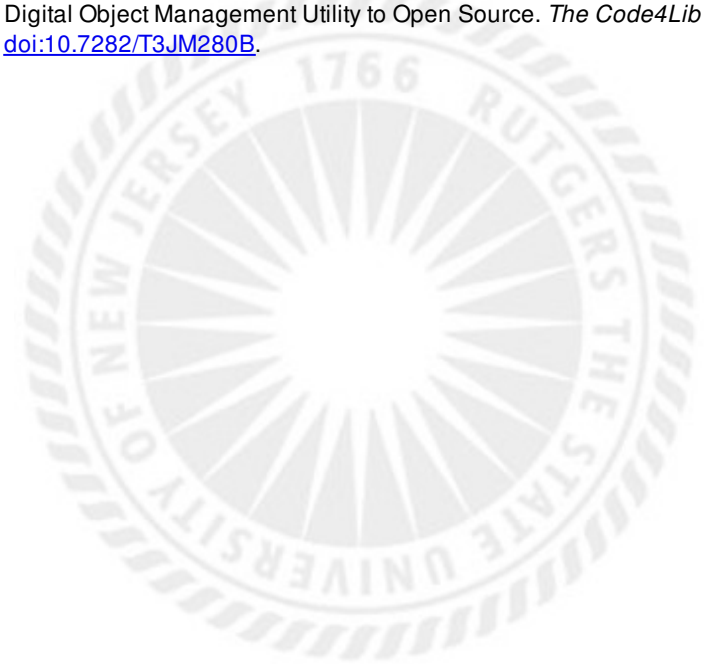
The Rutgers Workflow Management System: Migrating a Digital Object Management Utility to Open Source



The Rutgers Workflow Management System: Migrating a Digital Object Management Utility to Open Source

Authors: Grace Agnew, Associate University Librarian for Digital Library Systems and WMS metadata designer, Rutgers University Libraries

Yang Yu, Database Architect and WMS Architect, Rutgers University Libraries,

Abstract: This paper examines the development, architecture, and future plans for the Workflow Management System, a digital object management utility developed by Rutgers University Libraries (RUL) to create and catalog digital objects for repository ingest and access. The Workflow Management System (WMS) was created to solve two particular problems: a front-end utility for the Fedora open source repository platform and a vehicle for a flexible, extensible metadata architecture, to serve the information needs of a large university and its collaborators. RUL developed an application that meets its needs for digital information management and has described the capabilities of the application in papers and presentations, which generated considerable interest among other organizations interested in using the WMS, particularly as a vehicle to utilize the innovative metadata architecture. The Library of Congress contracted with Rutgers University Libraries (RUL) to develop the WMS as a bibliographic utility for its Moving Image Collections project. The next phase of development for the WMS shifted to a re-engineering of the WMS as an open source application. This paper discusses the design and architecture of the WMS, its' re-engineering for open source release, remaining issues to be addressed before application release, and future development plans for the WMS.

Introduction

One of the core services for a repository architecture is the web-based application that supports the ingest of digital objects into the repository, along with the creation and linking of metadata records that describe and manage those digital objects. Most repository services, from preservation and storage to discovery and retrieval, are dependent on the information collected about the digital object at ingest. The scalability of the repository, particularly to support the simultaneous ingest of many digital collections, is also dependent on this critical service. The Workflow Management System (WMS) was originally developed to meet RUL's need for a flexible, extensible web-

based service to support repository development. The WMS includes a sophisticated metadata architecture that was designed to support any digital collection, from any contributor, whether a faculty member depositing the research products of a large scientific experiment to a small museum or historical society participating in a collaborative cultural heritage portal. This article describes the development of an object ingest and metadata creation application that began as a front-end service for RUL's Fedora repository. As we presented our WMS to our peers at conferences [1], we were approached about sharing the application. We identified a need to significantly retool the WMS to remove Rutgers-specific dependencies and also to provide the level of customization that libraries, archives, and other organizations need to support their unique circumstances for information management and delivery. This paper describes the background and rationale for developing the WMS, its design and functionality, particularly to provide a sophisticated event-based data model and metadata architecture within a METS (Metadata Encoding & Transmission Standard) framework, and the re-engineering decisions that were required to create a robust open source application. The article closes with the policy and procedural issues that must be addressed before the WMS is released in the open source community, as well as next steps in WMS development.

Background

In 2002, the Rutgers University Libraries began exploring open source repository platforms to serve as the basis for a comprehensive cyberinfrastructure that would manage the preservation, access and use of the intellectual property of a large research university. The Fedora repository architecture [2] was selected for the sophistication of its service-oriented design and the simplicity of its approach to resource management. Within Fedora, anything can be an information object. No assumptions are made about the nature of the information to be managed or its intended use. Core services for preservation and management are provided for digital objects, but most services beyond basic preservation and access must be locally developed and layered upon the core architecture.

Our survey of Rutgers research, based on an analysis of grant-funded projects and other research products hosted on Rutgers websites, revealed an enormous breadth of digital content and a wide range of approaches for managing that content, from large scientific databases maintained on multiple Excel® spreadsheets to a complete portal with metadata, digital objects, and a suite of user services. We realized that we would need a flexible, extensible metadata architecture to

encompass the wealth of information available on campus and to support the metadata decisions that many faculty had already made to support discovery and access by colleagues in their disciplines.
In addition to developing an institutional repository to support faculty research and publications, Rutgers had received a grant from the Institute of Museum and Library Services to develop a statewide cultural heritage repository, the *New Jersey Digital Highway* [3]. Discussions with archivists, museum curators and librarians around the state identified a need to provide not just discovery and access but management of the state's cultural heritage resources. As one of the original thirteen U.S. colonies, New Jersey has a rich historic heritage, with many artifacts housed in small historical societies and museums around the state. The cultural heritage community was very interested in an information architecture that would support ongoing management of the analog source materials—photographs, papers, artifacts, etc., as well as the digital surrogates deposited in the New Jersey Digital Highway repository.

The Rutgers University Libraries needed an information architecture that would support several critical needs: (1) to enable the libraries to integrate and support the heterogeneous research products and publications of Rutgers University faculty; (2) to support management, preservation and access to historic and cultural source materials, particularly the New Jersey Digital Highway collection and RUL's special collections and (3) to integrate with the Fedora core architecture which supports both Dublin Core and the Fedora native XML schema, FOXML.

The Rutgers Information Architecture

Rutgers decided to use the Metadata Encoding & Transmission Standard, a metadata architecture supported as an international standard by the Library of Congress [3]. Fedora was initially designed as with a METS data architecture. Fedora migrated its data architecture to FOXML, which maps to METS and has been describes as "METS Lite," so the choice to use METS made sense from an architectural standpoint. In addition, METS provides all the categories of information needed to manage and provide access to a resource. METS concatenates different types of metadata with one or more versions of the object, as well as structural information and behaviors for relating METS components, navigating complex objects, such as the pages of a book and for displaying and using the digital object. The METS envelope provides a standardized XML wrapper for organizing, storing, managing and transporting all the METS

components as a single object. The METS document is a standardized transmission package that can be shared across METS-compliant repositories.

There are seven component metadata documents within a METS document:

- Header, which provides basic information about the creation of the METS object
- Descriptive metadata, supporting discovery and access to resources
- Administrative metadata, providing metadata about the creation, provenance and use of resources. Administrative metadata includes four subtypes: technical, source, rights and digital provenance
- File section, which groups together related files, such as the different digital manifestations of a resource—the TIFF digital master file, the JPEG access copy, etc.
- Structural map, which provides the hierarchical structure of a complex resource, and links the elements of the structure to relevant content files and metadata
- Structural Links, which enable hyperlinks between nodes in the structural map
- Behaviors are procedures or applications that can be executed upon content contained within the METS package. [4]

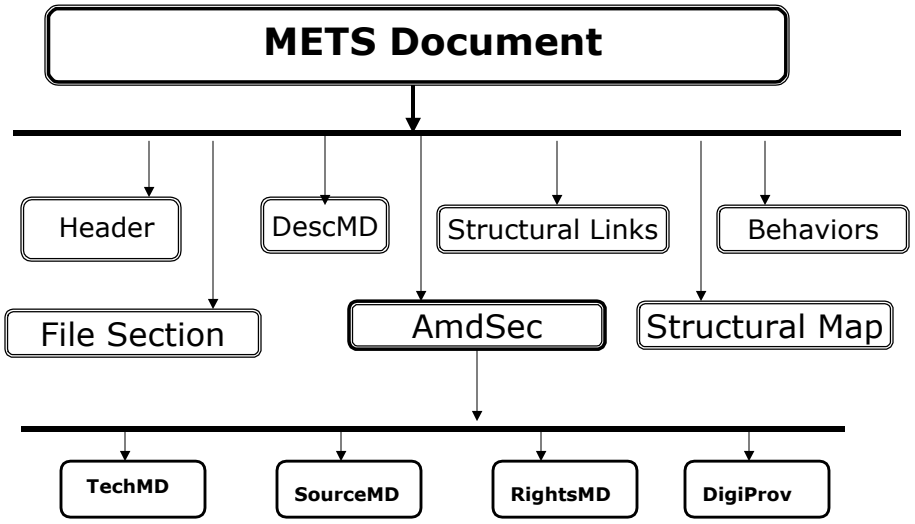


Figure 1. METS Information Package

METS accommodates all the physical manifestations of an information entity. In Rutgers' implementation of METS, the source object is the first generation of information under the control of the organization. For example, RUL may own a photograph of the Venus de Milo. The famous statue itself is not owned or managed by the library. Instead the source object, which is the first generation of information under the control of the organization, is the photograph of the Venus de Milo. Source information objects are generally information that can pass the "hurricane test," in other words it is the information you will save if your archive is in the path of a hurricane and you must evacuate the premises. RUL documents information about the provenance and condition of analog source materials in source metadata. RUL documents the characteristics of born digital source objects and digital master files in technical metadata. RUL places an equal focus on access to resources and long-term availability. This requires capturing information at ingest sufficient to manage objects and to safeguard and document intellectual property rights for rights holders. The Rutgers METS implementation includes a complete rights metadata implementation, with the ability to link rights documentation (deeds of gift, permission request letters, privacy releases, etc.) to the rights events that secure for RUL the ability to make copyright-protected resources available for Web dissemination.

One important way to insure the long-term usefulness of information is to provide durable context for resources, so that a user in the future knows what he is accessing, how it is created, and what rights he has to its use. Since copyright currently extends 70 years beyond the death of the creator, durable provenance is important both to insure authenticity of information and to insure its legal availability over time. We address provenance by supporting the ability to add lifecycle and use information via event metadata continuously throughout the life of a digital resource. The RUL data model is primarily an event-based data model, intended to document the lifecycle of each digital resource incrementally, over time.

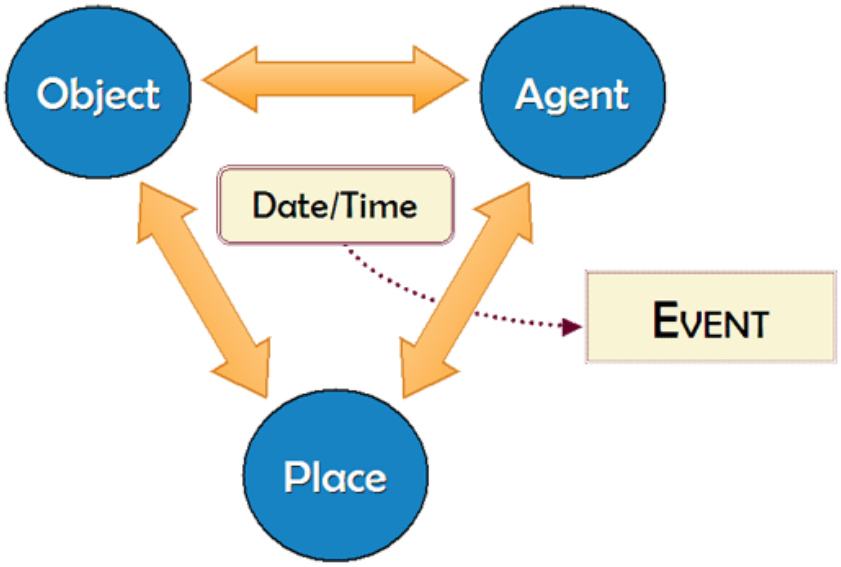


Figure 2: Rutgers University Libraries Data Model

RUL adds preservation and condition events, provenance events, rights events, and descriptive events, which document the cultural, pedagogical and research usefulness and impact of resources over time to the core metadata we create at ingest within the different METS documents. We chose the "event"—what happens to a resource at a specific time and place, within the context of METS metadata categories (descriptive, source rights, and digital provenance) as a standard conceptual data model that works with all information domains. An event can have associated entities, such as a granting agency, an awards agency, a preservation service provider, a rights holder, or an exhibit curator. An event can have associated objects, such as a deed of gift or website. A descriptive event will also provide us a standardized way to utilize social networking technologies, to capture critiques, recommendations, and use patterns by colleagues in a standardized way to provide nuanced access to resources. This is particularly critical for information products, such as experiment documentation and multimedia files, that may not have peer review status.

The event data model addressed our need to create rich "living" data objects that add the context necessary for provenance, discovery and use, and the METS document provides a framework for the contextual events we wanted to capture. Examples of provenance events include acquisition, donation, etc. Examples of preservation events include repair, reformatting, etc. Examples of rights events include license or permission, rights transfer, etc. Events provide meaningful context

and document the entire lifecycle of an object. Figure two provides an example of a descriptive event—the exhibition to which an object belongs. Events enable us to document associated entities and objects, such as the curator of an exhibit and the exhibit catalog.

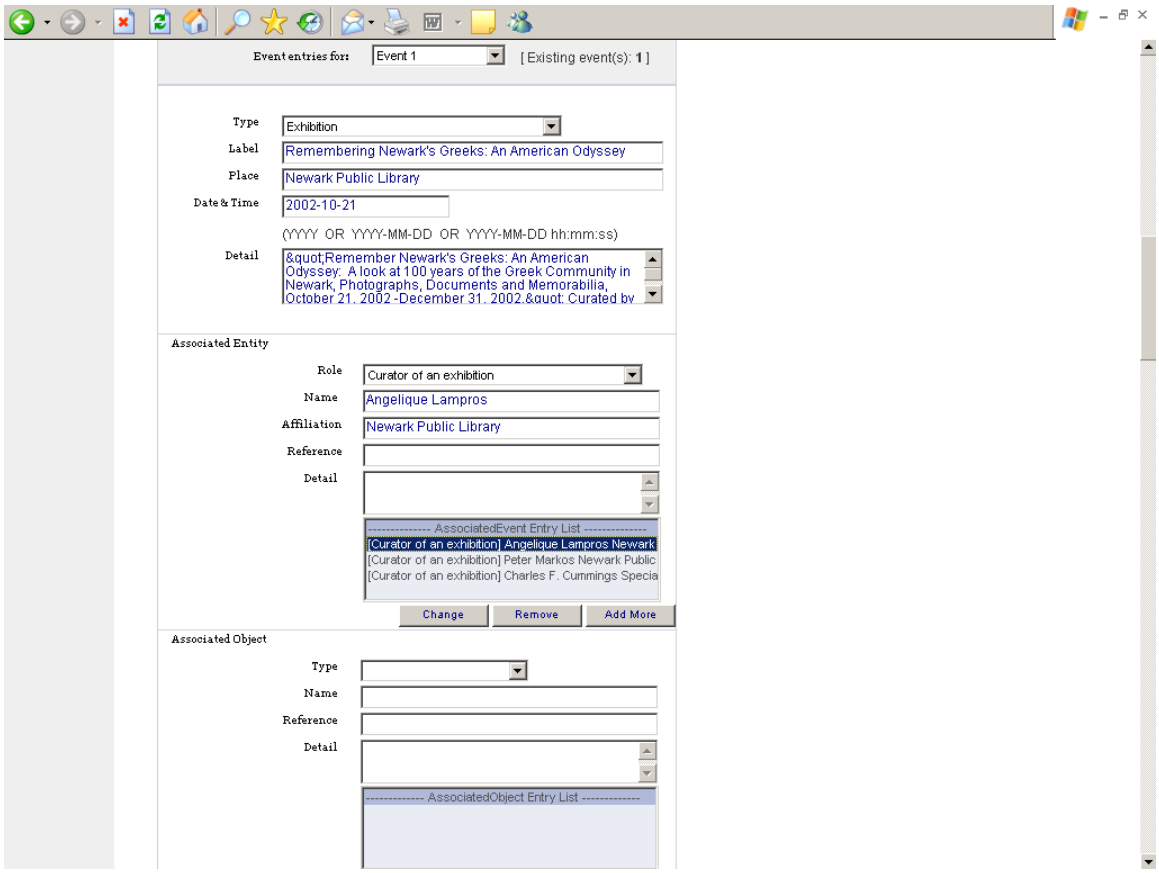


Figure 3: Example of a Descriptive Event in the WMS Exhibition to which the resource belongs

Another important event that can be captured in the WMS is the rights event. Whenever possible, we document the deed of gift or permission received from the rights holder that enables the repository to make a copyright-protected resource available for users.

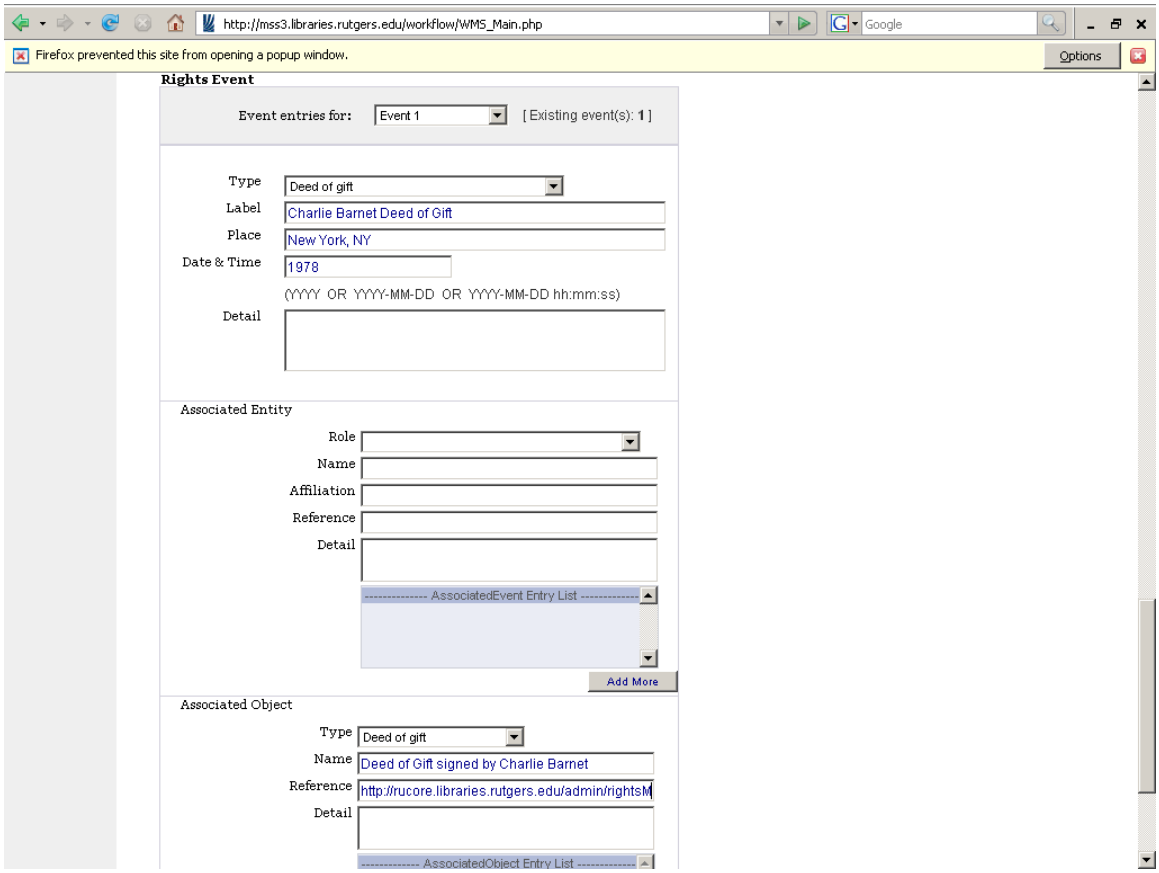


Figure 4: RUL rights event –deed of gift

The Workflow Management System

Once we had a conceptual data model, a metadata architecture and a data element registry, we needed to incorporate the complex metadata architecture into a web-based object ingest and metadata capture tool for the RUL fedora repository architecture. This tool was the Workflow Management System, which at that point consisted of a skeletal "placeholder" metadata implementation and a pipeline application for ingesting digital resources into the repository, creating access copies in multiple formats, and creating OCR files for textual materials or transcripts accompanying media files, which could be searched via full-text. The Workflow Management System needed to support a range of large and small libraries, museums and archives that would add resources and create metadata independently of the Rutgers University Libraries, as well as Rutgers faculty, who would deposit publications and research products in the RUcore, the Rutgers Community Repository [5]. For most *New Jersey Digital Highway* and RUcore participants, the Workflow Management System would be their

only exposure to the inner workings of the repository. The WMS needed to be robust, intuitive and yet support a very sophisticated data architecture that used concepts and terminology that were unfamiliar even to experienced catalogers, who worked primarily with MARC, Dublin Core and MODS (Metadata Object Description Schema) metadata standards.

The Workflow Management system that we designed over several years to support our sophisticated data architecture is now a core enabling technology for the RUL cyberinfrastructure. For two years, the WMS was largely tested through its use for NJDH by participating museums, libraries and historical societies around the state of New Jersey. Three important issues emerged that had a great impact on the continuing development of the WMS. To begin with, most participants had already created some level of metadata for their collections, even if just a simple spreadsheet or word processing file, and they were understandably reluctant to re-create this information, even through cutting and pasting. They were willing to iteratively add to their metadata once it was ingested, particularly if they felt the incremental event metadata added value, but not to trample old territory by recreating existing information from scratch. A flexible mapping utility that could accept and map data elements from any metadata schema, from the complex to the rudimentary, became a critical component for enabling museums and libraries to participate in the New Jersey Digital Highway. The mapping utility received a further test recently when RUL assisted the Virginia Tech Library in developing a commemorative repository for the April 16 shooting tragedy. RUL was able to successfully map the spreadsheets that inventoried the thousands of banners, cards, and other tributes that the university received in the aftermath of the tragedy into useful metadata to enable Virginia Tech to quickly create a permanent digital archive.

A second issue was the need to ingest large amounts of source objects in bulk rather than uploading each digital object one at a time. The WMS initially required that each object be individually loaded. This proved to be very time-consuming, particularly for large digital files. Requiring that users physically load each digital object is very inefficient and definitely not scalable. An important development was thus a mass-ingest capability that supported unattended bulk loading of objects.

A final requirement was to support some level of customization for participants. We were not able to anticipate the vocabulary needs of

all participants for populating metadata elements. We also needed to support local decision-making about data elements to display to their catalogers. We added a template capability to enable participants to select mandatory and recommended data elements for display and to add default values for data elements. Default values for technical metadata are particularly useful, since resources are either digitized to standard technical specifications or created as born digital objects in a standard digital format. The template capability allows users to customize the look and feel of the metadata input to suit their needs. Given the complexity of the RU metadata implementation, this was a critical feature. The template enables organizations to use the complex data architecture iteratively—adding from the large array of available data elements over time, as their expertise grows or their information management needs change.

The Workflow Management System was initially developed beginning in 2003 to provide a robust and intuitive user interface for the Fedora repository system. Its' initial design reflected its dual purpose for supporting the Fedora architecture and the needs of a diverse group of libraries, museums and archives with cultural heritage resources. As the WMS developers began to speak about the WMS in a wider environment, interest among other organizations in using the WMS intensified. In 2006, the Library of Congress Motion Picture, Broadcasting and Recorded Sound Division contracted with RUL to develop a bibliographic utility to support the moving image archives community, as part of its MIC (Moving Image Collections) project. [6] In the course of WMS development, RUL began thinking about adding modularity and additional customization features to the WMS to make it a usable application that could be used as a stand-alone application or integrated with other repository architectures by a wide range of organizations.

The features and functionalities associated with WMS have been constantly growing, and now it is moving towards becoming a generic integrated digital object workflow management system to be released to public as open source package in early 2008. The obligations imposed by open source required that the design of WMS must be flexible to meet different needs of diversified users; the functional components must be highly modular for software developers to easily add or remove; and the software must be readily maintainable. At Rutgers, the WMS currently serves as a front end for several user applications, the New Jersey Digital Highway, the Rutgers Community Repository (RUcore), the faculty deposits module, the open journal platform and the electronic theses and dissertations (ETD) module.

The WMS is a cornerstone application for the RUcore cyberinfrastructure

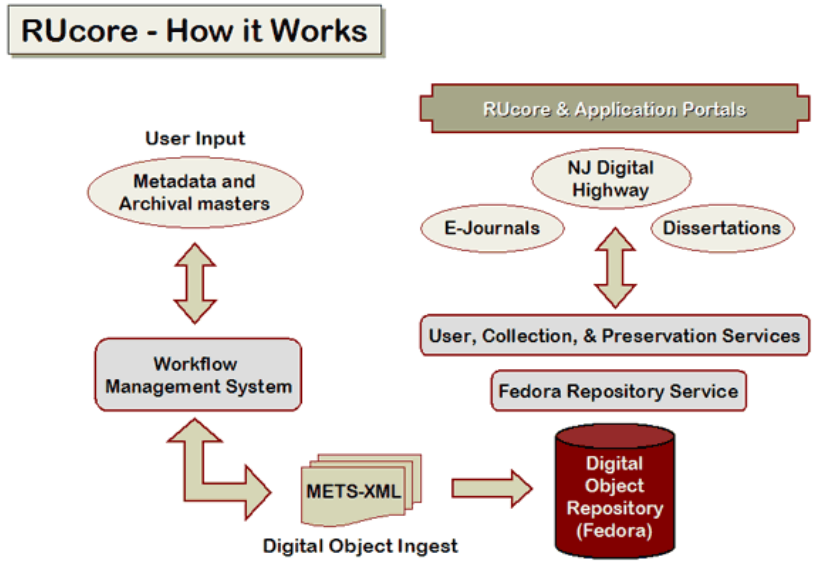


Figure 5: Role of the WMS in the RUcore cyberinfrastructure

Capabilities of the Workflow Management System

The Workflow Management System provides a complete object ingest and metadata creation system, with services to ingest objects and metadata and to export these objects and metadata, individually and in bulk. The WMS consists of object handling, which includes object ingest, object structuring for any format (image, text, multimedia), and object reformatting that sing the METS structure map, and the creation of multiple object formats. The pipestream application that creates multiple formats from the ingested digital master file is very RUL dependent, since we currently use a PDF server application as the "middleware" for reformatting multiple access formats for text and images. This capability has been modularized and abstracted from the WMS so that organizations can integrate their own application, which may be JPEG2000-based, for example. Object handling includes the ability to ingest transcripts and to provide OCR for text and transcript files. The WMS provides a full METS metadata architecture that supports access, discovery and management of an information resource in all its manifestations, from analog or born digital source objects to digital technical masters and access copies. Local customization includes authentication and authorization for managers and metadata creators; the ability to customize and add vocabularies to data elements, and the template capability to customize the look

and feel of the metadata input and to add default values to data elements. Objects are organized within collections and sub-collections, which is used both to provide intrinsic relationships among objects, such as articles within the issue of a journal and to enable objects to "inherit" collection level information via templates, such as a rights event for the collection level deed of gift. This insures that any collection level context is always readily available at the individual object level.

The Architecture of the Workflow Management System

Architecturally, the WMS is a container that includes many functional modules. All WMS modules follow the three-tiered design pattern that separates the user interface and persistent data storage from the business logic.

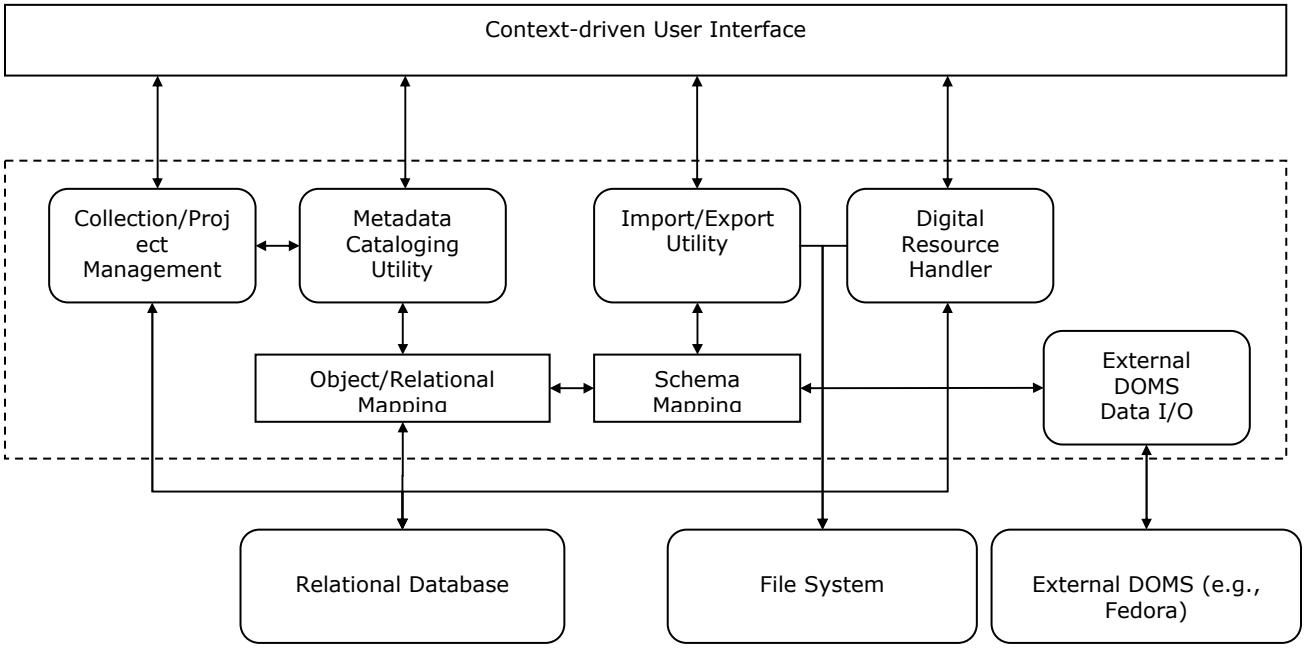


Figure 6: Architecture of the WMS

As figure 6 demonstrates, the WMS can be conceptually divided into three layers – the context-driven user interface layer, workflow policy/business logic handling layer, and persistent data storage (internal or external) layer.

WMS Database Design

The internal persistent data storage includes a relational database and file system. This layer does not directly deal with workflow logic, but its design, especially the schema design for the relational database, affects the overall ability for the system to function, the data integrity, and system performance.

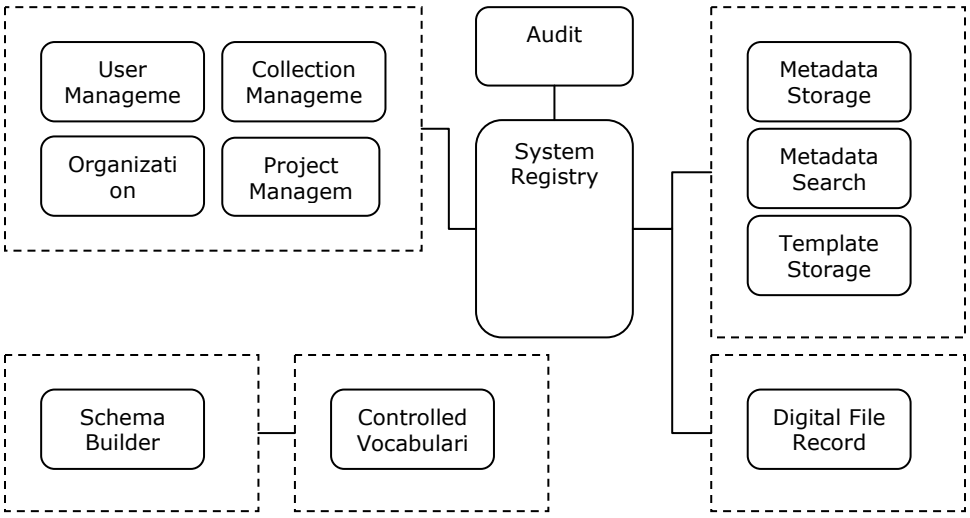


Figure 7 WMS Database Design

The WMS relational database schema is built upon the functionality class of the data. As figure 7demonstrates, there are five major functional blocks in the schema: WMS collection management, digital file handling, metadata and template storage, metadata schema builder, as well as controlled vocabulary management.

In addition to the standard relational database design considerations, two important decisions were made for the WMS database structure design. One is that digital object metadata is treated as an xml document as a whole instead of as a hierarchical data set, with a separate searching table for selected indexed terms. This approach ensures that the WMS database structure does not depend on any specific metadata schema; increases the overall system performance; and makes it much easier to port existing data to xml enabled database systems to utilize their xml searching capabilities. The second decision concerned the schema builder. The schema builder block stores metadata input form layout information as well as the associated xml structure for each metadata element. This method

significantly reduces the amount of work needed for web form maintenance, and provides, to some degree, the project manager with the capability to build and edit the metadata schema from a web input form.

Workflow Modules

The workflow logic layer is where all the workflow policy and business logic handling take place. This layer can contain as many functional plug-in modules as needed. Currently, the following functional modules have been developed:

- Authentication/authorization
- Collection/project/user management
- Digital resource handling
- Metadata cataloging utility
- Metadata schema mapping utility
- Digital object batch import/export utility
- External digital object access module

Each module operates independently from one another, but they share following common design patterns:

- Object-oriented software design. Though the WMS is written in PHP, the design of the software uses an object-oriented approach. Object/relational mapping is used as the solution for data access.
- Database driven application. A relational database structure forms the foundation for all WMS modules. Many of the modules, especially the metadata cataloging utility, heavily depend on information retrieved from the database for data processing and form rendering.
- XML as both data model and data carrier. The WMS uses XML to model internal data as well as exchange data with external resources.
- PHP session handling. The session matrix is used for all modules for keeping track of state information.

The metadata cataloging utility is one of the core modules and is the most heavily used module in the WMS. Currently the metadata entry forms use a METS compliant element structure. The descriptive metadata is built upon MODS and the technical metadata utilizes data elements from diverse schemas such as PREMIS [7], MIX [8], and the forthcoming AES-X098 metadata standard for audio resources. [9] The

WMS can be easily changed to fit other metadata schemas through the use of the schema builder configuration utility. The WMS was intentionally designed to be schema independent and thus to support all metadata schema. This is important because metadata is still an emerging area and schemas to support different communities and types of resources continue to proliferate. Metadata entered are stored using the internal XML data model, and conversions between different metadata schemas are done through a custom-built XSLT script. This design pattern makes the cataloging utility flexible and efficient for both utility users and software developers. The mapping architecture for both data import and data export is key to supporting interoperability with other communities, through data sharing using the OAI-PMH (Open Archives Initiative-Protocol for Metadata Harvesting) [10] or federated searching.

Digital resource handling deals mainly with digital file formats and conversion between them. WMS was initially tied to the RUL digital file handling policies, which requires TIFF as the archival master for images, and DJVu, PDF, and jpeg for presentation formats, for example. DJVu, which is a very functional but proprietary and not widely used digital file format, served as the intermediary format for the conversion of digital master files to multiple presentation formats, particularly JPEG and PDF. RUL is in transition for image presentation formats as we integrate Adobe PDF server into file handling, particularly to support faculty deposits and ETDs, and as we evaluate migrating from TIFF to JPEG2000 as the RUL master format for digital images. The goal now is to abstract this file transcoding capability and enable the individual institution to implement file handling capabilities specific to its needs.

The metadata schema mapping/import utility allows batch import of metadata and digital resources into the WMS system from an existing external database or files. Conversion between schemas is achieved with XSLT scripts custom developed for each specific schema. Currently this utility supports mapping and conversion from plain text bibliographic records to MODS, Dublin Core, and the RUL metadata schema used in WMS. PBcore and MPEG-7 are currently in development, particularly to support the MIC bibliographic utility. There are no limits to the number of mappings that can be supported, other than the limits on the cataloger time and effort needed to develop the mappings. A metadata mapping facility developed for the MIC project enables any participant in the MIC union catalog to input the data elements from the institution’s unique schema, map data elements to MIC data elements, test the mapping through a sample

record load and then batch load their metadata. The mapping is stored for the organization for future use. A MARC mapping exists as a standard map. This mapping functionality was critical for MIC because many moving image archives utilize their own custom metadata schema. This mapping utility will migrate to a future version of WMS to enable the WMS owner to map metadata that doesn't conform to a schema to the WMS. This is particularly critical for supporting university faculty, who often create their own metadata implementations to provide access to their research. The export module converts the metadata and digital resources in the WMS into a user specified format, then stores as files or exports to an external digital object management system, such as Fedora. Currently the WMS exports in METS and Fedora Object XML (FOXML) format. MARC is under development.

For the software developer, the key to maintaining the flexibility of the WMS is to follow the WMS modular design pattern. Using the common module libraries provided with the WMS core release, additional modules could be easily plugged into the system (for example, the Fedora object editing module developed for the Rutgers University RUcore repository, which provides for metadata or object editing after the object and metadata have been ingested into Fedora).

User Interface

The aim for the WMS user interface design is to provide a simple and logical workflow pattern that a cataloger or someone uploading digital objects can easily follow. At the same time, the interface code needs to be easily maintained and modified, particularly since requests for changes in the WMS have mostly involved the user interface. Several measures have been taken to fulfill this goal:

- WMS interface is dynamic and context driven. A user can configure the application for what he/she needs. Depending on the user's choice during the working process, the WMS displays just sufficient information for him/her to accomplish the task.
- Template for metadata cataloging. The template can be easily created, edited, enabled, or disabled, at the collection, project, or personal level.
- Context driven help is provided for metadata cataloging.
- WMS forms are designed to be clean and simple. Distracting elements, such as over decorating, audio, flash video, etc., are avoided.

- WMS user interface html code is dynamically generated using a handful of code templates, based on the business logic provided to the utility.

WMS System Requirements

The WMS is written in PHP, and it depends on relational DBMS to function. The WMS runs on PHP 4 and above, and theoretically any RDBMS, commercial or open source, can be used with WMS. However, considering the needs of open source software community, we focused on testing the open source RDBMS only. Currently it has been tested to run without problems on the two most popular open source RDBMS, MySQL (4.0 or above) and PostgreSQL (6.0 or above). WMS can be run on UNIX, Linux, or Microsoft Windows system with any web server, as long as it is configured to support PHP.

Re-engineering the WMS to Support Open Source

Open source or free software? The two terms are sometimes used interchangeably, but the difference between the two concepts is significant: for both open source and free software, you can download, install it on your system, and run it without a fee. However, for open source software, you can also read the source code and modify it to suit your needs. You can also build upon the code to create new applications, as long as you abide by any licensing or copyright restrictions.

From the software development point of view, providing open source software to the public implies fundamental changes in design decisions compared to creating proprietary institutional utilities, simply because the user base and their needs become much larger and unpredictable. We can no longer expect detailed requirement specifications and quick, convenient interactions with the potential users. Things that seemed so natural to one institution or organization, such as institutional policies that usually are embedded in the proprietary software, become problems to others. The commercial software that would have naturally been used in a proprietary utility for one institution may not be available to others. The operating systems and supporting software for the utility that worked fine with one institution could be unavailable or out of date for others. Open source means developers all over the world can modify or contribute to the code. This will exponentially increase the code maintenance and documentation issues and tasks. Therefore, moving from a proprietary institutional utility to open source software doesn’t mean simply making the utility available for

anyone to download. It is a whole new concept and usually means the software needs total redesign and development.

WMS was a typical proprietary institutional utility when the project was started. In order to convert the utility to open source software, WMS has undergone a total redesign since the beginning of 2006. The goal for the redesign was to address the issues mentioned above for the open source software as much as possible. The design philosophy is that the WMS needs to be flexible enough so that it does not depend on any specific external digital object management system (Fedora, DSpace, etc.); does not depend on the policies of any specific institution; and does not depend on any commercial software product; The design philosophy must also ensure that it is easy for individual institution to add functional modules and to customize for their needs, and also easy to maintain and support.

The redesigned WMS is no longer a utility tightly coupled with Rutgers University Libraries but an open source digital object workflow management framework. Under this framework, the WMS development team provides core modules for WMS to perform basic functionalities and code libraries for partners to use for extending the capabilities of the WMS. The core modules conform closely to this design philosophy. They are highly modularized and can be reassembled as needed. The tasks for which each institution or organization will most likely have its own policies, or want to have its own implementations, are moved from hard coded software implementation to software supported configuration decisions. The handling of digital resource files gives a good example of this scenario. Some institution wants TIFF to be the archival master file format for images while others might prefer JPEG2000. The WMS no longer forces people to use TIFF but instead hands over the decision to each individual institution, allowing each institution to decide what format to support and what software to use to create each format. With respect to metadata, the metadata schema builder module is an attempt to decouple the utility from any specific metadata standard. Creating a one-for-all metadata converter is very difficult, and while we are working toward this capability, we haven't yet reached it. However, creating a utility that enables users to significantly modify, the existing schema and associated html form input fields has proven to be totally feasible.

Documentation is a big issue. There are two kinds of documentation related to a piece of software--documentation for the user of the software (e.g., a user manual), and documentation for the developer

or potential developer/supporter of the software. For open source software, both kinds of documentation are very important. RUL's current user manual includes a full data dictionary for every data element in the METS metadata architecture and a tour of the WMS, demonstrating both navigation and the purposes and use of each METS metadata document. The user manual reflects a much earlier version of the metadata. Since the manual was written, a complete rights metadata schema has been added and descriptive events were added to the descriptive metadata, to accompany provenance, preservation and rights events in other METS documents. The user manual will need to be completely rewritten before the WMS software is released. The RUL metadata architecture, which is based both on METS and an event structure in each METS document to capture the lifecycle of the resource in many dimensions. These are somewhat novel concepts for catalogers, who have mostly focused on the descriptive information captured in MARC or MODS. Our experience has been that the training and support for initial metadata users within the *New Jersey Digital Highway* has been significant, involving hands on training and considerable remote support. Training tools that encompass the user manual and beyond need to be produced before the WMS is offered as open source. This is not an easy task and is the subject of much discussion at present. The documentation for the software architectural and coding details needs equal attention because of the need for developers to understand the code to add modules or customize options for each organization. This task inevitably falls exclusively on the software architect and developers.

The WMS Open Source Process

The Rutgers University Libraries are moving cautiously into the open source environment with the Workflow Management System, primarily due to the sophistication and complexity of the metadata architecture and the modular, customizable design. Changes to the WMS are incorporated in RUcore versions, which are specified and implemented at least twice and sometimes three times per year. While versions are well documented for programmers and stringently tested, user documentation lags significantly and is generally several versions out of date. This has not been a real problem to date because the people using the software generally specified, approved and tested the modifications to the WMS. In the future, when the number and composition of users is unknown, user documentation will be very important. RUL is discussing how to handle documentation requirements when no position with a specific assignment to provide documentation currently exists. RUL also currently develops version

specifications based on the needs of Rutgers users, as articulated by library users, or dictated by grant requirements, and as authorized by the RUL cyberinfrastructure steering committee. RUL utilizes a modification request process that identifies bugs, either during testing or during routine use, and incorporates any bugs that don't actively interfere with standard use into future WMS versions. RUL is discussing how bugs identified by open source users, as well as recommendations for enhancements (or enhancements created by open source users) can be effectively incorporated into the WMS development and testing workflow Initially, UL is exploring the development of a small user community of institutions with similar needs, particularly institutions currently using the Fedora repository architecture, who can provide testing and collaborative development for the WMS. Currently, RUL is discussing an open source development collaboration for the WMS with Princeton, Northwestern and Penn State, all of whom are currently testing both the installation and the functionalities of the open source bibliographic utility based on the WMS that will be provided to the Library of Congress in 2008.

Future Developments for the WMS

One major development for FY08 is the integration of Encoded Archival Description (EAD) finding aids into the WMS. This integration will support both ingest and export of metadata as an EAD finding aid with associated digital objects. EAD support is critical because the special collections and archives of the Rutgers University Libraries base their description and access practices exclusively on finding aids. In addition, a document management capability will be added to enable users to automatically store and associate relevant PDF documents with events, such as a deed of gift with a rights event or a bill of sale with an accession event, within an administrative documents section of the repository.

Rutgers University Libraries are also recipients, with William Paterson University and NJEdge, the statewide Internet2 utility, of an Institute of Museum and Library Services grant to build a statewide digital video network, NJVid. RUL will be extending the WMS METS structure map to enable faculty and K12 educations to easily segment and annotate videos via a simple web form. RUL will also use XACML to place constraints on object use, again based on a simple web form completed by faculty, to enable faculty to reserve video course lectures to users within a course. NJVid will also implement a statewide Shibboleth facility that provides centralized Shibboleth services to all participating organizations, regardless of technical

readiness for Shibboleth support. Digital video functionalities will appear over the course of the grant, from FY08-FY10.

Conclusion

There are a number of open source bibliographic utilities that support the creation and organization of digital information for libraries and archives. We feel that the WMS makes several unique contributions, including the event-based data model, a complete METS implementation with fully functional METS documents for description, source, technical and rights metadata, and XML-based customization capabilities that enable users to tailor metadata to their needs while still supporting metadata standards and interoperability. RUL has also extended and enhanced technical and source metadata for digital multimedia, to support the needs of the moving image archives community. We feel the WMS will be a strong option particularly for organizations that want to document both analog source objects as well as digital surrogates and that want to document and manage resources with copyright and other digital rights issues. We also feel it is a strong utility for organizations wanting to insure interoperability with other initiatives, even as they customize metadata to meet local needs. We are addressing the remaining issues, that are primarily organizational and policy driven, to enable release of the WMS in open source in early 2008.

Notes

[1] "Presentations Given about RUcore" *Development of RUcore.* Accessed 1 November 2007. http://rucore.libraries.rutgers.edu/collab/index.php

[2] Fedora Commons. Accessed 14 October 2007. http://www.fedora-commons.org/

[3] *New Jersey Digital Highway*. Last updated 24 September 2007. Accessed 13 October 2007. http://www.njdigitalhighway.org/

[4] Library of Congress. *METS—An Overview and Tutorial*. 13 September 2006. Accessed I November 2007. http://www.loc.gov/standards/mets/METSOverview.v2.html

[5] RUcore—Rutgers Community Repository. Accessed 1 November 2007. http://rucore.libraries.rutgers.edu/

[6] Library of Congress. *MIC—Moving Image Collections*. Last updated 25 September 2007. Accessed 13 October 2007. http://mic.loc.gov/

[7] Library of Congress. *PREMIS—Preservation Metadata Maintenance Activity: Official Web Site*. Last updated 31 July 2007. Accessed 13 October 2007. http://www.loc.gov/standards/premis/

[8] Library of Congress. *MIX—NISO Metadata for Images in XML Schema: Official Web Site*. Last updated 17 August 2007. Accessed 13 October 2007. http://www.loc.gov/standards/mix/

[9] Audio Engineering Society Standards Committee. October 2003 Meeting of SC-03-06. c2007. Accessed 13 October 2007. http://www.aes.org/standards/b_reports/b_meeting-reports/aes115-sc-03-06-report.cfm

[10] Open Archives Initiative. Open Archives Initiative Protocol for Metadata Harvesting: Protocol Version 2.0 of 2002-06-14. Accessed 13 October 2007. http://www.openarchives.org/OAI/openarchivesprotocol.html