Long-term Repositories: Taking the Shock out of the Future

Two-day forum on PREMIS Preservation Metadata and the Trusted Digital Repositories

August 31, September 1

National Library of Australia
Topics for Discussion Today

- Working Group Formation
- Digital Preservation - Background and Philosophy
- Concept Architecture and the Digital Object
- Fedora Preservation Services and the Audit Checklist
Working Group

Vision: Expand the Fedora framework to facilitate the creation of trusted digital repositories

Objectives
- Define the requirements and architecture for preservation services that can be integrated into Fedora.
- Process focus: from ingest throughout the object life cycle
- Organize and coordinate collaborative development

Formation of WG
- From Fedora Users’ Conference at Rutgers (May, 2005)
- Charter Members from: Cornell, Harris, Northwestern, Rutgers, Tufts, Yale
Reference Documents

- RLG/NARA draft “An Audit Checklist for the Certification of Trusted Digital Repositories”


- OAIS Reference Model


Cullen (2000) asks rhetorically “How confident can we be when an object whose authentication is crucial depends on electricity for its existence?”.

“. . . the proliferation of experience, research, and infrastructure throughout the cultural heritage community has made trustworthy digital repositories conceptually realistic”**


Archiving Digital Objects
(Some questions)

- Can we define the “digital original”?

- Can we trace back from the nth migration to the digital original?

- Is this object format at risk of obsolescence?

- Can this object be properly preserved/migrated?

- Can/should we preserve dynamic behavior?
Digital Preservation Definitions
(There are many)

The long term maintenance of a byte stream (including metadata) sufficient to reproduce a suitable facsimile of the document, and continued accessibility of the contents thru time and changing technology. (Research Libraries Group)

The ability to keep digital documents and files for time periods that transcend technological advances without concern for alteration or loss of readability (Association for Information and Image Management)

“. . . the process of migrating a digital entity forward in time while preserving its authenticity and integrity” (Moore & Marciano, 2005)
The Digital Object

- The digital object is the basic unit of management, encapsulating all essential information about the “document” to be disseminated and preserved.

- The digital object should be independent of the environment where possible.

- Use standards and non-proprietary formats to minimize dependencies.
Events (Transformations) in the Life Cycle of the Digital Object

- Submission Information Package (T1)
- Digital Original (AIP)
- Migrated Derivatives (T2)
- Repository
- Output Device (T3)
Preservation Services
(Candidate Capabilities)

Object Level Features
- Audit trails and datastream versioning (available in Fedora 2.1)
- Persistent Identifiers (available in Fedora 2.1)
- Checksum creation and validation (active)
- Object format validation (active)
- Content model validation (active)
- Whole object versioning

System Level Services
- Event management and alerting (active)
- Repository redundancy/mirroring service (active)
- Format migration
- Enable Repository static/active states
- History service of major repository events
- Preservation planning – set up object life cycle policies
- Statistics reporting – ingests, purges, signature failures, etc.
Digital Object Integrity

- Ability to create and compare checksums on a datastream

- **On-Demand Checksum** - A new Fedora API to support client-initiated checksums.
  - **CreateChecksum** - Allows the application to request the Fedora repository service to create a checksum for a datastream.
  - **CompareChecksum** - Compares a checksum from “contentDigest” on a datastream to a re-computed checksum

- **Auto-Checksum option** – A repository configuration option that will automatically calculate a checksum of datastream content for every datastream in every object.
Events and Outcomes

- An event is an:
  - . . . action that involves at least one object, agent, and/or rights entity (PREMIS).
  - . . . occurrence that is significant to the performance of a task

- Event outcome – a situation or state that follows an event and is a result of the event.
Fedora Event Management

- **Generic Framework**
  - Events can have messages which are associated with all types of services (preservation, collection, user, etc)
  - Messages represent events with actions and outcomes
  - Fedora will provide a middle-ware messaging solution based on open-source Java Messaging Service (JMS)

- **Fedora Working Group Focus**
  - Preservation events are atomic (i.e. associated with a Fedora API)
  - The event message will be based on the PREMIS event entity
  - Initial types: ingest, delete, modify, fixityCheck
The Event Message

- Event message structure
  - The message payload will be xml-based and use the PREMIS event entity semantic units
  - Global identifiers (URIs) will be used for event type and outcome

- An example might look like the following:

```xml
<event>
  <eventIdentifier>
    <eventIdentifierType>Rucore event</eventIdentifierType>
    <eventIdentifierValue>30169</eventIdentifierValue>
  </eventIdentifier>
  <eventType>info:premis/preservation/event/ingest</eventType>
  <eventDateTime>2006-07-16T19:20:30</eventDateTime>
  <eventDetail>(to be used for general information)</eventDetail>
  <eventOutcomeInformation>
    <eventOutcome>info:premis/preservation/outcome/success</eventOutcome>
    <eventOutcomeDetail>(more text)</eventOutcomeDetail>
  </eventOutcomeInformation>
  <linkingAgentIdentifier>rutgers-lib:200</linkingAgentIdentifier>
  <linkingAgentIdentifier>rutgers-lib:400</linkingAgentIdentifier>
  <linkingObjectIdentifier>rutgers-lib:4291</linkingObjectIdentifier>
</event>
```
Event Management - Ingest
(Using the publisher/subscriber model)

User Input

Workflow Management System

JMS Topic Queue
- <eventType>ingest<>
- <eventType>delete<>
- <eventType>
- <eventType>

Digital Object Ingest

XML

Digital Object Repository (Fedora)

Preservation Service (reporting)

Preservation Service (alerting)
Content Models
(Content Model Dissemination Architecture – CMDA)

- The CM object specifies constraints on the digital object (DO)
  - MIME type and format
  - Min/max of number of datastreams
  - Whether multiple datastreams are ordered

- The CM is used to determine runtime behavior
  - On ingest, Fedora validates DO based on CM constraints
  - Disseminators are not bound into the DO
  - Run time binding occurs through the CM object and the rels-ext datastream
  - The CM can point to a format registry
Content Models and Disseminators
(A book example)

Book Object
- Persistent ID
- Metadata
- Rels-Ext
- Data streams
  - SMAP1 – StrMap (TOC)
  - DJVU1- presentation
  - PDF1 - presentation
  - XML1 – OCR text
  - ARCH1- Archival master (tiffs of each page)

Content Model
- Persistent ID
- Metadata
- Rels-Ext
- Composite Model
  - <dsCompositeModel>
  - <dsTypeModel ID="ARCH1" ordered="false" min="1" max="1">
    <form MIME="application/tar"/></form>
  </dsTypeModel>
- <dsTypeModel ID="SMAP1">

Bmech Object
- Persistent ID
- Metadata
- Rels-Ext
- WSDL

Bdef Object
- Persistent ID
- Metadata
- MethodMap

Format Registry
A Trusted Repository

is one “. . .that establishes methodologies for system evaluation that meet community expectations of trustworthiness; that can be depended upon to carry out its long-term responsibilities to depositors and users openly and explicitly; and whose policies, practices, and performance can be audited and measured”*

Certification Checklist
(How Fedora Preservation Services Can Help)

B. Repository Functions, Processes, & Procedures

B1. Ingest/acquisition of content
- B1.1 Repository identifies properties it will preserve for each class of digital object.  
- B1.3 Repository has an identifiable, written definition for each SIP or class of information ingested by the repository.  
- B1.6 Repository’s ingest process verifies each SIP for completeness and correctness.  
- B1.7 Repository provides Producer/depositor with appropriate responses at predefined points during the ingest processes.

B2. Archival storage: management of archived information
- B2.1 Repository has an identifiable, written definition for each AIP or class of information preserved by the repository.  
- B2.4. Repository has and uses a naming convention that can be shown to generate visible, unique identifiers for all AIPs.  
- B2.6. Repository verifies each AIP for completeness and correctness when generated.

B3. Preservation planning, migration, & other strategies
- B3.3 Repository uses appropriate international Representation Information (including format) registries.  
- B3.7 Repository actively monitors AIP integrity.  
- B3.8 Repository has contemporaneous records of actions taken associated with ingest and archival storage processes and those administration processes that are relevant to the preservation.  
- B3.9 Repository has mechanisms in place for monitoring and notification when Representation Information (including formats) approaches obsolescence or is no longer viable.
Certification Checklist
(How Fedora Preservation Services Can Help)

B. Repository Functions, Processes, & Procedures (continued)
B4. Data management
  B4.1 Repository captures or creates minimum descriptive metadata and ensures that it is associated with the AIP. Content Models

B5. Access management
  B5.2 Repository logs all access management failures, and staff review inappropriate “access denial” incidents. Event Management
  B5.6 Repository enables the dissemination of authentic copies of the original or objects traceable to originals. Audit trails and versioning

C. The Designated Community & the Usability of Information.
C3. Use & usability
  C3.2 Repository has implemented a policy for recording all access actions (includes requests, orders etc.) that meet the requirements of the repository and information Producers/depositors. Event Management
  C3.4 Repository has documented and implemented access policies (authorization rules, authentication requirements) consistent with deposit agreements for stored objects. Security – XACML policy enforcement

D. Technologies & Technical Infrastructure.
D1. System infrastructure
  D1.2 Repository ensures that all platforms have a backup function sufficient for the repository’s services and for the data held, e.g., metadata associated with access controls, repository main content, etc. Journaling/mirroring
  D1.5 Repository has effective mechanisms to detect data corruption or loss. Checksum compare
  D1.6 Repository reports to its administration all incidents of data corruption or loss, and steps taken to repair/replace corrupt or lost data. Event Management
Proposed Development Plan

- Core Fedora Development
  - Support for checksums on content bytestreams (Fedora R2.2)
  - Messaging service in Fedora framework (R2.3)
  - Formal expression and registration of “content models” (R3.0)
  - Object validation based on “content models” (R3.0)
  - Fedora API-M journaling and replay (for repository replication)

- Sun Center of Excellence Partnership – Rutgers University Libraries
  - Preservation services
  - Digital Preservation Portal

- Community Development
  We’re looking for those in the Fedora community who would be interested in developing preservation features and services.
Next Steps

- Continuation of First Year WG Activities
  - Decisions on WG renewal and continuation
  - White paper on reference architecture

- Possible Second Year Activities
  - Workshop on Fedora-based preservation
  - Possible grant applications
  - Initiation of community development partnerships
Membership in the WG

- Grace Agnew – Rutgers
- Paul Bevan – National Library of Wales
- Dan Davis – Harris Corporation
- Kevin Glick – Yale
- Ron Jantz (chair) – Rutgers
- Karen Miller - Northwestern
- Sandy Payette – Cornell
- Eliot Wilszek - Tufts
Digital Preservation Process
(Working Group Focus)
Event Management Concept Architecture
(Using Java Messaging Service – JMS)

- Systems & Applications (send/receive msgs)
  - Crawler
  - Fedora
  - Applications

- Fedora Messaging Framework
  - Closed Msg Pool

- JMS Msg Broker
  - Msg 1
  - Msg 2
  - Msg 3
  - Msg 4...
  - Msg n

- Msg Structure
  - Header
  - Property
  - Body (payload)

- Listening & Communication Services
  - Collection
  - User
  - Preservation