OR2018 Presentation Submission
13th International Open Repositories Conference, June 4th-7th, Bozeman, Montana, USA

Defensive Design: Developing a System-Agnostic Repository for Sustainable Long-Term Preservation

Katherine Lynch, University of Pennsylvania Libraries, katherly@upenn.edu; Emily G. Morton-Owens, University of Pennsylvania Libraries, egmowens@upenn.edu

Session Type

- Presentation

Abstract

Colenda, the University of Pennsylvania Libraries' digital repository, was designed to promote long-term preservation. Its infrastructure is comprised of components selected to concentrate on factors that are of the most importance and that pose the greatest risks for long-term preservation of digital assets: safe file storage, the ability to track changes to objects over time, mechanisms for object management and discoverability, and migration paths that guarantee that objects can be safely migrated to new software and new versions of existing systems while preventing data loss. Favoring a pluggable architecture and preservation of software-agnostic representations of objects in order to keep future repository development plans flexible and open, our approach minimizes the risk of data loss in the long term and has allowed us to design a system in which the right tools for the task are always an option. In this paper, we will enumerate the risks/concerns influencing our design decisions and show how our approach addresses them while retaining a connection to the central open-source projects of the community, Fedora and Samvera, that make up significant portions of our stack.

Conference Themes

- Open source software - sustainability of software developed locally and large open source systems, legacy code
- Content - research data, digital preservation, persistent urls, archiving
- Teams/People - staff and knowledge within the community, contingency planning, training and development, and succession planning
- Infrastructure/Integrations - integrations between systems, changing technical environments
- Challenges of sustainability - funding, local, technical, community

Keywords

Repository architecture, sustainability, modular architecture, digital preservation, risk management

Audience

Developers, administrators, repository managers

Background

In 2016, the University of Pennsylvania Libraries began building a digital repository designed for long-term preservation of digital assets, offering open access to as much of our material as possible. Although the library had a legacy display interface for digitized books and manuscripts, and a hosted institutional repository for scholarly research, this was our first foray into taking on digital preservation requirements and factoring in the full lifecycle of our objects. In designing the architecture and features of Colenda, as the repository became known, we built on lessons learned at other institutions in the open-source community and by examining the broad scope of our assets, both digitized and to-be-processed and based numerous decisions on efforts to mitigate risks. In our repository software, we model a highly generalizable, easily adaptable workflow to accommodate not just the objects we process today, but the unknowable requirements of objects in the future. Finally, to factor in sustainability, we developed our software using Samvera and Fedora, open-source frameworks for digital repositories with highly collaborative, active communities of development that ensure we do not walk this path alone.

Content

Introduction

Before the present authors joined the staff of the University of Pennsylvania Libraries, its Repository Services Team had decided to hold off on developing a digital repository because of the community's transition from Fedora 3 to 4. The library had a display solution for some types of materials and a hosted institutional repository, but no preservation-focused general-purpose repository. By early 2016, the addition of a new Senior Developer and AUL, and the maturity of Fedora 4, allowed the Repository Services Team to begin serious work on designing a digital repository for this purpose. This late start allowed the team to benefit from many lessons learned at other institutions, projects in the open-source community, and Penn's existing platforms and the ways in which staff and users interacted with them.

Colenda (from the Latin for "to be cultivated or protected") is a Samvera-based repository backed by Fedora 4, with Ceph and Glacier serving as file storage layers. While our infrastructure has commonalities with many other Fedora and Samvera projects, some unique aspects of our architecture are the result our risk analysis.

Software

In our legacy display interface, the only way to manage objects was as a part of a collection; everything must be part of one collection, so dummy collections had to be created to house one-off objects. Further, this design imposed a drag on object processing, as the entire collection should be finalized before being posted. In Colenda, objects *can* belong to collections (and *should*, for curatorial responsibility), but the collection relationship is a piece of metadata rather than an architectural requirement. Objects can also belong to more than one collection, which is a more realistic model of how they are used in practice. Breaking this mold has required multiple exploratory conversations with curatorial staff, and we are already seeing the benefit of reducing overhead for processing arbitrary groupings of objects for the repository.

The objects targeted for inclusion in the first phase of repository development were digitized books and manuscripts that were already displayed through the legacy interface. However, because they were not covered by any digital preservation plan, they were considered an immediate priority. Many stakeholders were so focused on this material that there was a risk of over-specifying the design and creating an interface that only worked for objects like these books and manuscripts.

However, the Samvera-based ingestion workflow is designed to allow staff users to upload any type of object and any type of metadata, regardless of format, object structure, or metadata sources. The repository's metadata extraction workflow interprets information found in spreadsheets to assemble descriptive and structural metadata for objects, adhering to the repository's metadata schema. The user simply uploads a spreadsheet and, through a user interface, describes it so that the application is able to find important pieces of metadata in it. This flexibility allows us to offer some amount of workflow convenience for staff users while preventing the imposition of a single metadata standard as a prerequisite for inclusion in Colenda, and in turn makes the repository a natural solution for preserving increasingly unique sets of metadata.

The workflow for our legacy display interface was brittle, with new objects being instantiated by a script that ran overnight. This was attractively simple but in practice, if the script encountered an error or unexpected condition, or if a mistake had been made in assembling the files, the process failed and had to be repeated the following night. Additionally, reasons for the process's failure were sometimes difficult to pinpoint through troubleshooting, necessitating multiple reruns over multiple days before issues were resolved. In Colenda, there are distinct inspection steps that occur before ingestion, during object processing, that allow the staff user to correct any issues in real time. The Fedora ingest then occurs as a backgrounded process. Further, our legacy interfaces were designed for specific collections, which means that a new collection must have a display designed for it or be slotted into an existing one, however awkward. In Colenda, no content needs to wait for further design steps for ingestion.

To address large batches of like objects that require little or no additional human examination, we have built several automated workflows to either speed the operators' work or even automatically ingest objects that have already been curated with QAed images and metadata spreadsheets that follow a particular format.

Colenda's primary tools for interacting with Ceph are Git and git-annex, an extension that allows Git to version files by tracking file content and location in a key/value store where the key is derived from a checksum of the file's content. By using a tool that prioritizes tight version tracking and secure file transfer, we not only provide a safe mechanism by which to transparently move files to and from Ceph, but we also monitor changes to files over time, with the ability to revisit the entire history of an object whenever changes are made to it, deliberately or otherwise. In this way, we have capitalized on a tool not originally designed for preservation purposes (but to manage large files efficiently) to let us use the familiar Git paradigm while enjoying improved preservation characteristics.

Finally, in an effort to keep development efforts around our repository software sustainable, Penn Libraries elected to design Colenda within the Samvera/Fedora framework. Adhering to community models such as the Portland Common Data Model (PCDM) for structuring objects, we are able to participate in vibrant, active open-source communities for repository software, making decisions in line with the greater visions of these communities, serving common interests around long-term preservation, complex object modeling, and filesystem transparency.

Infrastructure

At the infrastructure level, we perceived a threat from the expense of our SAN storage system as well as the difficulty of expanding its capacity. The SAN system requires the purchase of expensive, proprietary disk hardware. Because it cannot support disks of unlike capacity, increasing its size requires all the disks to be swapped out, which is expensive and involves downtime. This system seemed a poor match for Colenda, which we expect to grow in fits and starts, especially due to

grants that periodically allow us to digitize large quantities of material, including from outside partners (meaning that our own university library's holdings of a certain type of material do not provide a ceiling on how much we may digitize). Instead of the SAN, we chose to build Colenda on the Ceph storage system. Ceph is a software-based block storage system that offers fault tolerance, high availability, and scalability by maintaining multiple copies of objects across its nodes, with self-healing functionality in case of node failure. Its self-management allows us to rely on cheaper, commodity disk storage; it is also sophisticated enough to be able to manage objects across a non-symmetrical node structure which allows us much greater freedom in adding storage capacity.

In addition, Ceph stores the files in a transparent way, addressing them by their own hashes. We felt that this was also an advantage, as it allows repository managers to view their files in a transparent and reassuring way.

The University of Pennsylvania has a relatively loose federation of schools and departments, with the result that the library runs its own data center instead of receiving service from campus IT. Storing our data in the same building as (in many cases) the collection objects that have been digitized is an obvious risk. We address this by including Amazon Glacier as a backup of our storage in Ceph; we have chosen the Oregon location because it is in a different disaster zone than Philadelphia. Both Ceph and Glacier are addressable using the Amazon S3 APIs, which keeps the software streamlined.

Organization

While we have described several deviations from prior practice in the Fedora/Samvera community, it's important to note that this project is fully based on that software and that we embrace the resilience that a multi-partner organization brings to library software, particularly to our project where much of the technical work has been done by a single developer.

Our pluggable architecture with its distinct layers of responsibility, conversely, insulates Colenda against the risks that come with relying on software that is not solely controlled by our institution. While it would be no trivial project, it would be possible to replace any layer of the infrastructure if needed. Because the objects have their own metadata embedded in them, we could even bootstrap an entirely new repository from the objects themselves.

Outstanding concerns

There remain some concerns that we do not feel our design adequately addresses; these are targets for future work.

Our de-emphasis of collections as an organizing principle for object management puts Colenda at risk of seeming, to users, to be a confusing sea of unrelated content or different granularities of content once its scope grows beyond the initial book and manuscript collections. We are currently employing faceted search to mitigate this and hope to adopt Spotlight to create exhibits that bring a curatorial voice to the display.

The design's completely agnostic view of file types means there is some danger of scope creep; there is no technical barrier to putting practically anything in Colenda. Despite the recent departure of the library's preservation officer, the Repository Services Team continues to work on policy documents to manage expectations. Further, the inclusion of documents that are still evolving could create a large volume of unwanted versions, so we are exploring the idea of allowing repository managers to squash-merge versions in certain cases.

The development of custom workflows for ingestion presents a concern about the sustainability of developer effort. It would be possible to create tools at almost any degree of customization for

specific collections, no matter how small. This is not necessarily the best use of local developer time given the roadmap of desired features. The level of customization in the first phase of development has raised a threat of unsustainable expectations. We are working with curatorial staff to help them learn to use the system independently and develop their own generalized curatorial approaches for project-based metadata models for objects that are part of heavily-customized projects, but that are also intended for Colenda.

Conclusion

In building Colenda, the University of Pennsylvania Libraries have made specific technical and policy decisions that could be adopted by other institutions and are based on experiences we have learned about at home and through hearing from others at events like Open Repositories. Through the entire stack, we have attempted to address sustainability through a pluggable, flexible, transparent architecture. Wherever possible, the design of Colenda is agnostic about tools, content types, and collection scope, allowing policy to dictate practice and object support rather than software limitations. We have attempted to make any part of Colenda replaceable—and that includes ourselves!

Repository System

- Fedora
- Samvera
- Ceph/Glacier/S3 storage systems