Fedora and the
Preservation of University Records Project

4.1 Analysis of Fedora's Ability to Support Preservation Activities

Version
1.0

Date
September 2006

Digital Collections and Archives, Tufts University
Manuscripts & Archives, Yale University

An Electronic Record Research Grant funded by the
National Historical Publications and Records Commission

Digital Collections and Archives
Tisch Library Building
Tufts University
Medford, Massachusetts 02155
http://dca.tufts.edu

Manuscripts and Archives
Yale University Library
Yale University
P.O. Box 208240
New Haven, Connecticut 06520-8240
http://www.library.yale.edu/mssa/

Co-Principle Investigators
Kevin Glick, Yale University
Eliot Wilczek, Tufts University

Project Analyst
Robert Dockins, Tufts University

This document is available online at
http://dl.tufts.edu/view_pdf.jsp?urn=tufts:central:dca:UA069:UA069.004.001.00011
(September 2006)

Fedora and the Preservation of University Records Project Website at
http://dca.tufts.edu/features/nhprc/index.html

Funded by the
National Historical Publications and Records Commission
Grant Number 2004-083

Fedora and the Preservation of University Records Project

PART ONE: INTRODUCTION

PART TWO: INGEST

PART THREE: MAINTAIN

PART FOUR: FINDINGS

TABLE OF CONTENTS

OVERVIEW

This report provides an analysis of Fedora's ability to support preservation activities.[1] This analysis is not as detailed as the Checklist of Fedora's Ability to Support Maintain Activities but is rather a broader examination of Fedora's ability to support the full scope of preservation activities, and not just Data Management and Archival Storage, which is the extent of the Checklist's scope. Rather than comprehensive analyze of Fedora's ability to support every preservation function, this report discusses two general capabilities of Fedora that cut across many preservation functions: security architecture and policy enforcement and resource records. The project team found it fruitful to carefully analyze Fedora's ability to support maintain services because Fedora is a repository architecture that at its core maintains digital objects. As a repository architecture, it presents a platform constructing a highly configurable repository. Thus, preservation functions such as Administration and Preservation Planning become highly dependent on their particular implementation, making a carefully mapped analysis of Fedora's ability to support these functions less fruitful (although still useful).

This report also examines Fedora's planned development and shift from a grant-funded project to a repository architecture maintained by a sustainable, community-supported organization and its ability to continue to support preservation activities. This report also discusses the difficulty of analyzing Fedora as a complete preservation system.

[1] See <www.fedora.info> for more information on Fedora.

ANALYSIS OF FEDORA AS A PRESERVATION SYSTEM

As described in the Project Overview, this project set out to prove the hypothesis that the flexibility and extensibility of the Fedora software would allow it to serve as a Preservation System. Over the course of the project, the project team's focus shifted from asking whether Fedora could serve as a preservation system to working on developing requirements for recordkeeping and preservation, the Ingest Guide, and the Maintain Guide. The focus changed in large part because the project team realized that it was asking the wrong question. Like many other archivists the team was looking for easy solutions to the very difficult problems posed by the long term preservation of electronic records. In hindsight, it seems obvious that no existing software application could serve on its own as a trustworthy preservation system. Preservation is the act of physically and intellectually protecting and technically stabilizing the transmission of the content and context of electronic records across space and time, in order to produce copies of those records that people can reasonably judge to be authentic. To accomplish this, the preservation system requires natural and juridical people, institutions, applications, infrastructure, and procedures.[2] As a result, Fedora cannot serve as the entire preservation system, but only as a preservation application, which is just a portion of the entire system. Without the appropriate people, infrastructure, policies, and procedures, even the best preservation application cannot ensure preservation.

In serving as the repository application of a preservation system, a Fedora instance (or instances) would be only one of many components that comprise a preservation system. Large portions of ingest and access activities and all preservation planning decisions, among other activities, would occur outside of the Fedora instance. Even though some preservation policies many be articulated and managed in Fedora, an institution still has to formulate these policies—they are not preset in Fedora. Rather than serving as an out-of-box repository solution, Fedora is a repository architecture upon which an institution can build a repository in many different ways. As a result, the suitability of Fedora as the basis of a preservation system depends significantly on its implementation.

The question we should have asked was: “Can a Fedora repository, surrounded by the proper preservation policies, tools, and Fedora services, serve as the basis of a trustworthy preservation system?” We feel the answer to this question is yes. The Fedora core provides a promising basis for a preservation system. Its agnostic view towards file formats and object types enables it to manage essentially any type of file. It has the ability to manage objects with complex—including hierarchical—relationships with its use of RDF or METS metadata. It can manage multiple bitstreams for a single object, which can enable archivists to track and store the original bitstream of an ingested record and the bitstreams of subsequent transformations. It has versioning and persistent identifier capabilities for all content objects, metadata, and disseminators. With Extensible Access Control Markup Language (XACML), an institution can articulate policies to help manage access to records. Fedora is a transparent system and Fedora objects are articulated in XML (usually FOXML or METS), making it feasible to migrate records out of Fedora.

[2] A preservation system has the same components as those of a recordkeeping system.

PLANNED FEDORA DEVELOPMENT

Future Management of Fedora Application

The Fedora Project, run jointly by Cornell University and the University of Virginia, currently manages the Fedora code base. The project is supported by a three-year Andrew W. Mellon Foundation grant that lasts through the end of 2007. This builds on initial design of Fedora at Cornell in 1997, a subsequent prototype implementation at the University of Virginia, and a 2001 Mellon grant that supported Cornell and Virginia's release of Fedora 1.0 in 2003.[3] The Fedora development team is currently developing a plan for moving Fedora beyond the grant-funded project stage to a sustainable, community-based organization that will take the responsibility for maintaining and developing the Fedora software and nurturing an active and growing Fedora community. The project directors have created an advisory board to help the project transition to sustainable organization led by a board of directors. The project directors have also established—or is in the process of establishing—the Outreach Committee to help foster growth of the Fedora community and manage Fedora's promotion; the Architecture Committee, which will manage and develop the specifications for the Fedora Service Framework; and three working groups, Preservation Services, Search Services, and Workflow Services that each investigate and undertake service development in their respective domains.[4]

This transition for Fedora being managed by a grant-funded project team to an organization supported by an active community is a crucial junction for Fedora's long-term ability to support preservation activities. To successfully serve as the repository core of preservation systems—many of which will demand a rigorous and high performance repository—Fedora will need a robust, clean, and well-constructed code base. As new technologies emerge, Fedora will need the organizational infrastructure to ensure that its software code is current and that these updates are smoothly added to existing code. Without a well-tended code base, Fedora's performance will lag and its ability to support scalable preservation activities will suffer.

Its object-oriented architecture and recently developed Service Framework (see below) has enabled Fedora to gracefully incorporate new services developed by the Fedora community. This ability to support new services should allow Fedora to serve as a repository core that can stay current with emerging preservation technologies, techniques, standards, and metadata schemas. However, this depends on a community developing the necessary Framework services that can embody—or at least communicate with—these new technologies, techniques, standards, and schemas. The Fedora community will also have to ensure that existing external tools needed for preservation activities, such as format validators and checksums, are able to become services within the Framework, guaranteeing that they will work smoothly with Fedora. The Fedora community will probably need some degree of leadership, either from a board of directors, the Architecture Committee, or the Preservation Services Working Group to provide a roadmap of preservation needs and priorities for new Fedora services.

The difficulty of moving beyond the project phase is a problem faced by all open-source endeavors and is not unique to Fedora. Like all open-source community-based efforts, Fedora

[3] "History," <http://www.fedora.info/about/history.shtml>.

[4] "The Future of Fedora," <http://www.fedora.info/community/fedorafuture.shtml>.

has the benefit of a diverse community building a range of services. For example, the work of the Search Services and Workflow Services working groups grow directly out of the efforts of particular members of the Fedora community. The community has to date developed a variety of Fedora tools and services.[5] If successful, this will grow into a rich array of tools and services that will allow Fedora users to select the appropriate resource from a sufficiently rich menu of options instead of building their own tools and services. On the other hand, like all open-source efforts, Fedora has the problem of finding an appropriate, dedicated entity to ensure continuity and sustainability. The Fedora project team aims to create that needed entity with its work to create an advisory board and later board of trustees, the committees, and the working groups.

Fedora Service Framework

As of version 2.1 (released February 2006), Fedora development will continue within the Fedora Service Framework, which establishes a structure in which new services that support a Fedora repository instance exist outside and independently of the repository.[6] New functionality for Fedora can be developed as a distinct stand-alone service that will interface gracefully with the core Fedora repository services. This allows new functionality to be developed in a more flexible, modular manner. Most importantly, it does not overburden the core repository software with endless new functionality. The project team expects the Fedora Service Framework to be the focal point of new development for the remainder of the Fedora project and beyond. Both the core development team and the Fedora community will contribute new services to the framework.[7] Two such services that have already been developed are Directory Ingest and OAI Provider. Members of Fedora community are currently developing additional open-source services. The Fedora core development team is also thinking of its own sustainability as an organization and is encouraging users and developers to continue working together to improve the application into the future. As a way of moving towards a development consortium that can sustain Fedora after the grant-funded project ends, the development team has initiated two groups that are of particular importance for preservation, one working on preservation services, the other on workflow.

Fedora Preservation Services Working Group

The Fedora Preservation Services Working Group is currently investigating and developing services to support preservation activities.[8] In 2006 it has focused much of its efforts on creating a messaging service that will support other preservation services in the Fedora Service Framework. The messaging service would serve as a generalized solution for sending messages to repository managers or machines about preservation-related events. The Fedora Development Team has begun work on message-enabling the Fedora Core and the Fedora Service Framework, which lay the architectural foundation for this messaging service. In addition, the Working Group has also spent time examining a variety of other services and their suitability for the Fedora Service Framework or as an external service that smoothly communications with Framework services or the Fedora Core. These services include format transformation, format validation, integrity checking, and repository histories.

[5] "Tools," <http://www.fedora.info/tools/index.shtml>.

[6] "Fedora Service Framework," <www.fedora.info/download/2.1/userdocs/server/features/serviceframework.htm> and <www.fedora.info/wiki/index.php/Fedora_Core_Repository_Service>.

[7] Internal Report from Thornton Staples to project staff, October 10, 2005.

[8] "Working Group: Preservation," <www.fedora.info/wiki/index.php/Working_Group_Preservation>.

Fedora Workflow Services Working Group

The Workflow Services Working Group has been formed in order to design and build a prototype set of business process or workflow orchestration services that could be used to load electronic records into a Fedora repository in a more automated manner. Such automation will be necessary for Archives to reduce the staff time required to process large volumes of electronic records.[9] There are several candidates for a standard to describe business processes using XML, including Business Process Execution Language (BPEL). This should enable an engine to "orchestrate" a business process by executing the steps, by messaging a human to begin a step and waiting for a specific response, or by running a computer program and waiting for the response. Complicated processes can be built up, allowing for concurrent steps or restricting them to be run in series.[10]

[9] "Working Group: Workflow," <http://www.fedora.info/wiki/index.php/Working_Group:_Workflow>.
[10] Internal Report from Thornton Staples to project staff, October 10, 2005.

FEDORA CAPABILITIES IN SUPPORT OF PRESERVATION ACTIVITIES

Most Fedora capabilities that support preservation activities center on maintain activities. The project team evaluates these capabilities in Checklist of Fedora's Ability to Support Maintain Activities. The project team examines two Fedora capabilities that cut across most preservation function: security architecture and policy enforcement and resource records.

Security Architecture and Policy Enforcement

Fedora provides a pluggable authentication module using Tomcat's standard approach to authentication, as well as a new access control module that enforces policies written in eXtensible Access Control Markup Language (XACML), an emerging standard that is beginning to be adopted on many fronts. Fedora has two plug-in modules for authentication: (1) a standard module that authenticates using a file of user identity and role information (i.e., tomcat-users.xml) and (2) an LDAP module to obtain user attributes from an LDAP directory. Fedora also provides an XACML-based policy enforcement module for authorization purposes. The choice of XACML allows institutions to record XML-encoded access control policies in Fedora, rather than in idiosyncratic database or file formats. XACML is very flexible and allows the specification of extremely fine-grained policies.

Fedora supports repository-wide policies, as well as object-specific policies. Policies can be written to permit or deny access to any Fedora API action based on attributes of the user, attributes of digital objects, and attributes of the environment (e.g., current date/time). This means that one can easily shut down all write-access to the repository by setting the policy for API-M at the highest level to deny all, while leaving the policy for API-A to allow all. This would result in a condition in which no changes could be made to the data in the repository, but read-access could continue.

Also, fine-grained object policies can be written to control access to a particular object as a whole, as well as its specific datastreams and disseminations. For example, an object could be set to have no public access until a certain date, or an image object could be set to allow free access to a thumbnail version while restricting all other versions to a specific group, such as a particular University community.

In a workflow that allows web-based users to submit records to a repository, administrators could change the permissions for an object as it progresses through the workflow. For example, once the submission is deemed complete by the originator, all content datastreams could be restricted from being changed, while the metadata datastreams could allow catalogers to update them.

Resource Records

The Ingest and Maintain Guides rely on resource records including resources like format information, record type information, submission agreements, producer metadata, retention schedules, knowledge base metadata, and the history of the repository itself. These supporting records can also be represented in a straightforward manner as Fedora objects with content models describing the kinds of information and abilities expected from such objects. Fedora also allows datastream versioning, which would be a very important capability for resource records whose content may change periodically. Encoded Archival Context can likely be adapted as a standard way for encoding producer records, but some way to encode information for the remaining supporting record types will have to be developed.