A Technology Analysis of Repositories and Services

Final Report

Submitted to the Mellon Foundation
March 28, 2006

Introduction

This document provides the final report for "A Technology Analysis of Repositories and Services." With funding from the Mellon Foundation, the Sheridan Libraries at Johns Hopkins University has conducted an analysis of repositories and services based on a methodology for connecting user requirements with repository programmatic features. The Sheridan Libraries considered a diverse range of content types and end user services by developing and gathering numerous scenarios from multiple institutions, and collaborating particularly with MIT, UVA, and ProQuest to evaluate DSpace 1.3.2 (http://dspace.org), Fedora 2.0 (http://www.fedora.info), and Digital Commons (http://www.proquest.com/products_umi/digitalcommons). In all cases, we worked with the "out of the box" system and documented APIs. It is important to note that our analysis focused on the ability of each of these systems to support specific functionality through documented APIs. Future work should include additional analysis of other means for supporting functionality (e.g., user interface or application based import or access), and of additional systems (e.g., ePrints).

During the Mellon Foundation's Research and Instructional Technology (RIT) Retreat in 2006, MacKenzie Smith described three aspects of interoperability: semantic, protocol and functional. This analysis examined the protocol aspects by assessing the existing protocols of JSR-170 (http://www.jcp.org/en/jsr/detail?id=170), Digital Repository OSID (DR OSID, DR OSID, http://www.okiproject.org/), and the eduSource Communication Layer (ECL, http://ecl.iat.sfu.ca/) and the functional aspects by testing the documented APIs from the aforementioned systems that can interface readily with applications.

While the specific results from this analysis are noteworthy, it is worthwhile to affirm the importance of the methodology and the recommendations for next steps. All project materials, including final results, are available at the main project wiki:

https://wiki.library.jhu.edu/display/RepoAnalysis/ProjectRepository

Methodology

Different audiences often refer to different concepts when using the term "repository." In order to bridge the different perspectives, we proposed a methodology that included scenarios, use cases and repository features. The details regarding these concepts are described in the original

proposal, on the project wiki and within the interim project report (also available from the wiki). Our initial idea rested upon the premise that a scenario, an "individual instance of use cases that traverse a specific path using specific data", represents the most accessible description of needs from the end user perspective. Faculty, students, collection managers, etc. can most readily describe what they need to do with various content types in a story format, rather than by defining technical requirements (or speaking the language of developers or programmers).

From these scenarios, we attempted to draw an explicit connection between elements defined in the scenario and specific repository features, which would be mapped to documented APIs. This connection would allow different individuals to understand repository needs in different contexts. For example, an end user might focus on scenarios to identify or articulate particular needs whereas a developer or programmer might focus on the repository features that relate to the scenarios. Initially, we felt that moving from scenarios to use cases to repository features would provide an explicit path for mapping between end user needs and technical specifications. However, our experience over the course of the project led us to alter this approach. We ultimately identified a set of repository features that encompasses a broad range of content types and service requirements, though the connection between the scenarios and repository features is *implicit*, reflecting the tacit knowledge of the project team gained through this analysis and previous repository-based projects such as the Archive Ingest Handling Test.

We deliberately lowered the barrier for submitting scenarios. In our experience, it is difficult, and perhaps even confusing, for end users to directly define their needs of repositories. By allowing end users to provide scenarios in the most accessible manner, we managed to create or collect eighty-three scenarios from seventeen organizations, which are available at:

https://wiki.library.jhu.edu/display/RepoAnalysis/ProjectRepository_Scenarios

While this approach contributed to our success with collecting scenarios, in many cases, end users did not tell a story, but rather provided a description of an existing system. Scenarios would also describe actions at such a high level such that it was difficult, if not impossible, to infer the interaction with the repository (or even if a repository would even be necessary to support the particular scenario). Consequently, scenarios did not often directly illuminate specific, desirable repository features. As an example, many scenarios assumed object level functionality (e.g., create object, modify object, etc.), yet did not provide specific statements in this regard.

Our initial methodology included the creation of use cases based on these scenarios, which are available at:

https://wiki.library.jhu.edu/display/RepoAnalysis/ProjectRepository_UseCases

As we attempted to harvest specific events from the scenarios into use cases, we discovered that the use cases typically offered a "vertical" description of an end user's interaction with a system. In this analysis, we were more concerned about the "horizontal" dimension—the ability for repositories to support a diverse breadth of content and uses. Additionally, as mentioned during

the final project meeting with our collaborators, use cases are most helpful when designing a new system or software, not when classifying or delineating characteristics of existing systems.

For these reasons, we developed the concept of a "key event", which is defined as "an interaction with the repository by an element of the system." These key events initially formed the basis for defining repository features, which were mapped to the APIs of the various specifications and applications that we analyzed. A description of the key events is available at:

https://wiki.library.jhu.edu/display/RepoAnalysis/KeyEvents

The key events proved more useful than use cases when considering repository features, especially for object specific actions. However, even the current instantiation of the key events proved inadequate for an explicit connection between the scenarios and repository features. In the interest of moving the analysis forward, we decided to identify repository features through an implicit process of carefully examining all scenarios and making appropriate inferences.

As an example, consider the scenario SharingContentMultipleInstructors available at:

http://wiki.library.jhu.edu/display/RepoAnalysis/SharingContentMultipleInstructors.

While this scenario does not explicitly mention versioning, it is reasonable to assert that this repository feature would be helpful, if not essential, to support the activity. In the scenario, two users create a new course in Sakai, a collaboration and learning environment, out of resources from an old course. The resources may need to be changed for the new course. If both users edit the same resource, one of the users' changes will be lost. Depending on the underlying repository and the timing of the edits, a user may not realize data has been lost.

Suppose the repository used by Sakai has versioning support. Each time a resource is edited, the repository automatically creates a new revision of the resource. Sakai could then give users the ability to examine all the revisions of a resource and, if needed, roll back to a previous revision. No data would be lost.

It would have been ideal to draw an explicit connection between scenarios and repository features. There are several ideas in the "Recommendations for Future Work" that might facilitate such a connection. However, it is worth mentioning that the tacit knowledge gained from this analysis, and from prior repository-related work at Johns Hopkins (including the Archive Ingest Handling Test) provide the solid, objective foundation for identifying the set of repository features, which are available at:

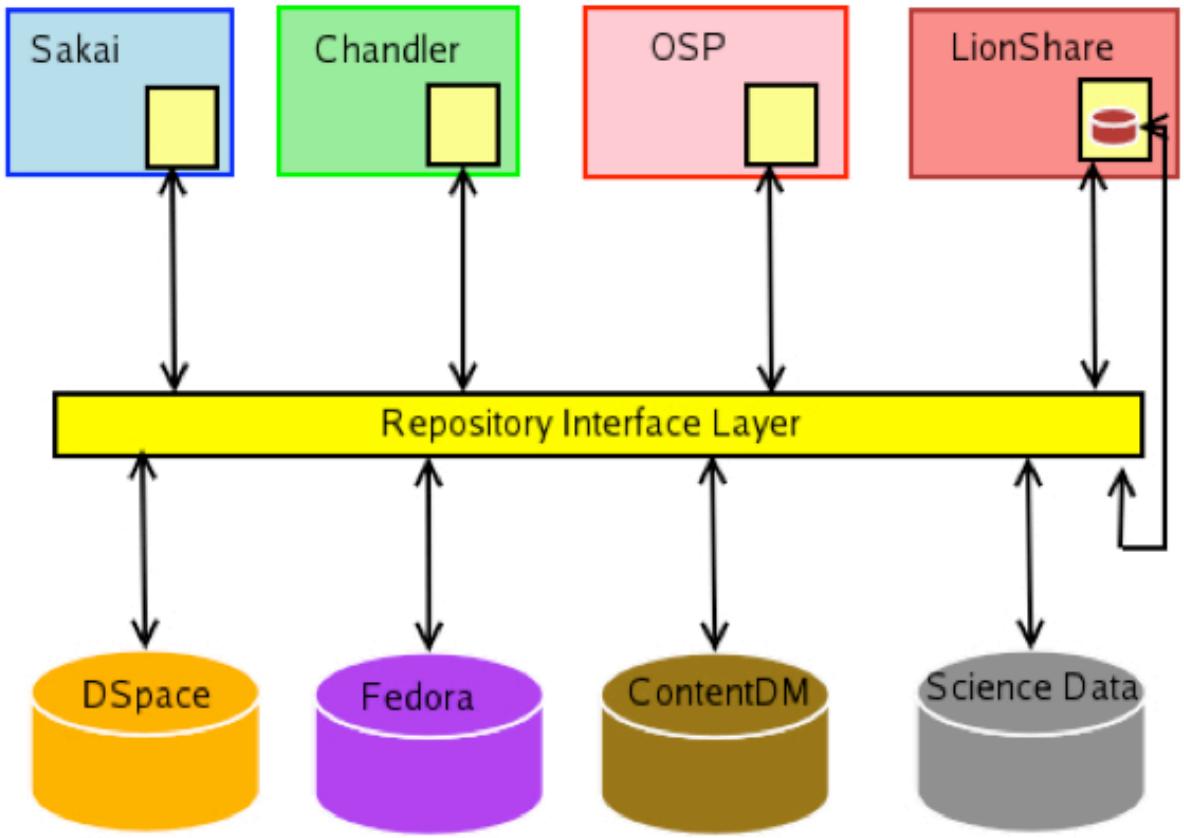
https://wiki.library.jhu.edu/display/RepoAnalysis/Features

This page contains our list of repository features that support the content types and user needs identified through the scenarios, and explains the different sections of the repository API evaluations.

Results

Motivation for this analysis relates to Johns Hopkins' belief that it will be necessary to support multiple repositories and applications through a repository interface layer. Ideally, this architecture would include multiple, agnostic, distributed repositories and applications such that organizations can choose different systems in an open, modular manner. This concept was presented during both the Spring 2005 meetings of CNI and DLF[1] and discussed in a First Monday paper available at http://www.firstmonday.org/issues/issue9_5/dilauro/index.html

The diagram below illustrates a set of applications against which to compare repository functionality, and a reference for mapping the subjects of our evaluation (DSpace, Fedora, Digital Commons, JSR-170, OKI DR OSID, ECL) to the various layers. This diagram also makes it easier to illustrate new service models, where the repository might reside elsewhere.



[1] Both presentations are available at http://ldp.library.jhu.edu/projects/repository/documents

We have presented results in both summary format and in detail for each repository interface. The summary results are available at:

https://wiki.library.jhu.edu/display/RepoAnalysis/ResultsSummary

The detailed results are available at:

DSpace -
https://wiki.library.jhu.edu/display/RepoAnalysis/DSpaceFeatures
Fedora -
https://wiki.library.jhu.edu/display/RepoAnalysis/FedoraFeatures
Digital Commons -
https://wiki.library.jhu.edu/display/RepoAnalysis/DigitalCommons_Features
JSR-170 –
https://wiki.library.jhu.edu/display/RepoAnalysis/JSR170Features
OKI DR OSID –
https://wiki.library.jhu.edu/display/RepoAnalysis/OKI_OSID_Features
ECL –
https://wiki.library.jhu.edu/display/RepoAnalysis/ECL_Features

Project Meeting with Collaborators

As part of the final debriefing for this analysis, we held a meeting comprising the Hopkins project team, MacKenzie Smith from MIT, Thorny Staples from UVA, Jeff Riedel from ProQuest, and Mahendra Mahey from JISC's Digital Repositories Support Team. While Smith and Staples appropriately indicated that both DSpace and Fedora have active communities, they were invited to represent DSpace and Fedora, respectively. Riedel was invited to provide feedback regarding Digital Commons. Mahey was invited to build upon the interaction and collaboration between this analysis and JISC's Digital Repositories Programme.

All participants discussed the methodology and results from this analysis. Each of the collaborators provided excellent feedback, which informed the final actions for the analysis, and the recommendations for future work. Perhaps most importantly, each collaborator indicated that there is potential for confusion, or even incorrect interpretation or understanding regarding the capabilities of each system. While our analysis focused on a specific lens, that of working through documented APIs, each collaborator explained that there might be other methods to support repository features. As an example, consider the repository feature of versioning.

Fedora offers utilities built into the system that provides inherent versioning by creating new objects every time a change is made to an old object (optionally, old datastreams may also be retained). During the final debriefing meeting, we asserted that DSpace and Digital Commons do not support versioning, given our focus on supporting such functionality through documented APIs. Smith pointed out that versioning support in DSpace might be offered by using

appropriate metadata.[2] Riedel indicated that Digital Commons works through a service contract arrangement, so if a particular institution needs versioning, they would develop a mechanism to be shared with the potential customer.

This exchange highlighted the complexity of making assertions regarding repository features, and the importance of appropriate presentation of results. Our original idea emphasized the creation of a table or matrix for presenting results, with a diagrammatic schema that would reflect full, partial or no support (e.g., the Consumer Reports circle-based rating system). A recent example is Thom Hickey's analysis available at:

http://outgoing.typepad.com/outgoing/2006/03/repository_comp.html

As noted on this blog post, he asserts that Fedora (and WikiD) supports versioning, but DSpace and ContentDM do not. Given our discussion with our collaborators, it seems that listing support for repository features in this manner may be incomplete, or even inappropriate.[3] While such a representation offers a simple and easy-to-comprehend format, it may mask the inherent complexity of answering such (seemingly simple) questions.

Consequently, the presentation of our results on the project wiki does not attempt to tabulate these results, or provide checkmarks, or any other simplifying notation or representation. The results pages on the wiki are intended for developers or individuals with technical knowledge, who can assess the *programmatic* capabilities of each system. It is essential to realize that there may be other methods to support particular repository features, so making assertions for partial or no support must be made very carefully.

Advice for Decision Makers

Many libraries find themselves asking questions about repository deployment in their local environments. It is completely understandable that a library director or dean would expect a relatively simple response when asking the question: What system will we use for our repository?

Our analysis has demonstrated that this question results in a host of complex considerations, including the real possibility that multiple systems may be required, or even a combination of in-house and external options.

Perhaps the more appropriate question is: What services do we need to offer our constituents?

[2] Further information is available at http://simile.mit.edu/dspace-mit-docs/versioning.pdf

[3] It is entirely possible that OCLC would also assert that versioning might be supported if a customer requested it as a feature.

As early as 2003, Cliff Lynch made the following statement: "In my view, a university-based institutional repository is a set of services that a university offers to the members of its community for the management and dissemination of digital materials created by the institution and its community members."[4]

While the scenario-based methodology did not proceed as we had originally envisaged, there is still great potential in asking end users to articulate their content and service needs in this manner (especially noting the recommendation in the final section of this report). Once these needs are well defined, it would be worthwhile to develop a service contract with local and/or external technology providers, who can assess which system(s) may be most appropriate. That is, decision-makers should consider the end user needs that, once translated into technical features, would be addressed by the technology providers. This approach is preferable to beginning with the question of which system should one adopt.

Recommendations for Future Work

Based on our findings from this analysis, and feedback from the final project meeting, there are a few recommendations that would further support decision-makers and technology providers to make informed, rational choices.

It might be possible to specifically highlight the repository aspects of scenarios with additional work. Specifically, it would be useful to approach individuals who submitted scenarios to revise them to focus on the repository aspects, or to create a set of a few archetypal scenarios that combine elements from the existing set. Additional examination or consideration of the key events might also highlight the explicit connections between scenarios and repository features. These activities would help decision-makers understand which scenarios reflect most closely their particular constituents' service needs.

During the course of this analysis, four individuals contacted us through the wiki or by email to evaluate additional systems: ePrints, ContentDM, CDSware, Nesstar and Virtual Data Centre. This analysis focused on DSpace, Fedora and Digital Commons for a few reasons. Johns Hopkins had been evaluating DSpace and Fedora, especially through the Archive Ingest Handling Test. This expertise, the open-source nature of the systems, and our existing contacts within the respective communities ensured that we had the necessary access and resources to conduct a proper analysis. ProQuest approached us following the Spring 2005 CNI meeting, and discussed the evaluation during a follow-up meeting at the annual ALA conference. They assured us that they would give us appropriate access to the system and available APIs, and that they would not interfere with the analysis in any manner. They should be applauded for this decision, which hopefully other vendors will consider.

It would be useful to conduct a similar analysis for additional systems, but it is necessary to have appropriate access and freedom to conduct the analysis in an objective manner. It would be worthwhile to follow up with individuals associated with ePrints, ContentDM, CDSware Nesstar

[4] http://www.arl.org/newsltr/226/ir.html

and Virtual Data Centre (and perhaps others) to ascertain whether such an arrangement is possible. Additionally, it would be useful to conduct additional analysis with more recent versions of DSpace and Fedora.

Finally, there are other lenses through which a repository analysis might be conducted. Even in the absence of APIs, it may be possible to support certain repository features through applications or add-on modules, or through associated metadata. Some may prefer the web-based, user interface for ingest and access offered through Digital Commons. An analysis of capabilities offered through applications, add-on modules, metadata, or the UI of each system would be worthwhile.

One of the clear lessons from this project is that this type of analysis is very resource intensive and time consuming, much more so than we had originally anticipated. Each of the recommendations for future work is a major effort, and while Johns Hopkins now possesses relevant expertise and experience, it seems clear that a broader or community-based effort might be more appropriate. It is our hope that our ongoing dialogue with JISC might build upon this analysis, including through the JISC E-Framework, or that our involvement with both NDIIPP and DLF might provide venues for additional work.

Making informed, appropriate choices for repositories is one of the most important issues facing libraries today. We thank the Mellon Foundation for its support of this project, which will contribute to addressing this important choice.