The Stanford Digital Repository, Version 2.0

Tom Cramer
Associate Director
Digital Library Systems & Services
June 2009
Agenda

• SDR to date
• Changes in the environment
• Version 2.0 design
• Observations & lessons
SDR is...

- a preservation system
- at version 1.7x
- in production since Dec 2006
- ready to jump to 2.0
Three Major Areas of Preservation Needs

• Digital Library
  – Legacy collections
  – Digitized collections
  – Licensed, locally loaded content
  – Born digital collections

• Institutional Repository
  – Research data,
  – Publications, dissertations,
  – Learning objects, university assets

• External Depositors
  – Publishers
  – Discipline-specific repositories
  – Reciprocal deposits with peer institutions

- Google Books (’00s of TB)
- Parker Manuscripts (75 TB)
- MJF Media (50 TB)
- NGDA (10 TB)
- ~30 other digi projects (15 TB)
- Purchased collections (25 TB)
SDR 1.0: Design Objectives & Assumptions

- Preservation-focused archive
- Replicated content
  - multiple copies, geographically distributed
- Auditable
- Simple
- Modular
- Tiered storage environment
  - online, nearline, offline
- Version rather than delete
- Content-agnostic
SDR 1.0: Core Repository vs. Repository Services

SDR Sample Use Case Diagram

- Ingest
  - BP Validation
  - Packaging
  - AIP Validation
  - Storage
  - OA
  - Geographic Distribution
  - Reporting
  - Cleanup

- Preservation
  - Risk Assessment
  - Policy Management
  - Triage
  - Preservation Actions
  - Modular Storage Migration
  - Format Migration
  - Technology Migration

- Access
  - Authorization
  - Negotiation
  - Retrieval
  - Packaging
  - Verification/Validation
  - Transmission
  - Logging
  - Cleanup

Digitized Book Use

Content Source

Selection

Digitization

Packaging

Reporting

(SDR Conceptual Boundary)

Geospatial Data Use

Content Source

Selection

Digitization

Packaging

Reporting

Sakai Data Use

Content Source - Sakai

Selection

Packaging

Reporting

Collections

- Shelves
- Files
- Documents

XML Data

- Person
- General
SDR 1.0: Assumptions

Core Repository Functionality

• Preserving access to digital information over time
  ...through generations of technology obsolescence and change.

• Maintaining integrity of that information over time
  ...through generations of migration and reformatting.

Repository Services Functionality

• All (or almost all) user-facing services
• Enhanced access & delivery through applications
• Data mining, dry research, new indexing, e-science, etc.
• Federation
E.g., The Parker Project

• 525 Anglo-Saxon manuscripts, 200,000 pages

• For each page:
  - 22 MB JPEG2000 delivery surrogate
  - 22 MB JPEG2000 delivery surrogate
  - 110 MB submaster TIFF
  - 220 MB master TIFF

Parker.stanford.edu:
Rich web application, tailored for general public, medievalists
SDR Component Diagram

Logging

Ingest
Validates, Packages (with TM), Sends to storage

Storage Manager
Directs objects to storage layers

Archival Disk
(Honeycomb)

Archival Tape
(Honeycomb)

TSM
L700

Archival
Cache

Access Director
Access requests; checks cache for content; queries & funnels requests to objects

Reconstructor
Reconstructs digital objects from AFIs to DIFs

Conversion
Generates TMs from existing MD

Staging Area
Acts as gatekeeper; value checking, file & format validation

Users

Applications

Fedora

Delivery Cache

SDR

Logging

Directory Watcher
Monitors and validates gatekeeper; transfers files to ingest

SDR WD

Access Requests from Accessors go to Access Director, which queries and funnels requests to Reconstructor, which reconstructs digital objects from AFIs to DIFs.
SDR today

- Version 1.7x stable & in production
- >60 TB of unique content
- >200 TB of managed data
- 14 different collections
- 4 content types: books, images, audio, manuscripts
- Basic & proven ingest and admin capabilities
- Tens of successful retrievals (bulk and atomic)
- 3.5 storage migrations
SDR today – 1.x tasks still to be done

- More robust administration capabilities
  - Automation
  - Monitoring
  - Reporting
  - Preservation actions

- More robust access capabilities

- More collections to ingest
  - Content types
  - Size

- Increased activity on audits
What’s Changed: Staffing

Current staffing @ 4.5 FTE
- ’07 reorg shifted 1 FTE to Web development
- ’08 reorg shifted 3 FTE to Infrastructure & Metadata team
- ’09 layoffs eliminated 1 direct and 1 indirect supporting position

Implications
- Reduction in dedicated staff
- Increase in complementary staff
- Challenges in continuity of knowledge
- Requirements to reduce & simplify SDR

2006
What’s Changed: SDR in Stanford’s DL Ecosystem

Specialty applications provide context-specific, user-facing deposit, and access services tailored to content types and disciplines

- **Library Management Applications**
  - EEMS (acquiring born digital content), digitization workflow, etc.

- **Institutional Repository**
  - ETDs, open access articles, faculty “papers”, research data, web sites, etc.

- **National Geospatial Digital Archive (NGDA)**
  - Geospatial data

- **SULAIR Digital Stacks**
  - Delivery for text, images, mss, media, data, & curated collections

**Stanford Digital Repository (SDR):** content agnostic, preservation repository

and SDR provides “back-office” preservation services: replication, auditing, migration, and retrieval in a secure, sustainable, scalable stewardship environment
SDR Serves As Common Preservation Infrastructure

Content Streams
- Digitized Materials
  - Google Books
  - Special collections
  - Media preservation
- Born Digital Content
  - Research data sets
  - Faculty & student writing, publications
- Purchased & Licensed Content
  - Ebooks
  - Ejournals
  - Databases
- Federated Content
  - Reciprocal content sharing agreements
  - Public domain information

Digital Library Holdings

Information Services provide scholars the prerequisite tools and environments to work with digital materials:
- Analysis
  - Text & data mining
  - Collection & management
  - Imaging & processing
- Collaboration
  - Data sharing
  - Group research & writing workflows
- Discovery
  - Indexing
  - Taxonomies
  - Semantic agents & processing
- Preservation
  - Web archiving
  - Preservation,
  - Datasets, research, archives,
  - Hosting & citation services
- Delivery
  - Books, images, media streaming
  - Data sharing

Digital Library Infrastructure (Content Management & Middleware)
- Identify
- Task
- Manage
- Report

SDR: Stanford Digital Repository (Preservation Infrastructure)
- Ingest
- Preserve
- Access

Hardware Infrastructure
- Storage
- Server

underlying Infrastructure supports content acquisition and user-facing information services

digital **Content** is the raw material for 21st century research & teaching
An Ecosystem of Activities to the Digital Library

Management Components
- Deposit
- Process
- Access Control
- Monitor
- ILS (Integrated Library System)
- DOR (Digital Object Registry)
- Work flow
- Report

Discovery Environments
- Search
- Fed Search
- Alerts
  - OPAC
  - Music
  - GIS
  - Gov Docs
- EAD
- Art
- Soc Sci

Delivery Tools
- Text
- MSS
- Web
- Data
- Image
- Media

“Portals” and Platforms
- Sakai
- Collab
- Drupal

Preserve (SDR)
Serve

Maintenance
Deposit, Management & Access Happen “Above” SDR

Deposit & Manage

- EEMs
- ETDs
- Research Data
- Digitization Workflow
- Google Books
- ILS

Digital Object Registry (DOR)

Preservation

SDR

Discovery & Delivery

- SearchWorks
- Socrates
- Text
- Images
- Media
- MSS
- Data

“Digital Stacks” Delivery Systems
What’s Changed: Surrounding Ecosystem

• Scoped repository: differentiation between 
  preservation (provided by SDR) and 
  ...content middleware (provided by DOR) 
  ...access (provided by the Digital Stacks apps)

• Implications:
  – Reduces pressure on SDR to be all things to all 
    depositors, for all content
  – Reinforces need to provide managed & secure storage at 
    scale
  – Reinforces requirement to focus on content integrity 
    services
  – Emphasizes need to integrate SDR to management & 
    access services through stable API’s
SDR 1.0: In Production & Working, BUT...

- Custom code, maintained by evolving & smaller team
- Bottlenecks
  - Need to be quicker to add new content types
  - Need to be quicker to add new collections
  - Need to increase the throughput to the storage layer
- Need to refine & hone the SDR service model
  - Need to deemphasize *curation* approach, shift away from “just in case” to a “just in time” mentality
  - Need to emphasize non-technical solutions & strategies
- Need to leverage other parts of the DL ecosystem
  - API’s for management, ingest, retrieval, reporting
  - Reuse of code
SDR 2.0

- Redefine (reduce & hone) service model
- Revise & streamline the data model
- Modify the technical architecture
- Reengineer the storage layer
- Integrate with other parts of the DL ecosystem
SDR 2.0: Refine the SDR Service Model

The Stanford Digital Repository (SDR) provides services to make scholarly resources available over the long term by helping ensure their integrity, authenticity, and reusability.

To fulfill its mission, the SDR must be secure, sustainable and trusted.
SDR 2.0: Refine the SDR Service Model

What does this mean?

SDR (the system) provides “back-office” preservation services: replication, auditing, migration, and retrieval in a secure, sustainable, and scalable environment.

SDR (staff) provide consulting, common-sense & commitment to long-term access for the institution.

Other (systems & people beyond SDR) will provide the domain expertise, curation functions, access support and overarching content management services.
SDR 2.0: Revise the Data Model

Currently use a METS-based “transfer manifest” as SIP, AIP and DIP, with many issues:

- Each Transfer Manifest is content & collection specific \(\rightarrow\) Doesn’t scale
- Transfer manifests require too much interpretation and analysis to change, augment
- Too complex: Stanford METS structure breaks apart related data across the object
- Wraps (somewhat dynamic) metadata with (mostly static) data files in same envelope
- Recursive nature of transfer manifest makes versioning self-referential, complex
- No one speaks METS natively: depositors, SDR & clients all forced to perform translation at handshakes
SDR 2.0: Revise the Data Model

• Revised data model currently in design
• Support “Zip ‘n’ SIP” for deposits
• Deprecate generation & storage of so much technical metadata
• Break object metadata into discrete files, track changes in change logs
  – Descriptive MD
  – Structural MD
  – Technical MD
  – Rights MD
  – Provenance MD
SDR 2.0: New Technical Architecture

• Adopt Fedora as a metadata management system
  – Clean mapping of new data model to Fedora content models
  – Reuse same design pattern, core technology as in DOR

• Support for highly parallelized operations
  – Multiple ingest streams to increase throughput
  – Decompose one process (e.g., “ingest”) into discrete, loosely coupled operations (“checksum”)

• Adopt a RESTful architecture
  – Shared services with DOR, access layer
SDR 2.0: Reengineer storage layer

• Current storage layer comprises:
  – Honeycomb for all MD and some data
  – TSM and tape robot for three LTO4 tape copies
  – Home-grown storage management system

• Throughput is acceptable for large objects, less than acceptable for many small objects

• 2.0 will adopt a full disk layer
  – Pool of SATA drives on Thumpers with an X4600 front-end

• Evaluating SAM-QFS vs. TSM & local storage management system
SDR 2.0: Reengineer storage layer

• Current storage layer comprises:
  – Honeycomb for all MD and some data
  – TSM and tape robot for three LTO4 tape copies
  – Home-grown storage management system
• Throughput is acceptable for large objects, less than acceptable for many small objects
• 2.0 will adopt a full disk layer
  – Pool of SATA drives on SUN Thumpers with an X4600 front-end
• 2.0 will support containerizing files for optimization for different storage platforms
• Evaluating SAM-QFS vs. TSM & local storage management system
SDR 2.0: Integrate with Stanford’s Ecosystem

- Fully automated download -> process -> ingest pipeline for Google-scanned books
  - Google -> DOR -> SDR chain
  - RESTful architecture of atomic web services
  - DOR manages object registration, validation, processing, indexing & packaging for deposit
  - Automatic handoff to SDR for ingest & call back to confirm successful deposit
  - Indexing & access to delivery surrogates managed through “Digital Stacks” environment

- Bulk retrievals from SDR for replenishing “digital stacks”, applying new operations
SDR 2.0: Integrate with Stanford’s Ecosystem

- Reuse of the Google Books pattern for preservation / management / access of other Stanford content
- “Hydra Project”
  - Reusable application framework that sits on top of Fedora
  - Basis for Stanford’s user-facing (institutional) repository services (electronic theses, open access articles, data deposit, etc.)
  - Support for deposit, edit, annotate, set permissions, search, browse, view objects in a Fedora environment
  - Also the basis for an SDR administrative user interface
Conclusion

• SDR 1.x has largely achieved its mission
• SDR 2.0 will address key gaps that have appeared
  – Focus on scoped set of services
  – Simplified data model
  – Reuse of technology (Fedora, Hydra)
  – Scale up on storage
• Mindset
  – Shift from “just in case” to “just in time”
  – Rely on integration with other services to provide holistic framework for management & access
  – Continue to stress steady progress over perfection