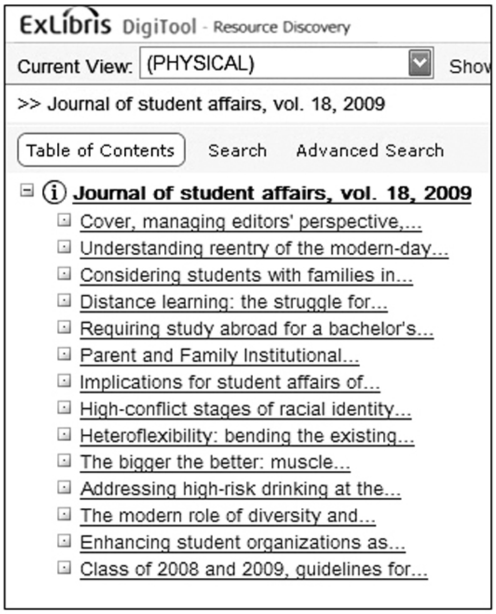


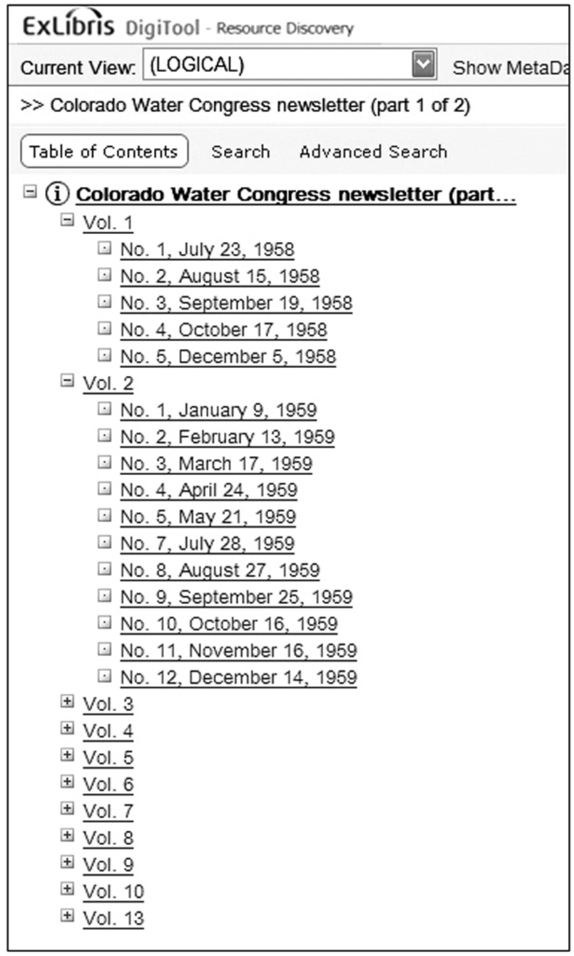
Boston College eScholarship



Publication of Archival Library and Museum Materials

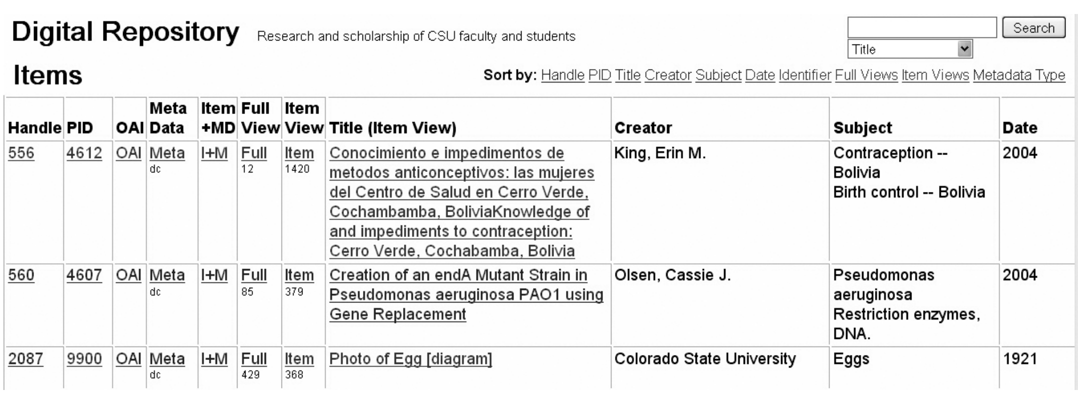
2.


ExLibris DigiTool - Resource Discovery
Current View: (PHYSICAL)
Show
>> Journal of student affairs, vol. 18, 2009
Table of Contents
Search
Advanced Search
Journal of student affairs, vol. 18, 2009
Cover, managing editors' perspective,...
Understanding reentry of the modern-day...
Considering students with families in...
Distance learning: the struggle for...
Requiring study abroad for a bachelor's...
Parent and Family Institutional...
Implications for student affairs of...
High-conflict stages of racial identity...
Heteroflexibility: bending the existing...
The bigger the better: muscle...
Addressing high-risk drinking at the...
The modern role of diversity and...
Enhancing student organizations as...
Class of 2008 and 2009, guidelines for...


ExLibris DigiTool - Resource Discovery
Current View: (LOGICAL)
Show MetaDa
>> Colorado Water Congress newsletter (part 1 of 2)
Table of Contents
Search
Advanced Search
Colorado Water Congress newsletter (part...
Vol. 1
No. 1, July 23, 1958
No. 2, August 15, 1958
No. 3, September 19, 1958
No. 4, October 17, 1958
No. 5, December 5, 1958
Vol. 2
No. 1, January 9, 1959
No. 2, February 13, 1959
No. 3, March 17, 1959
No. 4, April 24, 1959
No. 5, May 21, 1959
No. 7, July 28, 1959
No. 8, August 27, 1959
No. 9, September 25, 1959
No. 10, October 16, 1959
No. 11, November 16, 1959
No. 12, December 14, 1959
Vol. 3
Vol. 4
Vol. 5
Vol. 6
Vol. 7
Vol. 8
Vol. 9
Vol. 10
Vol. 13

3.

4.



Handle	PID	OAI	Meta Data	Item +MD	Full View	Item View	Title (Item View)	Creator	Subject	Date
556	4612	OAI	Meta dc	I+M	Full 12	Item 1420	Conocimiento e impedimentos de metodos anticonceptivos: las mujeres del Centro de Salud en Cerro Verde, Cochambamba, BoliviaKnowledge of and impediments to contraception: Cerro Verde, Cochabamba, Bolivia	King, Erin M.	Contraception -- Bolivia Birth control -- Bolivia	2004
560	4607	OAI	Meta dc	I+M	Full 85	Item 379	Creation of an endA Mutant Strain in Pseudomonas aeruginosa PAO1 using Gene Replacement	Olsen, Cassie J.	Pseudomonas aeruginosa Restriction enzymes, DNA.	2004
2087	9900	OAI	Meta dc	I+M	Full 429	Item 368	Photo of Egg [diagram]	Colorado State University	Eggs	1921

5.

Overview » lib.colostate.edu (Edit account settings)

May 19, 2009 - Jun 18, 2009
Comparing to: **Apr 18, 2009 - May 18, 2009**

All | Starred

Day | Week | Month | Year

Website Profiles + Add new profile

Name	Reports	Status	Visits	Avg. Time on Site	Bounce Rate	Completed Goals	Visits % Change	Actions
http://catalog.library.colostate.edu UA-1399660-7								+ Add new profile
catalog.library.colostate.edu	View report	✔	22,497	00:07:40	30.09%	0	-35.14%	Edit | Delete
catalog.library.colostate.edu non-library	View report	✔	17,976	00:07:12	31.07%	0	-35.98%	Edit | Delete
http://digital.library.colostate.edu UA-1399660-3								+ Add new profile
digital.library.colostate.edu	View report	✔	3,966	00:04:13	23.32%	0	-37.76%	Edit | Delete
digital.library.colostate.edu non-library	View report	✔	3,695	00:04:08	24.57%	0	-38.28%	Edit | Delete
http://digitool.library.colostate.edu UA-1399660-10								+ Add new profile
digitool.library.colostate.edu	View report	✔	7,024	00:02:28	79.61%	0	6.38%	Edit | Delete
digitool non-library	View report	✔	5,332	00:00:58	82.00%	0	0.32%	Edit | Delete

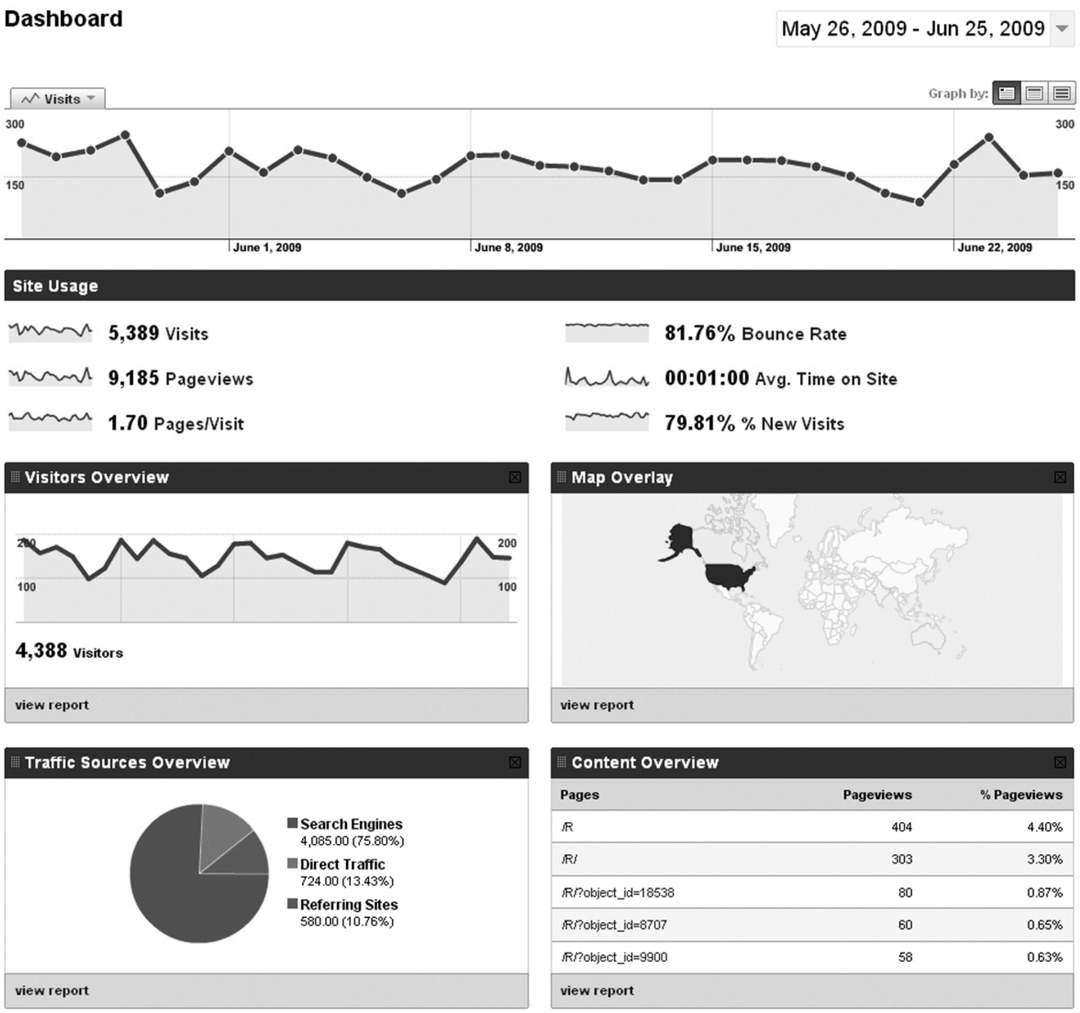
Find profile: Show rows: 10 1 of 3

Add Website Profile»
A profile allows you to track a website and/or create different views of the reporting data using filters. Learn more

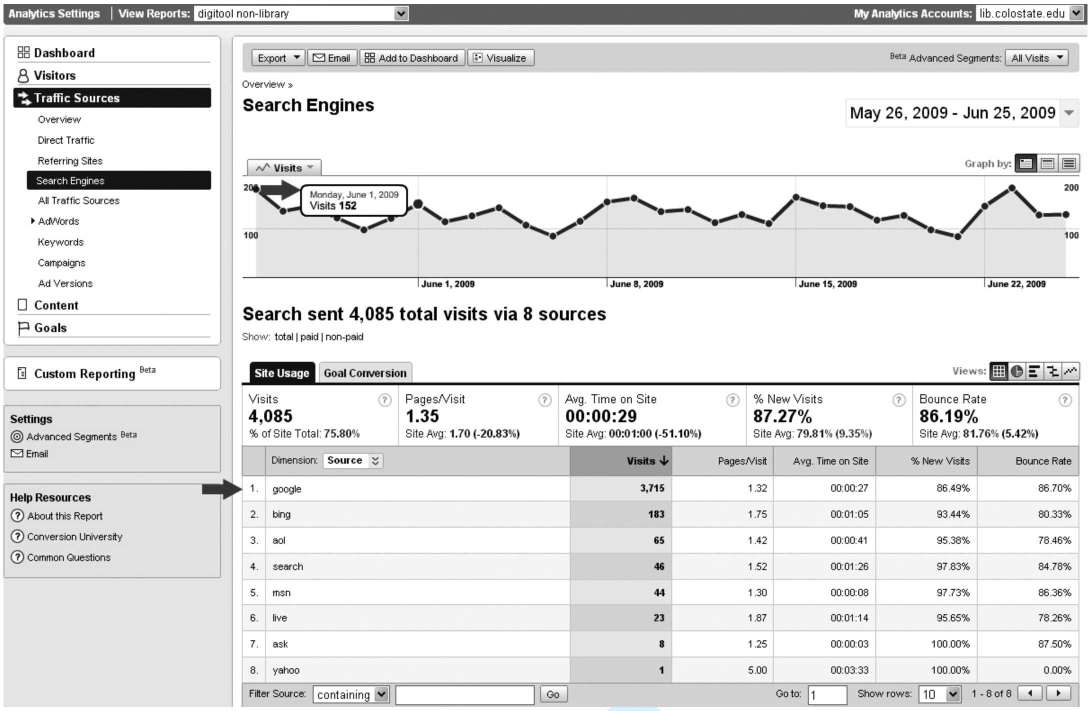
User Manager»
Number of Users: 17
Add or edit Users. Learn more

Filter Manager»
Number of Filters: 24
Filters can be used to customize the way data is displayed in your reports. Learn more

6.


Dashboard
May 26, 2009 - Jun 25, 2009
Visits
Graph by:
300
150
June 1, 2009
June 8, 2009
June 15, 2009
June 22, 2009
Site Usage
5,389 Visits
9,185 Pageviews
1.70 Pages/Visit
81.76% Bounce Rate
00:01:00 Avg. Time on Site
79.81% % New Visits
Visitors Overview
200
100
4,388 Visitors
view report
Map Overlay
view report
Traffic Sources Overview
Search Engines
4,085.00 (75.80%)
Direct Traffic
724.00 (13.43%)
Referring Sites
580.00 (10.76%)
view report
Content Overview
Pages Pageviews % Pageviews
/R/ 404 4.40%
/R/ 303 3.30%
/R/?object_id=18538 80 0.87%
/R/?object_id=8707 60 0.65%
/R/?object_id=9900 58 0.63%
view report

7.


Analytics Settings | View Reports: digitool non-library
My Analytics Accounts: lib.colostate.edu
Dashboard
Visitors
Traffic Sources
Overview
Direct Traffic
Referring Sites
Search Engines
All Traffic Sources
AdWords
Keywords
Campaigns
Ad Versions
Content
Goals
Custom Reporting Beta
Settings
Advanced Segments Beta
Email
Help Resources
About this Report
Conversion University
Common Questions
Export
Email
Add to Dashboard
Visualize
Beta Advanced Segments: All Visits
Overview »
Search Engines
May 26, 2009 - Jun 25, 2009
Visits
Graph by:
Monday, June 1, 2009
Visits 152
200
100
June 1, 2009
June 8, 2009
June 15, 2009
June 22, 2009
Search sent 4,085 total visits via 8 sources
Show: total | paid | non-paid
Site Usage
Goal Conversion
Views:
Visits 4,085 % of Site Total: 75.80%
Pages/Visit 1.35 Site Avg: 1.70 (-20.83%)
Avg. Time on Site 00:00:29 Site Avg: 00:01:00 (-51.10%)
% New Visits 87.27% Site Avg: 79.81% (9.35%)
Bounce Rate 86.19% Site Avg: 81.76% (5.42%)
Dimension: Source | Visits | Pages/Visit | Avg. Time on Site | % New Visits | Bounce Rate
1. google 3,715 1.32 00:00:27 86.49% 86.70%
2. bing 183 1.75 00:01:05 93.44% 80.33%
3. aol 65 1.42 00:00:41 95.38% 78.46%
4. search 46 1.52 00:01:26 97.83% 84.78%
5. msn 44 1.30 00:00:08 97.73% 86.36%
6. live 23 1.87 00:01:14 95.65% 78.26%
7. ask 8 1.25 00:00:03 100.00% 87.50%
8. yahoo 1 5.00 00:03:33 100.00% 0.00%
Filter Source: containing Go
Go to: 1 Show rows: 10 1 - 8 of 8

Developing an Institutional Repository Using DigiTool

Abstract

Purpose - This article aims to inform library professionals on technical issues relating to implementing and using DigiTool, proprietary software by Ex Libris, to develop an institutional repository (IR).
Design/methodology/approach - This article describes Colorado State University Libraries' experience to date in developing an IR using DigiTool. Topics discussed are based on our processes and workflows, and include local customization; metadata and object ingest; implementation of handles; incorporation with web discovery; and management of statistical data.
Findings - We consider DigiTool a powerful, complex, and relatively mature out-of-box IR platform that fulfills our needs to establish and maintain an IR.
Originality/Value - The experiential information and technical details on implementing and using DigiTool will be valuable to institutions who are interested in adopting this product for a similar purpose.
Keywords Colorado State University Libraries, digital repository, DigiTool, institutional repository, technical aspects
Paper type Technical paper/Case Study

Introduction

Institutional repositories (IR) have recently become a fast-growing area of academic institutions' information landscape. IR provides open access to valuable research and historical materials worldwide, and is a useful promotional tool for universities. In late 2006, Colorado State University (CSU) identified building an IR as one of the University's strategic directions, in response to the open-access movement in scholarly communication and the need of a central place for CSU scholarship and history. Leadership of this effort was assigned to the University Libraries (CSUL). Goals of the CSU IR include:

- Highlight CSU faculty and student research
- Preserve and make accessible CSU intellectual assets and institutional memory
- Increase CSU's visibility to the world

Assuming this important task, CSUL evaluated several software options, including DSpace, Digital Commons, DigiTool, Fedora, Hive, and Symposia. As a result, DigiTool, a digital asset management system by Ex Libris [1], was purchased in May 2007 to serve as CSU's IR platform. The reasons for choosing DigiTool include:

- It is a relatively mature product supported by a vendor, which requires less investment from local IT.
- It supports established file formats and metadata standards, and provides a pleasant user interface with desired features, such as JPEG2000 viewer, METS viewer, full-text searching, and user deposit.
- CSUL was already using other Ex Libris products, such as SFX and MetaLib. Integration with these products will be easy if needed.

From May 2007 to March 2008, CSUL's Digital Repositories Services (DRS) staff and Research and Development Services (R&D) staff worked closely together to implement DigiTool. Tasks included installing and configuring the system, customizing the repository web interface, implementing handles [2], testing metadata and object ingest workflows, testing user deposit modules, setting up a local test server, and others. Since the repository's official launch in March 2008 to the completion of this article in December 2009, nearly 15,000 digital objects have been made accessible online [3], including electronic theses and dissertations (ETD), faculty publications, student research posters, conference proceedings, journal publications, publications of CSU-affiliated centers and research institutes, and archival documents and images.

Prior to this endeavor, CSUL had adopted and used CONTENTdm [4] as its digital asset management system since 2001. From 2001 to 2007, CSUL created and made accessible online nearly 4,000 digital objects using CONTENTdm. These include a wide range of materials, such as the International Poster Collection, Garst Wildlife Photographic Collection, Germans from Russia Collection, Colorado's Waters Digital Archive, and Celebrate Undergraduate Research and Creativity Digital Showcase [5]. Two collections of a small number of items from these legacy collections have been migrated into the DigiTool and we anticipate more items to be migrated in the future.

There is a small amount of literature describing or mentioning tests and evaluation of DigiTool in developing institutional repositories or digital initiatives. As the first implementer in the United Kingdom, Liverpool John Moores University (LJMU) (Stevenson and Hodges, 2008) explored use of DigiTool in 2005 to create a university digital repository. They carried out several trial projects to test the system capabilities of DigiTool, exchanged information with Ex Libris, and found the product suitable for their purpose. They noted that the advantages of choosing DigiTool over open source software were guaranteed availability of customer support from Ex Libris and the purchase of one single platform for managing the full range of digital collections at LJMU including documents, images, audios, and videos. McGill University (Park *et al.*, 2006) tested DigiTool as the system platform for creating "an Open Archives Initiative (OAI) compliant electronic thesis model that will be linked to the university's institutional repository" in its "storage, cataloguing, and dissemination capability". However, this paper focused on describing workflows of their ETD initiative and did not report the test results with DigiTool. In addition, two articles (Lynch and Lippincott, 2005; Kennan and Kingsley, 2009) listed DigiTool as one of the less common IR systems in the United States and Australia, respectively.

This paper describes CSUL's experience of using DigiTool to develop an IR. The discussions are divided into the following sections: DigiTool overview, local customization, implementation of handles, metadata and object ingest, usage statistics generation, integration with CSU Discovery, and conclusions.

DigiTool Overview

DigiTool is a complex digital asset management system which allows institutions to create, manage, and preserve online-accessible digital collections. DigiTool's system architecture includes back-end databases, web services/components, and a client-server module that works on the Windows system.

The majority of DigiTool functions can be performed in DigiTool's six primary web-based modules: Resource Discovery, Management, Approver, Collection Management, Deposit, and Web Ingest.

- The Resource Discovery module is the public web interface where end users access digital collections and objects.
- The Management module consists of tasks for repository management, such as system maintenance and cleanup, editing configuration files, creating user deposit forms, adding/editing Dublin Core (DC) metadata fields, generating/publishing handles, and creating reports.
- The Approver module allows staff to review user deposits and approve or decline submitted materials.
- The Collection Management module allows staff to create and organize virtual collections for end user access.
- The Deposit module is intended for use by end users to submit their materials.
- The Web Ingest module is used by staff to upload digital objects and metadata into DigiTool.

In addition, DigiTool provides a Windows-based application called "Meditor" (Metadata Editor). Meditor is designed for performing back-end staff functions on individual PCs and includes all the above-mentioned functions with additional functionalities in metadata and object management.

Depending on whether user deposit is involved, workflows in DigiTool can be either:

- User deposit > staff approval > staff ingest > repository index > user access; or
- Staff ingest > repository index > user access

The most common scenario of user deposit for us is an ETD collection in which graduating students directly submit their works into the repository. Because the ETD submission and approval process is still in its beginning phase, we haven't yet involved DigiTool user deposit module in our workflow. Our current procedure is that the DRS staff members receive student submissions in electronic format via our ETD web site [6], create metadata, and directly ingest into the repository. However, we anticipate involving the DigiTool user deposit module in the near future.

"Digital entity" is a key term in the DigiTool language. A digital entity is stored in the DigiTool back-end repository and is comprised of several components: a persistent identifier called "PID", a control section with various attributes (label, note, entity type, usage type, etc.), metadata of various types (administrative, descriptive, technical, preservation), data stream(s), and relations with other digital entities ("Manifestations", "Includes", "Part of"). Digital Entities in DigiTool may be one of the following usage types: ARCHIVE, VIEW, THUMBNAIL, or INDEX. A digital object may consist of one or more digital entities. For example, a PDF document normally consists of the following three manifestations, each a digital entity itself: the PDF (VIEW), an image in JPG format (THUMBNAIL), and a full-text document in HTML format (INDEX).

DigiTool can generate both simple and complex objects. A simple object has no internal structures, such as a student's thesis in a single PDF document (example: http://hdl.handle.net/10217/16266). A complex object has internal structures, such as an issue of a journal that consists of a front cover, preliminary pages, articles, and a back cover (example: http://hdl.handle.net/10217/22042). There are multiple ways to create single and complex objects in DigiTool, which are described in the Metadata and Object Ingest section of this article.

Metadata is a crucial component of a digital repository in terms of facilitating resource discovery. DigiTool supports various established descriptive, technical, and preservation metadata standards, including Dublin Core (DC), Machine-Readable Catalog Records (MARC), Metadata Object Description Schema (MODS), Metadata Encoding and Transmission Standard (METS), NISO Technical Metadata for Digital Still Images (NISO), and PREMIS Preservation Metadata (PREMIS). In addition to a full set of basic and qualified DC fields, DigiTool allows the definition of local DC fields for specific collection needs. For example, we have added Degree Name, Degree Grantor, Department, Advisor, Committee Members, and others for the ETD collection in order to conform to the Networked Digital Library of Theses and Dissertations (NDLTD)'s ETD-MS [7]. DigiTool also supports adding access rights metadata governing digital objects' view, edit, and ownership rights. DigiTool stores all metadata in XML format, regardless of its original formats.

Ingest, meaning the uploading of digital files and metadata into the repository, is a crucial process executed by the DRS staff. DigiTool supports various ingest approaches, based on object and metadata types. Manifestations and technical metadata may be generated during the ingest process by DigiTool. Post ingest, digital objects and metadata are searchable via the Resource Discovery module the next day after system update. In the end, a digital object and its metadata are delivered to end users via a DigiTool web dispatcher called "object viewer".

DigiTool supports both a Google-like simple keyword search and advanced search options. Digital objects and metadata in DigiTool are discoverable by common search engines such as Google, Yahoo, and Bing. Collections in DigiTool are organized in a tree structure for browsing and can be created in one of the following three types: node (which contains sub-collections), itemized (to which individual items are linked manually), and logical (which is updated by the system automatically based on a predefined search). All of our collections are node or logical. We have implemented logical identifiers that are manually assigned by DRS staff to all DigiTool items in order to fully utilize DigiTool's logical collection feature. For example, an item with a logical identifier "ETDF20091000001FRWS" will automatically appear in a collection with a predefined search, "ETDF2009******FRWS" once it is indexed. Details of logical identifiers are discussed in the CSU Core Data Dictionary (version 1.1) [8].

Working with DigiTool requires close collaboration between librarians and IT staff. A system administrator is required to maintain all system functions, such as configurations, updates, and creating/updating staff accounts. The DigiTool system is periodically updated with service pack releases.

DigiTool Customization

Various customizations had to be completed during our local implementation of DigiTool. Among them, two of the most important customizations are for the repository web interface and the metadata full view.

Repository Web Interface

Customizing the DigiTool web interface is simple, but we intentionally minimized customization to avoid repeated work. This is because when new DigiTool service packs are installed, some customized settings may be overwritten. Most institutions implementing DigiTool take a similar approach, which result in similar repository interfaces. Figure 1 shows a comparison of the repository home pages of the CSU Digital Repository, Center for Jewish History Digital Collections, Boston College eScholarship, and Publication of Archival Library and Museum Materials by State University Libraries of Florida.

TAKE IN FIGURE 1

From this example, we can see that all sites have customized color schemes, menus, search buttons, headers, and footers; however, the main sections that provide search and collection browsing functionalities are similar.

Web interface customization in DigiTool is mainly controlled by a CSS file. When we made the customization, we followed rules such as no changes of file names, no changes for template structure, no removal of placeholders across sections, and no changes of section names. The following two examples are CSU Digital Repository's custom styles for hyperlinks and page titles.

Example 1: Define Hyperlinks

```
a.TB:link, a.TB:visited, a.TB:hover {
color:#13694E;
font-family:'Arial Unicode MS',TAHOMA,ARIAL,VERDANA,sans-serif;
font-size:80%;
height:18px;
padding-bottom:2px;
padding-left:6px;
padding-right:6px;   }
```

Example 2: Define Font style for Page Titles

```
.PageTitle {
border-bottom:1px solid #66AA55;
```

```
font-family:'Arial Unicode MS',TAHOMA,ARIAL,VERDANA,sans-serif;
.PageTitle #browsetab {
 background-color:#66AA55;
 color:white;
 font-size:80%;
 font-weight:bold;
 padding:0 15px;  }
```

Repository Metadata Full View

DigiTool has four approaches to display descriptive metadata: brief view, table view, full view, and object viewer. Metadata displayed in the object viewer is a complete list of descriptive metadata associated with an object and is displayed in its original format (DC, MARC, or MODS). The metadata fields chosen to display in the other three views can be locally configured, especially the metadata full view. The metadata full view is an intermediary display of fuller metadata that facilitates user assessment of the content of a resource prior to actual access. Links to our repository items retrieved by common search engines direct to the metadata full view in DigiTool.

Configuration of the DigiTool metadata full view applies to all items across collections. Thus, we took careful considerations when deciding what to display in our metadata full view and the process involved discussions with the librarians from the College Liaison Services and Metadata and Preservation Services (MPS). As a result, only metadata fields that we consider crucial to user assessment of resources are included (see http://hdl.handle.net/10217/28672 for an example of CSU ETD metadata full view). In addition, the DigiTool metadata full view can only appear in the DC format; therefore, other metadata formats must be mapped to DC in order to display correctly in the full view. DigiTool provides default mappings of other metadata formats to DC in its backend databases, although we also referred to the Library of Congress MARC to Dublin Core Crosswalk [9] and made some additional modifications in our local MACR to DC mapping for accuracy and local needs (see http://hdl.handle.net/10217/4200 for an example of MARC metadata full view).

Implementation of Handles

"The Handle System is a technology specification for assigning, managing, and resolving persistent identifiers (PI) for digital objects and other resources on the Internet." [10] In DigiTool, a handle is referred to as PI or handle PI but we normally call it "handle". In our repository, a handle is presented as a URL that links to a digital object of VIEW type. A handle is designated to only one digital object and it is permanent. This means one object will have the same URL regardless of any change of hosting server or operating system. A handle, for example, http://hdl.handle.net/10217/1553, consists of three parts, a prefix (i.e. hdl.handle.net), an institution ID (e.g. 10217), and an object ID (e.g. 1153). Prior to DigiTool implementation, we requested an institutional ID from HANDLE.NET and installed the handle system outside the DigiTool framework.

Implementing handles is more complicated than web interface customization. We were one of the first institutions to implement handles in DigiTool, and during the process we worked closely with Ex Libris technical support staff. In this article, we list the main steps we have taken. For security purposes, we replaced actual paths and file names in the examples with ***.

General Configurations

(1) pi_profiles_rules.xml

The first file we modified is pi_profiles_rules.xml which specifies rules of generating handles. This file has two main sections, profile settings and rule settings. The profile settings section describes how handles should be built and the rule settings section determines which profile to use. For most DigiTool users, they only need to change pi_prefix from the default to their own handle prefix. For example, we changed ours from the DigiTool demo prefix “12345” to “10217” on line 3.

```
Line 1: <pi_profile name="handle_profile">
Line 2: <pi_type>handle</pi_type>
Line 3: <pi_preffix  required="true">10217</pi_preffix> //omitted code
```

The rule section defines parameters for generating handles. By default, the <rl:de_rule> section specifies that only digital entities of the VIEW or VIEW (PRIMARY) types can have handles and their associated manifestations, such as thumbnails and full text, have no handles. At CSU, we kept the default rules.

(2) pi_publisher_rules.xml

Next we modified pi_publisher_rules.xml which also has both profile and rule sections. For the rule section, we only changed the default <pi_prefix> to CSU’s prefix (as shown in line 4). For the profile section, we changed the <parameter name="authentication.public.handleName"> to CSU’s prefix and administrator password (as shown in line 7 and 9).

```
<!-- CSU  Digital Repository’s rules section example -->
Line 1: <rl:de_rule name="view_handle_publisher">
Line 2: <usage_type>VIEW</usage_type>
Line 3: <entity_type/><file_extension/><relation_type></relation_type><preservation_level/>
Line 4: <pi_prefix>10217</pi_prefix>  //omitted code

<!-- CSU Digital Repository’s profile section -->
Line 5: <pi_publisher><!-- No changes in this section -->
Line 6: <params>
Line 7: <parameter name="authentication.public.handleName">0.NA/10217</parameter>
Line 8: <parameter name="authentication.public.privateKeyFile">/***/admin***.bin</parameter>
Line 9: <parameter name="authentication.public.passphrase">xxxxx</parameter> //omitted code
```

(3) hdltool.ini

The hdltool.ini provides default authorization information. There are two configurations for a public and a private key authorization. At CSU, we use a private key.

Line 1: {"SecIndex" = "300"
Line 2: "PrivKey" = "/***/adm***.bin"
Line 3: "PubHandle" = "0.NA/10217" }

(4) handleConfig.properties

The handleConfig.properties is DigiTool's key for locating and communicating with the appropriate handle server. We replaced the default handle prefix to CSU's handle prefix and reset the default authorization values.

Line 1: authentication.public.handleName=0.NA/10217
Line 2: authentication.public.privateKeyFile=/***/adm***.bin
Line 3: authentication.secret.handleName=10217/ADMIN
Line 4: authentication.secret.pass=***

Local Configurations

For our repository, we decided to display a handle in DigiTool's metadata full view along with other DC fields and label it "Bookmarkable URL". The www_r_silo_conf.xml file configures metadata display in the brief view, table view, and full view. It specifies which field to display and its label. We modified one line in <results_full> section, <name lng="ENG">Bookmarkable URL</name>.
By default, the Bookmarkable URL and other URLs in metadata are strings of text. In order to make them hyperlinks, we added two functions in script.js. One function specifically converts handles into hyperlinks, and the other converts text strings starting with http or https in other metadata fields into hyperlinks. Other modified files include results-full-body and digital_entity_urls_template.xml. The source code can be requested from CSUL.

Implementation

A handle is not automatically generated post-ingest. There are three processes to make handles available in DigiTool: generate handles, publish handles, and repository harvest. There are two options to generate handles, global handle generation via the management module and create URN by PID via Meditor. At CSU, we use global handle generation via the management module because handles can be generated for a specified range of PIDs. After handle generation, we run a process to publish handles in the Management module. This process links a published handle on the handle server, which is outside of the DigiTool framework, to the corresponding digital entity on the DigiTool server. Last, the DigiTool repository must be reharvested before handles are available in the Resource Discovery. At CSU, we have set up an automated process in which the repository is harvested daily during low-traffic times.

Metadata and Object Ingest

DigiTool supports multiple approaches for ingesting metadata and digital objects, based on metadata (DC, MARC, MODS, or METS) and object types (simple or complex object). We currently have two types of descriptive metadata in the repository, DC and MARC. DC is flexible, easy to create, and works for all common types of electronic resources. We use DC when new metadata needs to be created. MARC is a standard metadata format for print library resources. In order to reduce metadata creation cost, we repurpose MARC when the electronic resources have corresponding prints that have been previously cataloged locally. Staff involved in metadata creation includes the members from DRS, MPS, and Archives and Special Collections (Archives), while ingest is exclusively executed by the DRS staff. The major file formats in our repository at present are documents in PDF and images in JPEG2000, although we do have seventeen videos and audios in various formats. Based on our experience to date, we have identified the following common types of metadata creation and ingest approaches to be discussed in detail.

DC XML and Associated File Streams

DC XML is Dublin Core metadata created in Extensible Markup Language (XML) [11] format. DC XML is easy to create and can be used to ingest one or multiple files at a time. DC XML templates can be created for materials of a similar nature (for example, ETDs, faculty publications, conference proceedings, journals, or student posters), which facilitate metadata creation and staff training. We create new metadata in DC XML when we have relatively small to medium-scale collections. In addition, DC XML can be used in conjunction with METS XML as descriptive metadata (see METS XML section). XML files are essentially TXT files that can be created and edited using any text editor; however, we recommend using a professional XML editor. This is because professional XML editors provide much friendlier working environment and offer features such as XML validation and transformations among different metadata schemas. At CSUL, we have chosen to use <oXygen/> XML Editor [12]. The academic version of <oXygen/> can be acquired at a very low cost and it offers an excellent set of functionalities.

MARC XML and Associated File Streams

We repurpose MARC when we have corresponding catalog records for digitized materials. Examples include Atmospheric Science Papers (see http://hdl.handle.net/10217/27), Colorado Agriculture Bibliography publications (see http://hdl.handle.net/10217/4458), and Colorado Water Institute publications (see http://hdl.handle.net/10217/3167). When repurposing MARC, we follow these procedures:

1. Catalogers in MPS extract MARC records from our cataloging system, the Millennium ILS powered by Innovative Interfaces.
2. Catalogers use MarcEdit [13], open-source software, to edit the exported MARC data. This includes adding filenames and format (in MARC 856 field subfield u and q, e.g. "856 00 $u 0211_Bluebook.pdf $q application/pdf"), logical identifiers (in MARC 592 field subfield a, e.g. "592 $a FACFATMS100042BLUE") and copyright information (in MARC 506 or 540 field subfield a, e.g. "506 $a http://www.acns.colostate.edu/?page=copyright"). Adding filenames in the MARC 856 field is required for linking individual MARC records to their respective file streams for this type of ingest.
3. Catalogers use MarcEdit to transform the edited MARC data into MARC XML.
4. Catalogers notify DRS staff available MARC XML files.

5. DRS staff members validate MARCXML files in <oXygen/>XML editor and ingest them into DigiTool with associated file streams.

Similar to DC XML, MARC XML can be used to ingest either one or multiple files at a time and be used in conjunction with METS XML as descriptive metadata (see METS XML section). Repurposing MARC reduces our cost of metadata creation. It is also an efficient way to make digitized materials available via the repository as quickly as possible.

CSV File

In large-scale digital projects when many staff members are involved in new metadata creation and in migration projects when we move metadata and digital objects from CONTENTdm to DigiTool, we use CSV files for ingest. Examples of large-scale digital projects include the University Historic Photograph Collection (see http://hdl.handle.net/10217/35514) and the Water Resources Archive (see http://hdl.handle.net/10217/31642). Examples of migration projects include the Celebrate Undergraduate Research and Creativity (CURC) Showcase (see http://hdl.handle.net/10217/24128) and the Information Science and Technology Center (ISTeC) Student Research Posters (see http://hdl.handle.net/10217/530).

A comma-separated value (CSV) file is a common and simple text file that stores tabular data. Each line in a CSV file corresponds to a row in a table and each column corresponds to a field that is separated by commas. "CSV files are often used for moving tabular data between two different computer programs, for example, between a database program and a spreadsheet program" [14]. CSV files storing metadata, used in conjunction with their mapping XML according to DigiTool specifications (see Appendix A), are an option of batch ingest that DigiTool provides. Procedures of CSV ingest for large-scale projects and migration projects are slightly different.
In large-scale digital projects, we follow these procedures:

1. MPS staff members create metadata in MS Excel. The first row in the Excel file contains names of metadata fields outlined by a metadata librarian, which varies by project. The following rows contain metadata records. The reasons we create metadata in MS Excel are that Excel provides many efficient text editing features and most of our metadata staff members are familiar with the application.
2. DRS staff members create mapping XML files based on the order of the metadata fields outlined in the Excel files. The purpose of mapping XML files is to interpret the tabular data stored in CSV for DigiTool processing.
3. MPS or DRS staff members convert the Excel files into CSV files using the "Save As > CSV (Comma delimited) (*.csv)" function in Excel.
4. DRS staff members ingest the CSV files, mapping XML files, and their associated file streams into DigiTool.

In migration projects, we follow these procedures:

1. MPS staff members export metadata from CONTENTdm in tab-delimited TXT files.
2. DRS staff members create mapping XML files based on the order of the metadata fields in the TXT files.
3. DRS staff members convert the tab-delimited TXT files into CSV files using a local program.

4. DRS staff members ingest the CSV files, mapping XML files, and their associated file streams into DigiTool.

CSV ingest is a very efficient way to batch upload metadata and file streams into the repository. Using such an approach, we have made available online nearly 10,000 images in the University Historic Photograph Collection and nearly 600 images and documents in the Water Resources Archive within a year.

Complex METS Objects

The Metadata Encoding and Transmission Standard (METS) schema "is a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library, expressed using the XML schema language…" [15]. METS XML supports creation of complex digital objects with internal structures, such as books, journals, and archival aggregations. A few examples of our complex METS objects are the CSU Board of Governors meeting materials (see http://hdl.handle.net/10217/18499), Journal of Student Affairs (see http://hdl.handle.net/10217/22042), and Rocky Mountain Farmers Union historical photographs (see http://hdl.handle.net/10217/3691). At this time, we have 366 METS objects in our repository. Complex METS objects can be generated either by DigiTool internally using ingest types specified below or by creating METS XML files offline and ingesting them with associated file streams.

(1) Generate METS Objects Internally by DigiTool

DigiTool can create complex METS objects internally by ingest type "File stream(s) that will become part of one parent record" or "File stream(s) utilizing DigiTool file naming convention". Using the "parent record" approach can only generate METS objects with physical structural maps, while using the "file naming convention" approach can generate METS objects with either physical or logical structural maps. The difference of a physical and a logical structural map lies in the navigation structure that they provide. A physical structural map provides flat navigation and a logical structural map provides hierarchical navigation within the METS object viewer (see Figure 2).

TAKE IN FIGURE 2

Using these approaches, descriptive metadata will need to be added and labels in structural maps will need to be edited later in Meditor. These ingest approaches are not limited by file format; however, they require more time in structural map editing, especially when a METS object consists of a large number of files. We use these approaches when we need to create METS objects of mixed file formats, for example, when a METS object consists of both PDF files and JPEG2000 images.

(2) Create METS XML Files Offline and Ingest with Associated File Streams

One of our R&D staff members developed a utility based on the METS object generation procedures that we acquired from the Center for Jewish History, a fellow institution that uses DigiTool. The utility works specifically with DigiTool and helps the DRS staff automatically generate raw METS XML files offline when a METS object consists of either all PDF files or all JPEG2000 images. The utility is available upon request to readers of interest.

The procedures of creating METS objects using this utility are:

1. Collect all files that will be included in the METS object and group them into one folder.
2. Number the files so that all files are listed in a desired sequence for end user navigation. For example:

   101_Cover.pdf
   102_Article1.pdf
   103_Article2.pdf
   104_Article3.pdf

3. Ingest these files into DigiTool as file streams with no relationships. During this ingest, the files are uploaded to the DigiTool server and manifestations of the files (e.g. thumbnail, full-text) may be generated by DigiTool as needed.
4. Run the METS utility. Select corresponding options according to the file and metadata format. Based on local needs, technical metadata may be extracted. The utility generates a raw METS XML file from the previous ingest.
5. Edits are made to the raw METS XML file, including providing correct information in the METS header, adding descriptive metadata in either DC or MARC XML, and completing the structural map. Structural maps can be either physical or logical (See Appendix B).
6. Ingest the completed METS XML file and the associated file streams generated from the first ingest.

The advantage of creating METS XML files offline is the flexibility and efficiency it provides to structural map creation, especially with logical structural maps.

Integration of DigiTool with CSU Discovery

We have several additional custom interfaces for searching items in DigiTool utilizing the Ex Libris DigiTool API [16], such as the University Historic Photograph Collection web site (http://lib.colostate.edu/archives/uhpc/) and CSU Discovery (http://discovery.library.colostate.edu/). CSU Discovery is an integrated search tool that allows end users to search our libraries' online catalog, the digital repository, and CSU web sites at one location (see Figure 3).

TAKE IN FIGURE 3

This search tool is built on VuFind [17], an open-source library OPAC project using PHP as a web component and Solr [18] as the backend database. To incorporate DigiTool data into CSU Discovery search, metadata is harvested via the Ex Libris DigiTool API and the output is in XML format. The harvest is conducted five days a week and the results are transformed for insertion into the Solr database. With CSU Discovery, end users can search items in DigiTool directly via the library home page

Managing Statistical Data

Statistical Data at the Item Level

In the DigiTool management module, there are two report functions. The Collection Distribution Summary report lists the top-level collections available in the repository and their distributions by the number of files in collections and the size of collections in MB. The Delivery Usage Statistics report shows the number of user requests and delivered requests on items for each top-level collection. For example, the report tells us that from June 10, 2009 to Nov 13, 2009, users made 219 visits to our Theses and Dissertations collection and 71 documents in this collection have been viewed. The report can also list what titles have been viewed and the number of requests and delivered requests on each title.

Besides the DigiTool report functions, we developed our own statistics page at the item level for additional monitoring of repository usage, which is written in PHP. It displays all objects of VIEW type in our repository. Information on each item includes its PID, handle, number of delivered metadata full views, number of delivered object views, title, creator, subject, date, identifier, and so on. The display can be sorted and searched by these attributes (see Figure 4).

TAKE IN FIGURE 4

In Figure 4, the handle number, 556, corresponds to the actual handle, http://hdl.handle.net/10217/556. For staff reference, we also have a column, Open Archives Initiative (OAI), linking to an item's OAI record in XML format that is created for external harvesting. Our statistics only include hits from non-library-staff IP addresses in order to help us monitor usage exclusively by outside end users. We also use this page to keep track of collection usage. By searching the logical identifiers, for example, by searching "FACFATMS" in the Identifier field, we can gather all faculty publications from the Atmospheric Science department and obtain a summary of usage statistics on this collection.

To create this page, our system administrator created a Perl script file to collect data from an Apache server log file which excludes hits from staff and web crawlers based on IP addresses, and output this data into a CSV file. A PHP file was then created to read the CSV file and display the results on the web. The code for creating this page can be made available to the audience of this article upon request.

Statistical Data at Site Level

At CSUL, we use Google Analytics [19] to track and analyze our main components' traffic on the library web site. These main components include the online catalog, the Digital Collections page, and the CSU Digital Repository. An individual profile was created for each component in Google Analytics. Figure 5 is a screenshot of the Google Analytics administrator interface. This interface can show the total number of site visits, average visiting time, and traffic trends for each profile by day, week, month, or year.

TAKE IN FIGURE 5

When clicking a "View report" link in the administrator interface, we can view a site's traffic trends, site usage, visitors' overview, traffic sources overview, and other statistical information. Figure 6 shows that from May 26, 2009 to June 25, 2009, there were a total of 5,389 non-staff visits to the CSU Digital Repository and 75.80% of the visits were redirected from search engines.

TAKE IN FIGURE 6

When clicking on the "View report link" in the Traffic Sources Overview section, we can view a breakdown of repository usage by different sources. For example, we can further tell from Figure 7 that out of the 4,085 visits redirected from search engines from May 26, 2009 to June 25, 2009, 3,715 were from Google.

TAKE IN FIGURE 7

Conclusions

From March 2008 to the end of 2009, CSUL has been able to make over 15,000 digital objects available online with DigiTool. Working with DigiTool requires close collaboration between librarians and IT professionals because of the complexity of its system architecture. We are most satisfied with DigiTool in its completeness of IR functions, support of established metadata standards and file formats, multiple ingest approaches that offer flexibility and efficiency, and the capability to integrate with our Discovery Tool and external search engines and harvesters. We found it beneficial to set up a test server that has helped us to investigate DigiTool system functionalities and ensure the accuracy and soundness of each system update. DigiTool users form a small yet close community, and we often find our questions answered via the DigiTool listserv by helpful colleagues. In conclusion, we consider DigiTool a powerful, complex, and relatively mature out-of-box IR platform that fulfills our needs to establish and maintain an IR.

Web sites

[1] http://www.exlibrisgroup.com/category/DigiToolOverview [accessed March 12, 2010]

[2] http://www.handle.net/ [accessed March 12, 2010]

[3] http://DigiTool.library.colostate.edu/R [accessed March 12, 2010]

[4] http://www.contentdm.org/ [accessed March 12, 2010]

[5] http://digital.library.colostate.edu/ [accessed March 12, 2010]

[6] http://lib.colostate.edu/repository/etd/ [accessed March 18, 2010]

[7] http://www.ndltd.org/standards/metadata/etd-ms-v1.00-rev2.html [accessed March 12, 2010]

[8] http://hdl.handle.net/10217/3147 [accessed March 12, 2010]

[9] http://www.loc.gov/marc/marc2dc.html [accessed March 12, 2010]

[10] http://en.wikipedia.org/wiki/Handle_System [accessed March 12, 2010]
[11] http://www.w3.org/XML/ [accessed March 12, 2010]
[12] http://www.oxygenxml.com/ [accessed March 12, 2010]
[13] http://people.oregonstate.edu/~reeset/marcedit/html/index.php [accessed March 12, 2010]
[14] http://en.wikipedia.org/wiki/Comma-separated_values [accessed March 12, 2010]
[15] http://www.loc.gov/standards/mets/ [accessed March 12, 2010]
[16] http://en.wikipedia.org/wiki/Application_programming_interface [accessed March 12, 2010]
[17] http://vufind.org/ [accessed March 12, 2010]
[18] http://lucene.apache.org/solr/ [accessed March 12, 2010]
[19] http://www.google.com/analytics/ [accessed March 12, 2010]

References

Valerie, Stevenson and Hodges, Sue (2008), "Setting up a university digital repository: experience with DigiTool", *OCLC Systems & Services*, Vol. 24 No. 1, pp. 48-50.

Park, Eun G. et al. (2007), "Electronic thesis initiative: pilot project of McGill University, Montreal", *Program: Electronic Library & Information Systems*, Vol. 41 No. 1, pp. 81-91.

Lynch, Clifford A. and Lippincott, Joan K (2005), "Institutional repository deployment in the United States as of early 2005", *D-Lib Magazine*, Vol. 11 No. 9, http://webdoc.sub.gwdg.de/edoc/aw/d-lib/dlib/september05/lynch/09lynch.html [accessed March 08, 2010].

Kennan, Mary Anne and Kingsley, Danny A. (2009), "The state of the nation: a snapshot of Australian institutional repositories", *First Monday*, Vol. 14 No. 2, http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/2282/2092 [accessed March 08, 2010].

Appendices

Ingest Appendix A: Examples of Metadata and Mapping XML for CSV

(1) A simple CSV file storing metadata:

File Name,Title,Author,Subject
Image1.jp2,Ocean,"James, Thompson",Ocean
Image2.jp2,Flower,"James, Thompson",Flowers
Image3.jp2,Sunset,"James, Thompson",Sun -- Rising and setting

(2) Mapping XML used in conjunction with the CSV file storing metadata:

```
<?xml version="1.0" encoding="utf-8"?>
<tm:x_mapping xmlns:tm=http://com/exlibris/DigiTool/repository/transMap/xmlbeans
start_from_line="2">
        <x_map>
                <x_source position="2"/>
                <x_target md_name="descriptive" md_type="dc">dc:title</x_target></x_map>
        <x_map>
                <x_source position="3"/>
                <x_target md_name="descriptive" md_type="dc">dc:creator</x_target></x_map>
        <x_map>
                <x_source position="4"/>
                <x_target md_name="descriptive" md_type="dc">dc:subject</x_target></x_map>
        <x_map>
                <x_source position=""/>
                <x_target>control/usage_type</x_target>
                <x_default>VIEW</x_default></x_map>
        <x_map>
                <x_source position="1"/>
                <x_target>stream_ref/file_name</x_target></x_map>
        <x_map>
                <x_source position=""/>
                <x_target>stream_ref/directory_path</x_target>
                <x_default>/</x_default></x_map>
        <x_map>
                <x_target>stream_ref</x_target>
                <x_attr>store_command</x_attr>
                <x_default>copy</x_default></x_map>
</tm:x_mapping>
```

Appendix B: Examples of METS Physical and Logical Structural Maps

(1) A physical structural map for flat navigation

```
<structMap TYPE="PHYSICAL" ID="smd001">
   <div ORDER="1" ID="Auto_Generated_Qualifier_STRUCTMAP" TYPE="image folder"
LABEL="Colorado - Brighton - Agricultural Machinery" DMDID="dmd001"
xmlns:mets="http://www.loc.gov/METS/"  xmlns:mods="http://www.loc.gov/mods/"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns="http://www.loc.gov/METS/" >
     <div TYPE="image" LABEL="Photo 1" ORDER="1">
      <fptr FILEID="s001" /><fptr FILEID="t001" /></div>
     <div TYPE="image" LABEL="Photo 2" ORDER="2">
      <fptr FILEID="s002" /><fptr FILEID="t002" /></div>
     <div TYPE="image" LABEL="Photo 3" ORDER="3">
      <fptr FILEID="s003" /><fptr FILEID="t003" /></div></div></structMap>
```

(2) A logical structural map for hierarchical navigation

```
<structMap TYPE="Logical" ID="smd001">
  <div ORDER="1" ID="MANUAL_GENERATED_STRUCTMAP" TYPE="document" LABEL="CSU Board
of Governors 2004 meeting agendas and minutes" DMDID="dmd001"
   xmlns:mets="http://www.loc.gov/METS/"
   xmlns:mods="http://www.loc.gov/mods/"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://www.loc.gov/METS/" >
   <div ID="div1.1" LABEL="Chapter 1" TYPE="document" ORDER="1">
     <div LABEL="Section 1" TYPE="document" ORDER="1">
      <fptr FILEID="s001" /><fptr FILEID="t001" /><fptr FILEID="i001" /></div>
     <div LABEL="Section 2" TYPE="document" ORDER="2">
      <fptr FILEID="s002" /><fptr FILEID="t002" /><fptr FILEID="i002" /></div></div>
    <div ID="div1.2" LABEL="Chapter 2" ORDER="2">
     <div LABEL="Section 1" TYPE="document" ORDER="1">
      <fptr FILEID="s003" /><fptr FILEID="t003" /><fptr FILEID="i003" /></div>
     <div LABEL="Section 2" TYPE="document" ORDER="2">
      <fptr FILEID="s004" /><fptr FILEID="t004" /><fptr FILEID="i004" /></div></div>
   </div></structMap>
```