The Big Migration: Lessons Learned at the Completion of the 10-year DRS2 Project

Andrea Goethals
National Library of New Zealand
70 Molesworth St, Thorndon, Wellington
New Zealand 6011
Andrea.Goethals@dia.govt.nz

Tricia Patterson
Harvard Library
90 Mt Auburn St, Cambridge, MA
USA 02138
tricia_patterson@harvard.edu

ABSTRACT

This paper shares lessons learned at the completion of a large multi-year project (2008-2018) to move to a next-generation digital preservation repository at Harvard Library. These final stages included activities running in parallel to migrate the metadata for millions of files, train fifty-five different units to use the new repository tools, and migrate the audio files from obsolete formats.

Harvard Library's digital preservation repository, the Digital Repository Service (DRS), holds over 69 million files and 285 TB of data from across the university. When it first launched in October 2000, it was one of the first digital preservation repositories at an academic institution. At the time it was designed, there were little-to-no digital preservation/digital library standards, best practices or tools to adopt. Planning for a next-generation DRS under the name the "DRS2 project" began in 2008, with the purpose of modernizing the DRS to take advantage of the latest technologies, standards, and practices and to provide curators, depositors, and preservation staff with significantly enhanced tools.

The DRS2 project included four different types of migrations: infrastructure, metadata, file format, and repository users. It included such daunting tasks as re-architecting the entire repository; re-parsing all of the millions of files; changing the underlying data model, Archival Information Package (AIP) format, and all of the XML schemas; rewriting all of the repository metadata; retraining all of the repository users; and reformatting all of the audio deliverable files. A repository migration of this size and breadth was unprecedented in the digital preservation community, so there were few preexisting examples to learn from. By sharing the experience of this project, the authors hope that other institutions can benefit from the lessons learned during a repository migration of this magnitude.

This paper provides an overview of all four migrations, but delves deepest into the results, challenges and lessons learned from the metadata migration. These lessons are applicable to preservation planning and interventions, future migrations, preventing metadata and content errors, and conducting very large projects in general.

CONFERENCE THEME(S) ADDRESSED

Lessons learned within and across domains, Technological infrastructure

KEYWORDS

Migration; Preservation Repository; Data Models; Audio Files; Metadata Migration; Format Migration; Technical Infrastructure

1 INTRODUCTION

1.1 DRS2 Project

When the DRS launched in October 2000, it was one of the first digital preservation repositories at an academic institution. During the years after the DRS was put into production, new functionality was added incrementally. Some of the modifications were relatively small, such as developing a more efficient ingestor, while other changes were larger, such as adding new viewers for delivering content to researchers and other users. In 2007 it became clear to the maintainers and staff users of the DRS that continuing modifications to the repository codebase and infrastructure were no longer sufficient. Planning began for a transformative change to the design, technology, and functions of the DRS, under the name of the DRS2 project.

The purpose of the DRS2 project was to modernize the DRS to take advantage of the latest technologies, standards and practices and to provide curators, depositors and preservation staff with significantly enhanced tools. Specifically, the goals were to (1) replace aging technical infrastructure, (2) implement the latest digital preservation standards and best practices, (3) provide more collection management functions, (4) support preservation planning and activities, (5) improve access to content and metadata, and (6) support more digital formats. The DRS2 project included four different types of migrations: infrastructure, metadata, file formats and repository users[1]. This paper provides an overview of all four migrations, but delves deepest into the results, challenges and lessons learned from the metadata migration.

1.2 A New Approach to Metadata

While the DRS2 Project included broad changes to the repository infrastructure, the original impetus for the project was largely due to the perceived deficiencies of the repository metadata, Archival Information Package (AIP) "packaging", and the underlying data model. When the DRS was first designed, there were little-to-no digital preservation/digital library standards or best practices to

[1] The DRS is used by fifty-five different units at Harvard University, including libraries, archives, museums and departments. For the purpose of this paper, any staff within these departments who deposit to, or manage, content in the DRS are referred to as the repository users, or just "users."

adopt. Most of the metadata schemas used in the DRS had to be custom-created. The metadata varied in quality and accuracy. The metadata elements had grown organically over time as needed. Some of the metadata elements were not specific enough to support preservation planning. For example, the file formats were recorded at a very coarse level (e.g. TIFF), not documenting the specific "flavor" of the format. Some elements were overly restrictive in the permitted values. For example, only two values were permitted for the text character set, which in practice meant that one of these values was recorded even if incorrect. Some elements were not restrictive enough, permitting free-text fields, or were interpreted in a variety of ways rendering the meaning hard to parse. Some generic elements, such as methodology, could only be recorded for particular formats. All of the metadata was contributed by staff in a variety of roles, ranging from professional staff working in reformatting labs, to curators and librarians who made occasional deposits. There was very little automatic validation of the metadata. In summary, the metadata was insufficient and not accurate enough to support long-term preservation planning. To improve the DRS metadata, the relevant standards and schemas that had since been developed were adopted where possible as part of the DRS2 project. The most fundamental change was to adopt the PREMIS data model [7]. The key metadata changes are listed in Table 1.

A key part of the metadata redesign was implementing strategies for improving the DRS metadata throughout the lifecycle of the digital content under preservation management, for example:

- During ingest, automatically generate the technical metadata including format-specific metadata.
- During ingest, or on request, pull descriptive metadata from catalogs, the Harvard LibraryCloud [4] or other systems.
- During ingest, or when files are added or removed from objects, validate the object according to its content model.
- During migrations, sync descriptive metadata with catalogs/the Harvard LibraryCloud, and try to fix metadata errors.

The central idea was to iteratively improve the repository metadata over time in both an intentional and opportunistic way, rather than assume that the best and final metadata will be deposited with the object.

Table 1: Key Metadata Changes Made as Part of the DRS2 Project

	Old DRS (before DRS2 Project)	**New DRS (after DRS2 Project)**
Repository data model	One data entity was explicitly modeled – files. For some types of content, objects could be derived by dereferencing relationship metadata.	Hierarchical data model in which each data entity is explicitly modeled based on the PREMIS data model - objects, files, bitstreams, and associated agents, events and rights statements.
Object categorization	Object types were not explicitly documented in metadata.	Every object conforms to a DRS content model describing rules for valid repository objects, valid file formats and relationships, known delivery and rendering applications, associated assessments and preservation plans.
Object characterization	Minimal descriptive metadata could be stored at the object level in a METS descriptor, for some types of content.	Descriptive, administrative, rights, and provenance metadata can be stored for any object in a METS descriptor.
File characterization	Minimal technical metadata was manually supplied by depositors.	Generic and format-specific technical metadata is automatically determined by a new tool (FITS). Additional technical metadata, that cannot be automatically determined by tools, can be supplied by depositors.
Metadata storage and preservation	All of the metadata was stored in a database which was backed up. For some types of content, the metadata was also stored in a METS descriptor.	Every object has a METS metadata descriptor file containing all of the metadata for the object and its files. The descriptor file receives the same preservation treatment as the content files (replicated storage, integrity checking, etc.). The metadata is also stored for convenience in a database and index.
Metadata schemas	In most cases, the metadata schemas were created at Harvard specifically for the DRS.	Where available, metadata schemas created by standards bodies or community efforts were adopted. The administrative metadata schema, while still custom to Harvard, was redesigned to better support curatorial and preservation management. The preservation metadata schema supports provenance and rights statements.

2 FOUR TYPES OF MIGRATION

2.1 Sequencing of the Infrastructure, User, Metadata and File Format Migrations

At a high level, the DRS2 project work was sequenced in the following way:

1. Redesign the metadata foundation using the PREMIS data model, standard and community-accepted metadata schemas, the new AIP design using METS descriptor files, and object content models (2008)
2. Assess open source and other preservation repository software for suitability for the new DRS (2008)
3. In parallel: (2009 - 2013)
    a. Begin infrastructure migration (develop/install new tools built on the redesigned metadata foundation)
    b. Begin repository user migration (training on new data model and metadata concepts)
4. Start planning for the metadata migration (rewrite of repository metadata into the new format) and for the repository user migration (switch over to using the new repository tools) (2013 - 2014)
5. Perform the metadata and user migration in parallel (2015 - 2018)
    a. While migrating the metadata for the audio objects, perform a format migration on the audio files (2018)

2.2 Infrastructure Migration

Although the infrastructure migration is not the main focus of this paper, a summary of it is included here to give readers a sense of the overall scope of the DRS2 project. After assessing the repository software available at the time, it was determined that there was not an existing open source or commercial product that could meet the needs of the Harvard Library. The options that were available at the time (2009) did not have the extent of features already present in the DRS suite of software. It was decided that the preservation and curatorial requirements could be most successfully met by implementing a new solution rather than working from one of the existing solutions. The changes to the infrastructure made as part of this project included:

- A redesign of the repository architecture to use RESTful services and APIs, and to support high-risk confidential and other sensitive material
- New modular applications for building batches for deposit; ingest; content, metadata and repository management; and delivery of content in various formats to end users
- Integration of the File Information Tool Set (FITS) [2], an open-source tool for format identification, validation, metadata extraction and generation
- Integration of the Object Tool Set (OTS), a tool for reading, writing, and modifying METS metadata files containing PREMIS preservation metadata, format-specific technical metadata and administrative metadata
- New database and index schemas to support search and reporting
- Integration with a new email archiving tool (EAS) [3]

2.3 Planning for the Metadata and User Migrations

For this project, the metadata for all content in the DRS needed to be migrated, i.e. regenerated from multiple sources, rewritten into newly-generated METS files which would be ingested into the new DRS, and copied to a database and SOLR index. The content files themselves did not need to move from their storage locations. After metadata was migrated to the new DRS in its new format and location, it could only be managed by the new tools specially built for the new DRS. This meant that staff within Harvard units who deposited material to the DRS, or managed content already in the DRS, would need to use the new DRS2 tools once their material's metadata had been migrated to the new DRS. For these reasons, the metadata and user migrations had to be planned together and closed coordinated. To inform the migration and user migration process, two types of analysis were performed – a technical analysis and an analysis of the DRS users.

The technical analysis involved grouping the content into virtual buckets that could be migrated at one time, determining any dependencies between content types, and determining databases, catalogs or other systems where metadata could be pulled from during the migration. To drive and track the migration, six new metadata elements were added to the old DRS database table that had a row for every file in the repository.

Analysis was also performed on the DRS users and depositors. Fifty-five different units owned content in the DRS. Switching over to the new DRS meant that these units would need to learn the new concepts, start using the new tools, and potentially change their deposit and management workflows. If the user migration wasn't planned carefully, it could be very disruptive to them. The user analysis included determining the content makeup of each owner account; determining which users performed their own deposits, which made use of the Library's reformatting/deposit services, and which users did both; and surveying the users to gauge their perceived readiness to switch over to the new DRS tools. The users were encouraged to attend training sessions, and their attendance was tracked to ensure that everyone was prepared for the switch. Special attention was paid to the units that owned relatively large amounts of content and that actively managed their content/metadata in the administrative interface to make sure that the plan took any concerns they had about timing or impact into account.

To help communicate the migration plan, a DRS Advisory Board was created to help communicate about the project to Library staff. The members were composed of Library staff, Library administration and the Repository Manager. The Advisory Board members held an open meeting to explain to Library staff the goals of the project, what would be achieved when the project was complete, and upcoming project milestones. In addition, regular updates were posted to a Library staff newsletter and several web pages about the DRS2 project were posted to the Library's web sites.

Based on the content and user analysis, a migration plan was developed that respected all the technical requirements while minimizing the impact on the users. A key requirement for the users was that the amount of time that they needed to work using the old and new DRS tools simultaneously be minimized. The plan called for the migration to happen in "tiers," or phases, in which particular content models were migrated before proceeding to the next tier. Each tier would include the content for all owning units, except for the Still Image and PDS[2] Document tier ("Tier 2"), because the vast majority of DRS content is Still Images or PDS Documents. Tier 2 would be migrated owning unit-by-owning unit to minimize the length of migration time for each owning unit. The owning units were sequenced within Tier 2 based on a combination of their current deposit and management activity, and their level of preparation (training and participation in beta testing).

The migration tiers and associated content types are shown in Table 2. The simplest objects to migrate were done first so that those performing the migrations could gain experience and the migration tool bugs could be discovered before moving to more complex objects. The content was also sequenced this way because of dependencies between content. For example, the Color Profiles and Target Images had to be migrated before the Still Images so that metadata relationships (e.g. HAS_COLOR_PROFILE, HAS_TARGET) could be recorded from Still Images to the Color Profiles and/or Target Images.

Table 2: The Content Models Included in Each Migration Tier

Tier / Sequence	Content Models
1 (all owning units together)	Text (Methodologies, ESRI World Files), Document, Color Profile, Target Image
2 (individual owning units sequenced)	PDS Document, Still Image
3 (all owning units together)	Audio, Text (SMIL Playlists)
4 (all owning units together)	Google Document Container 1, 2, 3; Web Harvest
5 (all owning units together)	Biomedical Image, Opaque Container

The DRS has a development and qa instance as well as a production instance. In the early stages of the migration, it was thought that the migration could be tested in the development and qa instances but it was found that these instances did not have enough "real" content in them to be useful for the testing. To address this issue, a copy or "clone" of the production DRS1 database was set up. This turned out to be invaluable for testing the migration code.

There were particular types of content in the DRS that were no longer needed anymore and therefore close to 325,600 files, or 0.6%, were not included in the migration. These "left behind" files can be broken into two different groups – the legacy METS descriptors that were replaced by new METS descriptors generated during the migration and the other files that are no longer needed anymore for delivery (obsolete audio files and playlists that were replaced during the migration, index files and MOA2 [6] files that had been superseded years before).

It was important that delivery to end users, from access copies in the DRS, not be interrupted by the migration. This was achieved by a combination of strategies. One was to retain the same unique IDs for the files in the new DRS database as they had in the old one (which was based on a numeric sequence). Retaining these same IDs meant that the persistent names (URNS), which resolved to URIs containing the IDs, would still resolve correctly without any interventions. In addition, the DRS delivery applications were made "migration-aware." They were enhanced to be able to locate the metadata in either the old or new DRS – choosing the metadata from the new DRS if the file was migrated.

2.4 Metadata Migration

A 1.0 FTE developer was hired to create the migration tools and to run the migration. The migration tools (or "migrator") were designed to be modular and parallelizable. The migrator had three main modules, as shown in Figure 1. The first module (the "selector") identified the files that should belong to the same object (e.g. the archival, production and deliverable images, in the case of a still image object). The second module (the "descriptor generator") aggregated the metadata for the files and the object itself from various places and generated a METS metadata descriptor for the object. The third module (the "ingestor") ingested the object into the new DRS. Each of the modules was designed to work independently and in parallel. Oracle queues [5] were used by the migration modules to know when there were objects ready for their stage of the migration. Each module could be run in multiple threads, and tuning experiments were run with various thread counts to find the ideal number of threads per module.

A database field, MIGRATION_STATUS, was used to track where each file was in the migration. Various typed "administrative flags" were recorded in the database when the migrator encountered potential irregularities either in the metadata or in the content itself that should be examined by preservation staff in the future.

The technical metadata was produced by parsing each file with the Harvard Library-created FITS tool. During the tuning experiments, it was discovered that running FITS on each file was the single most time-intensive activity of the migration so the FITS output was pre-generated as a separate process. Descriptive metadata for the object was imported from Harvard Library's LibraryCloud, a metadata hub providing access to bibliographic metadata aggregated from several Harvard data sources, including the central catalog, an image catalog and finding aids.

The existing DRS ingest code was used as a base for the ingestor to ingest the new METS files, adding the relatively minor changes needed by the migration (e.g. adding an event for the migration to the provenance metadata). The ingestor copied the descriptor file to DRS storage, created DRS2 database records for the new object and its files, and then set the migration status of the files in the old DRS database to a value that meant the ingest was successful.

[2] PDS is a Harvard Library application for delivering page-turned objects. Where this paper refers to PDS Documents, it means objects composed of one or more page images and optionally page text files, such as a scanned book or manuscript.

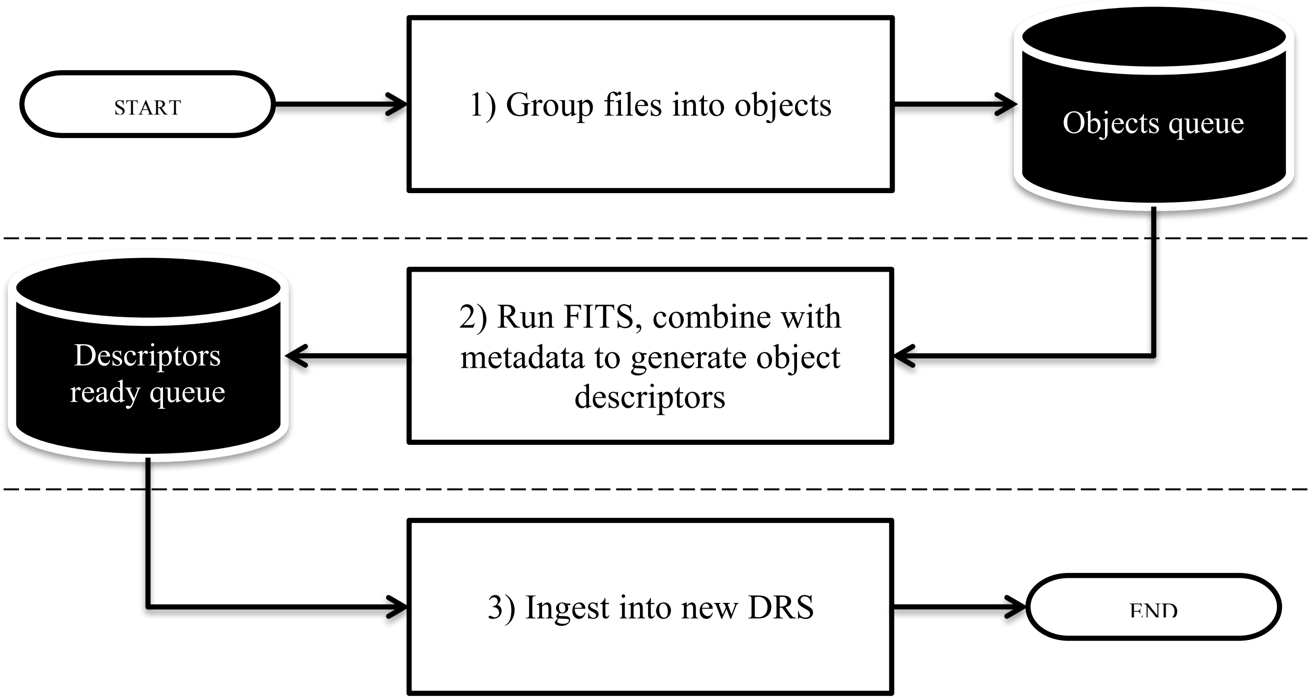
START
1) Group files into objects
Objects queue
2) Run FITS, combine with metadata to generate object descriptors
Descriptors ready queue
3) Ingest into new DRS
END

Figure 1: The Three Modules of the Migrator: 1) Selector, 2) Descriptor Generator, 3) Ingestor

A migration checklist was developed by the repository manager to use for each migration. Each checklist described the set that would be migrated, listed the sequence of steps to be performed by the responsible parties, summarized any problems found, and provided a place for the responsible party to put a timestamp next to each completed task. The checklist covered tasks related to internal and external communication, pre-migration analysis, official kickoff, migration steps, post-migration verification and cleanup.

An integral part of the migration design was the expectation that some things would fail, because of errors in the metadata or content, bugs in the migration code, or not fully understanding the metadata or content when the migrator specifications were written. All of these cases were encountered many times throughout the migration, described in more detail in the Results section of this paper. An "expunge" method was developed to be able to undo a migration for any given set of files or objects. A spreadsheet of files and objects to expunge was maintained by the repository manager doing the migration validation, and used by a senior developer to perform the expunges. The number of files needing expunging was so large that the expunge function had to be scaled up at one point.

2.5 Audio Format Migration

The audio format migration is still underway at the time of composition of this paper, due to be finished by June 2018, and its process merits a longer and more in-depth description than outlined here. Because of space constraints, a summary of it is presented to give further context to the expanse of the migration project as a whole, and the details will be covered in another form at a later date.

In the old DRS, access copies of audio content were comprised of RealAudio files with SMIL files to delineate playlists. Harvard Library's streaming delivery service used the Helix server to stream RealAudio files to RealPlayer clients on users' desktops. These formats and the RealPlayer application has been declining in standard use for many years, and as this content became increasingly difficult for users to access, the need for an audio format migration became evident. In 2015 Harvard Library's Digital Preservation Services, Media Preservation Services (MPS), and Library Technology Services (LTS) teams began collaborating on a plan to change the delivery format and platform, while maintaining the playlist functionality. The decision was made for the following format migrations to be executed:

- RealAudio files would be replaced with the more commonly used MP3 format by creating new deliverables from the production master Broadcast WAVE files
- SMIL files would be migrated to the AES60 standard for audio metadata [1]

The old DRS held 6,994 audio objects deposited by 12 different units across Harvard University. Audio objects can be comprised of archival master files, production master files, and/or access files. The archival and production master files were stable and not in need of migration. The total number of access files in the old DRS that required format migration was 12,242 RealAudio files and 56,400 SMIL files. To start the process, LTS provided a copy of the original audio METS file along with all of the audio and SMIL files to MPS. MPS extracted metadata from the audio METS and created new deliverable files in MP3 format based on Audio Decision Lists (extracted from audio METS), originally used for creation of the RealAudio delivery files. Additionally, MPS transformed SMIL playlists to AES-60 format using XSLT stylesheets. Then the new Submission Information Package (SIP) was deposited into the DRS. During this process, MPS ran error checking tools, and Digital Preservation and LTS performed quality control measures, such as validating that URNs resolved correctly to the new Streaming Delivery Service that had been created as part of the project.

The structure and metadata of audio objects deposited to the DRS had evolved over time. Eventually the audio objects began to be deposited in a standardized form, but especially in the first couple of years of depositing audio to the DRS, the objects were different enough that the migrator would need to expect that some objects would need to be treated differently than the rest. The basic strategy was to migrate the audio objects in tiers of complexity (easiest and automatable to hardest and manual). For simplicity, tier 1 included all the audio objects in an automated migration process designed to handle the vast majority of the audio objects in the DRS, with the expectation that some would fail due to anomalies. Those that failed would be moved into tier 2, to be addressed in a more hands-on manner, then tier 3 if necessary, and so on.

3 CONCLUSION

3.1 Metadata Migration Results

At the end of the project, the metadata had been migrated for almost 52 million files as shown in Table 3. A little over 81% of the files were migrated to PDS Document objects. Almost 17% of the files were migrated to Still Image objects. The remaining less than 2% of the files were migrated to the remaining content models – Google Document Container 2 (1.6%), Web Harvest (0.09%), Audio (0.08%), Document (0.07%), Google Document Container 3 (0.06%), Opaque Container (0.04%), Text (0.009%), Biomedical Image (0.007%) and Google Document Container 1 (0.002%). Because the migration aggregated the files into objects, for the first time object-level statistics could be run:

- Number of objects migrated: 6,545,514
- Average number of files per object: 7.9

In the old DRS, the file format was not automatically detected – it was asserted by DRS depositors. The depositors could only choose from a short list of formats which were at a very coarse level (e.g. TIFF or PDF). During the migration, the format of each file was re-characterized by the FITS tool. Using FITS, we were able to get much more specific about the formats. For example, in the old DRS, all PDF documents of any flavor were simply identified as being in the "PDF" format. In the new DRS, these documents were identified as being in one of thirteen flavors of PDF, e.g. PDF/A 1a:2003.

3.2 Data Errors Identified and Fixed

One of the largest benefits of the DRS metadata migration is that it revealed many problems with the content and/or metadata that we were able to clean up in most cases. During the migration, 47 different types of content or metadata problems were encountered, affecting 248,127 objects or 4% of the 6,545,514 objects migrated. Some of these errors were not detected prior to the migration because the errors could only be detected through deep analysis of the metadata relationships between objects, as was done as part of the metadata analysis. An example of this is the 89 PDS Document objects that were found to be merged into themselves. Some of these errors were only detected because the FITS tool parsed every file. For PDF documents, FITS was able to detect when the PDS documents were encrypted. There were 7,707 files in three flavors of PDF that were identified as having some form of encryption. 99% of these had been deposited by a single owner code, who generally uses an external vendor for digitization services. All of the encrypted PDF Documents examined used Password Security for the Security Method, and 128-bit RCA for the encryption level, to restrict what could be done with the document. Other problems were only detected during the verification phase after a migration. An example of this is a malformed persistent name (URN) that prevented users from accessing a PDF Document before this was fixed.

The data errors that were found were analyzed (as shown in Table 4) so that we could learn from them and potentially prevent them from occurring in the future. The data errors encountered during the migration were associated with objects in five different content models: Audio, Document, PDS Document, Still Image and Target Image. There were more different kinds of errors found for PDS Documents (30 error types) than other object types, in part

Table 3: Summary of the DRS Files Migrated, Not Migrated, and Replaced

Disposition During Migration	Description	File Count	% of Files
Migrated	All archival master and production master content files; all content files used for delivery; all auxiliary files containing documentation, provenance or process history information, or that are used to support delivery (e.g. world files)	51,726,150	99.4%
Not migrated	Obsolete delivery files (e.g. RealAudio files, SMIL playlists); files no longer needed to support search or display (e.g. index files, MOA2 files)	158,117	0.3%
Replaced	Legacy DRS METS descriptors that were replaced with new METS descriptors during the migration (e.g. METS for PDS documents or for Web harvests)	167,441	0.3%
		52,051,708	100%

Table 4: Key Differences Between Content Models and Errors Found

Content Model	Manual deposit?	Multi-file object?	Tools for structural manipulation after deposit?	Types of Errors Detected	Number of objects with detected errors
Audio	YES	YES	NO	Deposit errors	847
Biomedical Image	YES	YES	NO	None	0
Color Profile	YES	NO	NO	None	0
Document	YES	NO	NO	Deposit errors	7,707
Google Document Container 1, 2, 3	NO	NO	NO	None	0
Opaque Container	YES	NO	NO	None	0
PDS Document	YES	YES	YES	Deposit & Management after deposit errors	1,299
Still Image	YES	YES	NO	Deposit errors	238,083
Target Image	YES	YES	NO	Deposit errors	191
Text	YES	NO	NO	None	0
Web Harvest	NO	YES	NO	None	0

because these objects were more complex than the others. However, the errors affecting Still Image objects were much more common than for other object types, affecting 234,000 Still Image objects. Another way to analyze these data errors is to classify them according to when the error was introduced – either at the time of deposit, or during management of the content/metadata sometime after deposit. It was determined that 69% of the objects with identified errors already had these errors at the time of deposit; the other 31% of the objects had errors introduced as a result of managing the files (changing content or metadata) after they were already in the repository.

There were not any data errors detected for objects in the following content models: Biomedical Image; Color Profile; Google Document Container 1, 2 and 3; Opaque Container; Text; and Web Harvest. In each case where the deposits are automated from another system or tool (i.e. the manual intervention is removed), there were no errors detected in those objects. Another factor that appears to play a key role is how simple the object is. With complex objects, the structural and relationship metadata must be correct in the first place, and maintained over time. When the repository provides tools to manipulate that structure after objects are already in the DRS (as in the case of PDS Documents), that puts a further strain on maintaining the integrity of the object over time.

3.3 Challenges Encountered

The challenges encountered during the project were both technical and organizational. The DRS contains many different object formats, each with different structures, requiring unique migration rules per content model. There are also tens of millions of files in the DRS, each that had to be parsed by FITS to re-characterize their formats and technical characteristics. Some of the largest challenges we faced are discussed here.

3.3.1 Preserving database identifiers

The DRS is an access as well as a preservation repository. The deliverable files all have URNs that resolve to URLs containing the Oracle file IDs for the deliverable files. To maintain user access to these files during and after the migration, the database identifiers for each file in the new database needed to be kept the same as they had been in the old database. An added complexity was that new files were also being deposited to the new DRS and we had to make sure that there was no overlap between the file IDs that would be reused and the new file IDs being minted with new deposits. The way this was done was by leaving a gap in the file ID sequence so that the IDs of newly deposited files and objects would start at 400000000, reserving the IDs less than this for the migrated files.

3.3.2 Custom code

A few of the DRS owning units had deposited content in the past without the recommended derivative metadata, requiring custom code to be written for their content. For one of these units it took six attempts before we were able to successfully migrate their Still Image and PDS Document content because the idiosyncrasies of their content/metadata weren't fully understood by anyone when the specification for the custom migration code was written.

3.3.3 Large and multi-volume works

One of the functions supported in the old DRS was merging of PDS Documents together. This was often done for multi-volume works or series in which parts were digitized over time and combined into a single PDS Document so that users could navigate and search the work/series from one interface. Over time it was determined that the merge practice was problematic, resulting in overly large objects that were difficult to maintain in the repository, difficult to deliver to users, and were prone to having structural and other errors because of all the manipulation post-deposit over time. For these reasons, the merge function was not implemented in the new DRS. Instead a hierarchical content model, the PDS Document List, was designed to permit large and multi-volume objects to be broken into smaller objects that could be maintained separately and combined for delivery. Over the course of the migration, 419 PDS Document List objects were created for 18 different owning units. The specification, migration, and validation required for these objects was much more complex than regular PDS Documents and took many staff hours to complete. In addition, the list of objects to become PDS Document List objects had to be vetted with representatives from each owning unit in advance.

3.3.4 Long-running project

This was the longest-running digital preservation project ever conducted for the repository. The Library did not have any experience with very large digital preservation projects, long enough to span a Library reorganization, and to be affected by changes in staff, administration and workflows. Attention had to be spent periodically on answering the "why is it taking so long?" question, and reminding stakeholders of the benefits we would eventually accrue. Also, the development team changed to using an agile development process mid-project, which took additional time to adjust to, especially in the learning stages when there was a great deal of tension between deadlines and functional iterations. To complete the project, we needed to be flexible and opportunistic, changing plans to adjust to organizational needs. For example, we reordered the planned sequence of content to migrate several times, in response to Library billing requirements, and to minimize disruptions to the reformatting labs.

3.4 Recommendations

There is much that can be learned from this project overall, but especially from the metadata migration, in terms of ongoing preservation planning and interventions, informing future migrations, and in avoiding/detecting/recovering from metadata and content errors.

3.4.1 Preservation planning and interventions

This migration revealed many errors in metadata and/or content that were not known to anyone beforehand. The errors were found by repository staff through a combination of techniques – deep analysis of the metadata, parsing of files with automated tools and scripts, and verification routines after each migration. Particular problems (lack of clarity on the file format, encryption, etc.) were flagged through metadata elements introduced as part of this project. As time permits, over time the repository metadata and content should be analyzed to identify additional errors, and to explore further those already found.

3.4.2 Future migrations

There were lessons learned during this metadata migration that are applicable not only to future migrations, but for future large digital preservation projects in general. A few of these lessons are technical in nature:

- Aim for success but design for error. Migrations in general are relatively new to the digital preservation community. Experience and tools are still being developed from the ground up. It is to be expected that mistakes will be made especially when the metadata and/or content is not fully understood. The ability to do-over any action that could have a detrimental effect on the content/metadata under preservation should be built into the design. This was done in the DRS project by implementing a way to undo a migration for any set of files after errors were found, "expunge"; implementing a way to track the migration status of files and any errors that occurred; having documented steps to verify successful migrations and checklists to record this verification.
- Not all repository changes are equal – some have a much larger impact on the architecture, workflows, and tools. In the case of the DRS2 metadata migration, the changes to the data model, metadata schemas and AIP packaging caused a ripple effect, requiring almost everything else to change. Large changes like this may be warranted but the impact should be fully realized by all stakeholders.
- Other lessons learned were more organizational in nature:
- Very large projects need deep organizational buy-in by all the key stakeholders – higher administration who control the budgets and priorities, repository staff doing the development over a very long period of time (restricting their ability to work on other projects), and anyone who's workflows/tools/knowledge must adapt to the changes (e.g. curators, reformatting specialists, etc.). This buy-in must be periodically "refreshed" to remind everyone of the purpose and benefits of the project.
- Be willing to redefine the project's completeness criteria to cope with an organization's declining interest in the

project. Periodically revisit what has to be included as part of this project and what can be delayed for a future project or not done at all. In the case of the DRS2 metadata migration project, the curatorial functions in the new Web Admin were prioritized over the preservation planning activities, because those could be done at a later time and it was more important that the curators see immediate benefits of the project.

- Communication about the project is critical for large projects like this because people will make their own (often negative) assumptions if they don't know the project status.
- Unless the roles and responsibilities of people involved in the project are very clear, this will be a distraction and will negatively affect project meetings, project workflows and project reporting.

3.4.3 Preventing metadata and content errors

Many metadata and content errors were uncovered by this migration project. Some of the lessons learned include:

- In the context of the DRS, it would be more beneficial to concentrate resources on preventing errors introduced during deposit as opposed to errors introduced after deposit during metadata and content management.
- The more that deposits can be automated by tools from systems, the less likely we will see errors.
- More broadly, minimizing manual interventions during deposit can lead to less errors and more accurate/richer metadata, by using automated tools like FITS or by pulling from authoritative sources such as LibraryCloud.
- More effort should be spent on enhancing repository tools to validate content and metadata, especially for errors seen frequently.

4 ACKNOWLEDGEMENTS

As can be expected of a project of this size and duration, many people contributed to it. The authors would like to acknowledge the following individuals (in alphabetical order): Bobbi Fox, Franziska Frey, Wendy Gogel, Chip Goines, Jonathan Kennedy, Spencer McEwen, Grainne Reilly, Tracey Robinson, Randy Stern, Janet Taylor, Chris Vicary, Robin Wendler, Julie Wetherill and Vitaly Zakuta. Individuals in the following groups also contributed to the work: the DRS Advisory Group, DRS beta testers, the DCSWG, HUIT Security, Imaging Services, LTS Operations and Media Preservation Services.

5 REFERENCES

[1] AES60-2011: AES standard for audio metadata - Core audio metadata. Audio Engineering Society. Retreived from http://www.aes.org/publications/standards/search.cfm?docID=85.

[2] Documentation and Official Code Releases of the FITS and the FITS Web Service Projects. File Information Tool Set (FITS). Retrieved from https://projects.iq.harvard.edu/fits/home.

[3] Electronic Archiving System (EAS). Library Technology Services, Harvard Library. Retrieved from https://library.harvard.edu/lts/systems/eas.

[4] Harvard LibraryCloud and PRESTO APIs. Harvard Library. Retrieved from https://library.harvard.edu/librarycloud.

[5] Introduction to Oracle Advanced Queueing. Oracle9i Application Developer's Guide. Retrieved from https://docs.oracle.com/cd/B10501_01/appdev.920/a96587/qintro.htm.

[6] MOA2 Digital Object Document Type Definition Tutorial. CDL Digital Object Document Type Definition Tutorial. Retrieved from http://xml.coverpages.org/moa2-dtdtutorial2.html.

[7] Priscilla Caplan. 2009. *Understanding PREMIS*. The Library of Congress. (February 2009). Retrieved from https://www.loc.gov/standards/premis/understanding-premis.pdf.