The Stanford Digital Repository
Lessons Learned – A Report for NDIIPP

December, 2009
Tom Cramer

Introduction
The Stanford Digital Repository (SDR) is a preservation repository designed to make digital assets available over the long-term by helping ensure their integrity, authenticity and reusability. In production since December of 2006, the SDR could be fairly termed a “first generation” repository, in that it represents the first, full-fledged, sustained attempt by Stanford to build a preservation system as core part of its enterprise information services.

By design, the Stanford Digital Repository serves as a back-office, content-agnostic preservation store for Libraries, University and scholarly community digital resources. (See Figure 1.)

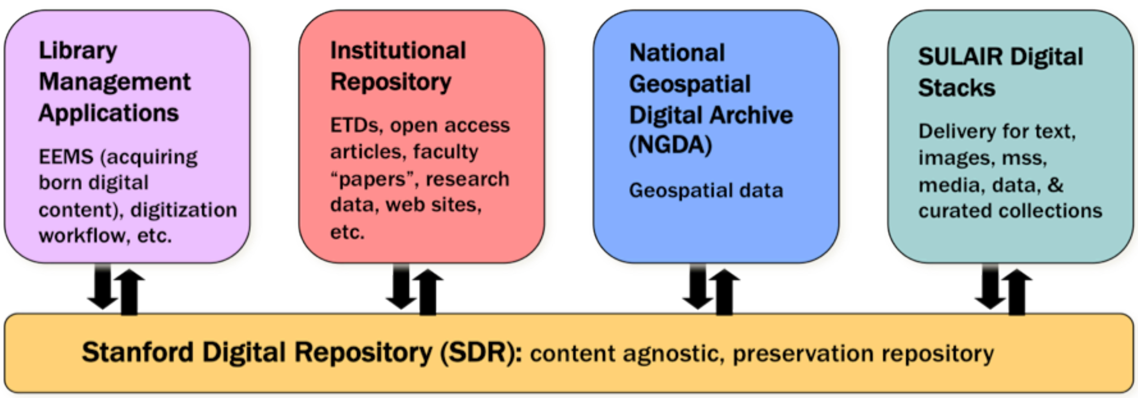

Figure 1: The SDR as back-office preservation infrastructure.

As of December 2009, the SDR has been relatively successful in achieving its primary mission. It is the preservation home for more than 80 TB of unique content (more than 300 TB of managed content), comprising tens of thousands of objects and millions of individual files. It holds five disparate content types (geospatial data, books, images, audio and manuscripts); has proven ingest, administration and retrieval capabilities; and as it has grown its contents have successfully migrated their media multiple times.

If it were to continue on its original trajectory, the SDR would be poised to incrementally add additional content types and collections, through an increasingly automated set of process; to enhance its administrative suite of tools to give repository staff and depositors increased views into and controls over ingested and managed content; introduce a more robust program of auditing, and build (or integrate with) a basic set of access / retrieval applications to facilitate content access.

But with five years of increased maturity in the preservation community since SDR 1.0 was originally conceived, three years of live operational experience, and substantial shifts in the underlying Stanford environment (both technical and organizational), SULAIR now has both the information and the need to revisit and revise SDR’s original architecture and service model, and produce a second generation system. While these changes are in many ways a natural extension of the SDR's current nature, collectively, the sum of these changes is substantial enough that we consider the new system to be "SDR 2.0".

Environmental Changes

Recent years have seen substantial shifts in the environment in which SDR operates. The three most significant shifts in the external environment since 2006 have been 1) substantial staffing changes, 2) development of a robust set of systems Library access and management systems, outside of SDR, and 3) maturity of the preservation function, community and supporting systems.

Staffing

When SULAIR assembled a team to focus SDR's development and operation in 2005, the team initially comprised ten staff, and consisted of a software architect, several engineers, metadata specialists, preservation analysts, and a pair of operations staff. Most staff were reassigned from other units, and brought their existing job responsibilities with them, and in reality, the team was far from being solely dedicated to the SDR development. That said, the swarming of the task with a sizeable team had the calculated effect, and SDR's first components were designed, built and operational within six quarters.

Now, three years later, the dedicated SDR team is less than half its original size, at 4.5 FTE. Further, only two of these FTE were on the original development team for SDR 1.0. Two factors contributed to this reduction; the first was the dramatic budget reductions University-wide resulting from the economic shock of 2008. The second factor was the constitution of two complementary, and the transfer of SDR staff to these functions.

The formation of these new teams actually served to bolster SDR's operations by offloading ancillary or supporting functions to other units, and allowing the core team to focus on development and operation of the repository. The cross-pollination from staff transfer also helped ensure understanding of and alignment with the SDR's services and operational needs, promoting better interoperation across the various units supporting the Stanford Libraries' digital efforts. The turnover in SDR personnel, and the overall reduced level of staffing, have been instrumental in the conception of SDR 2.0 as a simpler, better documented, and more narrowly scoped system.

Development of SULAIR's Digital Library Ecosystem

SDR by design is a back office system, designed to complement user-facing access and management systems for digital assets. During SDR's original build out, these other systems were largely conceptual or existed but integration was a distant concern. This forced SDR to compensate for their absence by taking on several general purpose management services (specifically content analysis and preparation) and access functions (optimizing packaged content for delivery). In the intervening years, these front-office systems have taken shape and begun to deploy. On the management side, SULAIR has developed a *Digital Object Registry* (DOR) that serves to register, track and relate digital content regardless of its location in the digital library. Based on Fedora, DOR orchestrates the services and workflows necessary to accession and manage digital content, and also prepares assets both for preservation (in SDR) and access (via Stanford's growing suite of digital library applications). The advent of DOR has provided a scalable, robust system for content receipt, conversion and packaging upstream of SDR. DOR has also provided a successful technology pattern combining Fedora (as a metadata management system), RESTful web services, and workflow to process objects. (See Figure 2.)

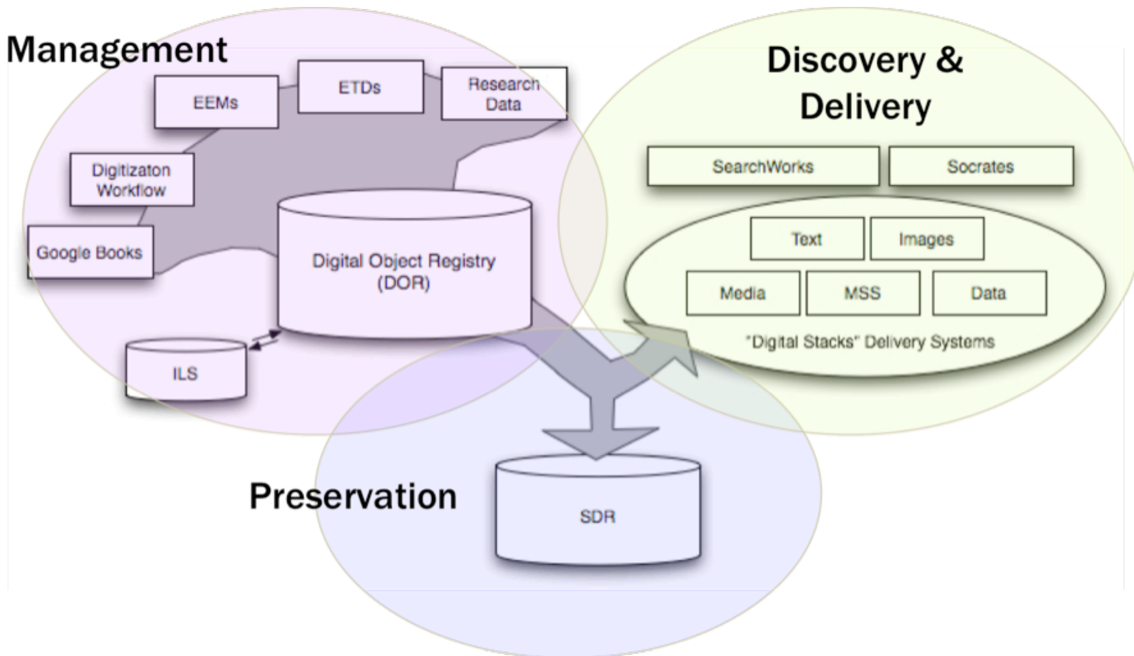

Figure 2: Stanford's digital library ecosystem has three main spheres: Management, Preservation and Access.

Stanford's digital library environment has also matured to include the beginning of a platform for access (here defined as both asset discovery and delivery) in a suite of tools that we collectively term "the digital stacks." The digital stacks comprise a suite of Ruby on Rails based applications, with solr index metadata stores and NFS-based digital asset data stores, URL and location resolvers, all integrated with Stanford's enterprise authentication and authorization services. Adopting Ruby on Rails-based applications allows for the rapid development (and shared components) of flexible (and ultimately disposable) front-end interfaces, tailored to specific content types (books, images, media, geospatial data, etc.) or disciplines (e.g., Medieval Studies). The definition of this digital stacks platform and the initial development of applications in this suite has given the SDR a clear destination and data form into which it must ultimately be able to deliver at least its library assets.

Maturity and Understanding of the Preservation Function, Community and Systems
The initial development of SDR included a technology strategy to adopt or buy components where possible, and to build where needed. Based on an initial marketplace survey (including assessments of both Fedora and DSpace in 2005), Stanford opted to build modules for ingest, storage management, administration and access, and to purchase storage management subsystems (Tivoli Storage Manager from IBM for library and tape management; Sun's Honeycomb system for online storage). This decision was also influenced by the strengths of the team, a sense that the marketplace had not yet focused on this space, and a desire to keep the system as streamlined and modular as possible, unencumbered by ancillary functions (and limitations) that might be imposed by shoehorning a product into a function other than its original purpose.

Further, the exact nature of and necessity of digital preservation was at the time quite open ended. What type of preservation planning would be required? How much emphasis would be needed on format migration? How often would we conduct fixity checks and perform media migrations? What other types of preservation actions would be needed? What type of management and administrative functions would be most critical to maintaining the repository? The general understanding that informed much of SDR's original design was that necessity, and best practice, would both emerge over time, and that the system requirements would correspondingly evolve. Further, it was well understood that the SDR mission and capabilities would need to develop in an environment of constrained resources. Opting to build most of the higher level logic locally offered the best promise of the flexibility required to adapt as needs became better understood.

Since this time, the increased maturity of the digital preservation marketplace, and a deeper local understanding of its strengths and weaknesses have also influenced SDR's path. In 2005, when there was no clear (at least to Stanford) front-runner technology in the digital preservation arena, the predictable complexity and challenges of building local modules seemed acceptable. In 2008, the capabilities of the Fedora repository as an object management framework, in combination with an increasingly obvious rate of community adoption, made it a compelling choice as a module in SDR's architecture for metadata management.

First Hand Operational Experience

Three years of production, operational experience have also been instructive in helping inform the requirements and design of a second generation SDR. While many of the questions on the exact nature and need of digital preservation functions remain unanswered, a very clear set of first order objectives in building an enterprise scale digital preservation system have emerged. New scholarly resources of different types—data format, content structure or intellectual context—must be able to be acquired, processed and ingested with no more than a modicum of incremental analysis, and no major development. The system must be able to accommodate large collections, in terms of memory consumed or number of objects, or both. All modules require sufficient logging and reporting capabilities to track and administer content through the entire stack. Deposit and retrieval processes must be simple and flexible enough to support hand-offs that work for all parties, with little to no engineering work. These lessons in priority and necessity, garnered through first hand operational experience as well as the published wisdom of the digital preservation community, have seasoned the thinking and design behind SDR's second generation incarnation.

Physical Bottlenecks: Compute Cycles, Bandwidth and Storage

While the original SDR design prioritized a modular architecture (with ingest very distinct from conversion, storage and access, e.g.), experience has shown that further atomizing functions within a function would benefit both overall throughput and manageability. Rather than running ingest as one continuous pipeline, for example, SDR's redesign breaks this into discrete functions (i.e., checksumming, virus checking, format validation, AIP writing, AIP validation), each capable both of being run in parallel instances, and of being invoked asynchronously.

In its current configuration, SDR's second major physical bottleneck lies in its tape storage subsystem. While large objects are written and replicated with acceptable throughput to each of three TSM-managed tape copies, speed of ingest for small objects in great numbers drops to an unacceptable level due to the transactional overhead of establishing a unique connection for each object. An obvious work-around has been to "containerize" numerous smaller objects into fewer large objects on ingest; a more flexibly and systematic approach will be required for SDR 2.0, however.

Logical Bottlenecks: Data Modeling and Conversion Analysis
SDR's original data model used a METS-based "transfer manifest" as a submission information package, and with slight modifications, also its archival and dissemination information packages. While the SDR data model emphasized the reuse of the administrative and technical metadata sections across like object types (e.g., a book from one set of content should and did greatly resemble a book from another content pipeline), the transfer manifest design necessitated a great deal of analysis and sometimes complex packaging scripts to be written for new content types. This represented a substantial bottleneck in taking in new formats. The data model also preserved as much descriptive metadata as possible for incoming collections, and, in anticipation of optimizing the population of as-yet-to-be-built discovery and delivery systems, attempted to capture and structure descriptive metadata in as standard a way as possible across diverse collections, while losing none of the nuances. In this case, a text from one collection might indeed look different from another text, even if structurally identical. Overall, this might be described as a "just in case" or approach, with each new content type and collection requiring a substantial analytical effort to determine a detailed and appropriate transfer manifest. Ultimately, experience demonstrated this tailored, curatorial approach to data modeling for each content stream was not sustainable. It became clear that to apply preservation services to the diversity and scale of content demanding it, SDR would need to shift its focus to more of a "just in time" approach, and the phrase "Zip 'n SIP" came into frequent use as an alternative strategy to increase the rate of ingest.

Beyond the lessons learned about streamlining the idiosyncrasies of the transfer manifest, the SDR experience with METS was instructive in planning for an updated repository. While it is certainly possible to have a fairly straightforward METS schema, and it is arguably an excellent, widely adopted package for transmitting digital objects, Stanford's METS schema was designed to leverage more of METS' sophistication and power, and was used as the package for objects throughout the repository. This created several issues that complicated the operation of SDR. This included multiple layers of wrapping, abstraction and references that bloated objects, and introduced challenges in both interpreting and manipulating them. Perhaps most critically, experience indicated that neither depositors nor clients spoke METS natively, and that with both SIPs and DIPs wrapped in METS, every use case for ingestion and delivery required non-trivial conversions to translate from or to a useful package for the depositor or designated community for that content type. With these lessons in mind, SDR's revised data model deemphasizes the use of METS within the repository itself, and will support it as a dissemination package if and when required.

Conclusion

The Stanford Digital Repository has largely achieved its original mission. With three years of continuous operation, it has grown to support more than 80 TB of unique scholarly assets, comprising hundreds of thousands of digital objects in a diversity of formats. With numerous successful media migrations and significant changes in staffing, the Stanford's preservation system has navigated the first of its ongoing sustainability challenges. That said, the last few years of operational experience and shifts in the environment have shown the need for a revised service model, system architecture and overall preservation strategy.

First, SDR's future service profile can be firmly scoped around a few core functions ensuring content fixity, authenticity and security. Content deposit, accessioning, conversion and overall management occur "above" SDR, orchestrated through a digital object registry. Content access, including discovery and delivery to scholars and the general public, occur in purpose-built access systems, in Stanford's "digital stacks". This separation of concerns allows SDR to focus its efforts on large-scale content ingestion, administration, selective preservation actions and limited retrieval. Upstream conversion processes, and rich discovery and delivery systems will be supported through well-defined API's.

Second, SDR's technical architecture will address and improve on the critical priorities that have emerged in operating the first generation repository. These include adopting Fedora as a metadata management system to leverage the community's investment in and ongoing support for an open source platform that aligns well with SDR's overall technical design. Experience has also shown the need to decompose functions into more granular and loosely-coupled services (i.e., from "ingest" to "checksum"), both for increased control of processes as well as for throughput. Finally, the preservation subsystems will require balancing support for accommodating large objects and a multitude of smaller objects.

Third, SDR's data model must shift to reduce the incremental analysis and development required to support new content types and collections. Content files will be stored in directories following the BagIt design, with metadata files stored in discrete chunks, leveraging Fedora's object design and XML management capabilities.

Taken individually, the changes along any one of these vectors represents an incremental enhancement; taken altogether though, these changes are substantial enough to move the Stanford Digital Repository to a second generation system and set of services.