Project Overview and Goals

Overview

The wide adoption of digital repository software by cultural heritage organizations has led to significant increases in effective management and access to digital assets. While these software systems may provide some digital preservation features, the digital materials and their associated metadata managed by such systems need to be preserved in a way that ensures they will persist over time. This project addresses this need by developing a specification for an integration model that will allow libraries and archives to seamlessly deposit system content into distributed digital preservation systems (DDPs) such as Chronopolis, APTrust, and LOCKSS. This project is funded by the Mellon Foundation.

Background

While digital repository systems are generally backed-up, this back-up tends to involve managing just one copy of the data. As such, it does not fulfill standards-based digital preservation requirements, which mitigate risks associated with the lack of geographical redundancy, technological diversity, and also human error or malfeasance, all of which can be accomplished by distributing multiple copies of deposited data across a network of nodes which are based in different geographical locations and managed by different entities using different storage infrastructure stacks. Distributed digital preservation systems (DDPs) like Chronopolis, APTrust, and LOCKSS are designed to fill this preservation gap, and many organizations have adopted these services in conjunction with their local repositories. However, DDP services and local repositories are not designed to work together and their incongruencies present a multitude of issues for organizations wishing to employ both types of systems. Typically, local repositories are driven by access concerns, and their design and internal workflows are directed thusly. For example, these systems tend not to track preservation events (e.g. synchronization timestamps, checksum comparisons) but are more focused on access information (e.g. access counts and downloads). In contrast, DDP services are focused on the status of data among their nodes, so things like synchronization timestamps and checksum comparisons are crucial. Currently, there are no good options for users who are running these systems to efficiently (or automatically) link these two kinds of systems in a way that allows for management of data between them. To address this problem a new specification is needed that will facilitate preservation-quality deposit from local repositories to DDPs.

This grant proposes to complete the architecture overview, technical specification, and user interface specifications needed to integrate Hyrax, a Samvera-based repository built on top of Fedora, with the Chronopolis DDP service. The project will bring together representatives from Hyrax, Fedora, Chronopolis, DuraSpace and APTrust in order to ensure that the proposed work is representative of community needs and can be extended for use in other DDP systems. The project outcomes include a detailed specification for software development that will enable the integration between Hyrax and Chronopolis as well as a generalized set of user stories and specifications that can be used to develop integrations between other local repositories such as DSpace and DDPs. Hyrax and Chronopolis were selected for this proposal because of existing working relationships between the software communities. The outcomes of this project could be incorporated other DDP systems such as LOCKSS.

Goals

The project goals are:

1. To define requirements for an interface or dashboard for collection curators to send digital objects from their local Hyrax/Fedora repository to a DDP (specifically Chronopolis, but configurable to other DDPs) with a simple click of the mouse, allowing for updated versions to also be automatically delivered to the DDP service.

2. To define the development work needed to integrate the Hyrax/Fedora, DuraCloud Bridge, and Chronopolis systems in order to allow for integrated object versioning and reporting between the systems.

3. To define the requirements for the user interface for version information and tracking of data sent to a DDP service within the Hyrax/Fedora interface, as well as information about data replication and audit throughout Chronopolis.

4. To ensure that the created definitions, specifications and design documents are applicable to other digital repository software and other DDP services. Project personnel will spend significant time working with project partners and advisors at all stages, with the goal that the outcomes created are in sync with other systems and services.

These project goals are designed to provide a model for how local repositories should interact with DDP services. Currently, the entire process of sending local repository data to any existing DDP service is manual. Data residing in the local repository is exported out of the system using the export functionality provided by the repository and then ingested into the DDP network using the tools provided by that network. It is important to note that an institution may not want to send all of the data in their local repository to a DDP service; these services can be expensive, and the institution may decide that not all of their repository data warrants that level of preservation. In these cases, there is no way for the local repository to internally track which materials have been sent to a DDP service.

In addition to identifying the use cases the overall technical architecture will support, one of the first tasks of this project will be to determine a versioning mechanism that is implemented by the local Hyrax/Fedora repository system. This versioning mechanism needs to be able to export relevant versioning information in a way that can be tracked by Chronopolis. Specifically, a versioning mechanism should help ensure that new versions of objects and/or object metadata can automatically or manually be sent to Chronopolis and Chronopolis will maintain the versioned relationships in a manner that will make it easier to restore the objects into the local Fedora system. One possible means of accomplishing this versioning process is through the implementation of the Oxford Common File Layout (OCFL) specification. This new specification is a community-led initiative designed to provide a standard mechanism of arranging objects and their versions on the storage filesystem in a manner in which their version relationships are essentially human-understandable. Even though the OCFL specification is fairly new, it is based on the Moab design for object versioning, which has been developed and used by Stanford University for a number of years.10 The Moab design is recognized as one of the few (if not only) systematic approaches designed to address versioning considerations within a digital preservation system. Versioning is a key issue for the work in this proposal because it is one way to track how requested changes in an access system (i.e. Hyrax) are propagated and tracked in a preservation system (Chronopolis). Because OCFL is based on Moab, it has versioning included within its default set of functions. The OCFL initiative started in late 2017 and the editorial committee is expecting to have an initial specification release by fall 2018. This project will review OCFL as a possible versioning mechanism and incorporate it into the system design if it provides support for the project's selected use cases. Two of the project participants serve on the OCFL editorial committee and will be able to provide expertise on the specification.

In addition to increased version interoperability between Hyrax/Fedora and Chronopolis, this project will also outline the specification needed to deliver information about the object once it is distributed into the Chronopolis network. This information includes preservation event data such as date of registration into Chronopolis, date of replication to each Chronopolis node (and node location), and date of last audit.