Migration at Scale: A Case Study

Sheila M. Morrissey
ITHAKA
100 Campus Drive, Suite 100
Princeton NJ 08540 USA
1-609-986-2221
sheila.morrissey@ithaka.org

Vinay Cheruku
ITHAKA
100 Campus Drive, Suite 100
Princeton NJ 08540 USA
1-609-986-2232
vinay.cheruku@ithaka.org

John Meyer
ITHAKA
100 Campus Drive, Suite 100
Princeton NJ 08540 USA
1-609-986-2220
john.meyer@ithaka.org

Matthew Stoeffler
ITHAKA
301 East Liberty, Suite 250
Ann Arbor MI 48104 USA
1-734-887-7079
matthew.stoeffler@ithaka.org

William J. Howard
ITHAKA
100 Campus Drive, Suite 100
Princeton NJ 08540 USA
1-609-986-2217
william.howard@ithaka.org

Suresh Kadirvel
ITHAKA
100 Campus Drive, Suite 100
Princeton NJ 08540 USA
1-609-986-2273
suresh.kadirvel@ithaka.org

ABSTRACT

Increasing experience in developing and maintaining large repositories of digital objects suggests that changes in the large-scale infrastructure of archives, their capabilities, and their communities of use, will themselves necessitate the ability to manage, manipulate, move, and migrate content at very large scales.

Migration at scale of digital assets, whether those assets are deposited with the archive, or are created as preservation system artifacts by the archive, and whether migration is employed as a strategy for managing the risk of format obsolescence, for repository management, or for other reasons, is a challenge facing many large-scale digital archives and repositories.

technical, events, and structural metadata for approximately 15 million e-journal articles in its archive. It describes the migration purpose, planning, technical challenges, and quality assurance demands associated with digital object migration at very large scales.

1.2 Portico Preservation Workflow and Metadata

Portico is a digital preservation service for electronic journals, books, and other content. Portico is a service of ITHAKA, a not-for-profit organization dedicated to helping the academic community use digital technologies to preserve the scholarly record and to advance research and teaching in sustainable ways. As of May 2012, Portico is preserving more than 19.4 million journal articles, e-books, and other items from digitized historical collections (for example digitized newspapers of the 18th century).

Content comes to Portico in approximately 300 different XML and SGML vocabularies. These XML and SGML documents are accompanied by page image (PDF, TIF, and JPG) and other supporting files such as still and moving images, spreadsheets, audio files, and others. Typically content providers do not have any sort of manifest or other explicit description of how files are related (which ones make up an article, an issue of a journal, a chapter of a book). This content is batched and fed into a Java (ConPrep) system, for assembly into what the Open Archival Information System (OAIS (([3].

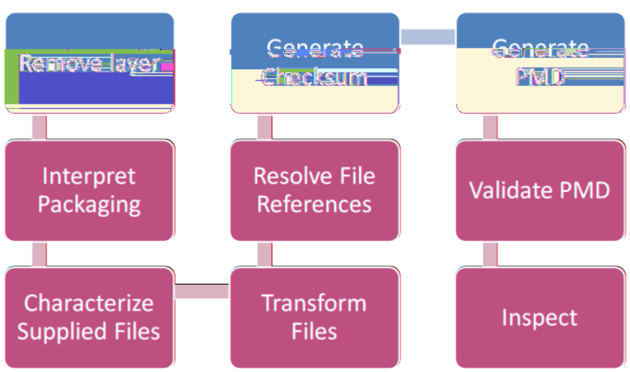

Figure 1 Portico ConPrep High-Level Workflow

The ConPrep workflow maps the publisher-provided miscellany of files into bundles that comprise a single article or book or other content item, (((((where a format specification and validation tool is available, validates each file against its format specification. Publisher-provided XML and SGML journal article files are normalized to the Portico profile of the National Library of M (Archiving and Interchange Tag Set; e-book files are normalized to So ConPrep, which ((((itself an instance of migration at scale, of both the (implicit) package format and of files within the package, at the point of receipt of content.

Some of the steps in this workflow are automated quality-assurance checks of the XML content both the content provided by the publishers, and artifacts produced by Portico in the workflow itself. This QA includes validation against XML and SGML document type definitions (DTDs) and schemas. It also includes the assertion, via Schematron (a rule-based validation language for making assertions about the presence or absence information in XML files [7]) of other constraints on content values. Additionally, the workflow includes visual inspection of sample content.

ConPrep generates preservation metadata for each AU. Modeled on PREMIS [10] and METS [4], the generated information includes descriptive, or bibliographic, metadata; structural metadata specifying the relationships among the components of the archival unit, technical metadata about files and their formats; provenance and event metadata, detailing the tool chain, including hardware and software information, used in processing the ((right to preserve these digital objects. These metadata are instantiated as XML, and are stored with the preserved digital object. Just like the publisher-provided XML files, the preservation metadata is schema-validated, and then further validated via Schematron.

2. THE MIGRATION

2.1 Motivation

2.1.1 Archive Life Cycle: Continual Review and Revision

As with its preservation policies, practices, and procedures, ((including its hardware, software, and key data and metadata structures has been subject since inception to a continual process of review and revision. This review and revision is intended to incorporate both lessons learned from our own experience with content that has steadily expanded both in volume and in type, and with the continually developing understanding of best preservation practice in the larger preservation community.

The first major refinement of the original Portico platform was undertaken to scale up the capacity of the ConPrep system from 75,000 e-journal articles (and approximately 750,000 files) per month (900,000 articles/9,000,000 files per year) to 10 million articles and 100 million files per year an order of magnitude increase. The system was in fact increased to a capacity of 24 million articles and 240 million files per year, operating at 50-75% of peak capacity. [11]

2.1.2 New Requirements, New Knowledge: New Content Model

As the Portico archive was extended to handle new content types beyond electronic journal content, its content model and the Portico metadata (PMETS) schema (which had key conceptual dependencies on that content model), were subjected to review and revision. The PMETS schema, whose design was based on METS 1.4, and informed by early work on the then-uncompleted

PREMIS data dictionary, had undergone 6 minor, backwardly compatible revisions (typically to accommodate changes to subsidiary schemas which specified descriptive and events metadata) since it was designed and implemented in 2002-2003.

By late 2008, the review process indicated the data model underlying the PMETS schema would be stressed by new requirements for the Portico archive. These included

- new content types (such as books and digitized collections), with richer and more complex relationships among the components comprising a single digital object
- new preservation activities, such as versioning, the creation of access artifacts, and the export of metadata in standard formats
- extended use cases in the ConPrep system, including the ability to assign preservation level by business policy rather than only by file format validity; to de-duplicate content in the archive; to process externally updated content (new versions of all or part of a content unit) as well as internally updated content (such as new technical metadata generated by newly available tools); (

  another image file); to record and mange migration and re-migration of content

The main components of the Portico content model (both the old and new versions) are:

- Content Type (CT) This allows Portico to group content belonging to specific preservation services (
- Content Set (CS) This allows Portico to group together archival units that belong together. For example, all archival units for a single journal of a particular publisher will be placed together within a single content set.
- Archival Unit (AU) The main digital object or abstract intellectual object that is being archived. For example an E-Journal Article.
- Content Unit (CU) A complete version of the content for an AU. In most cases, an AU will only contain a single CU.
- Functional Unit (FU) A container for grouping together components that serve the same function within a content unit. For example, the high-resolution, web ready and thumbnail versions of an image for a single equation or chemical formula would be grouped together in a single FU.
- Storage Unit (SU) A container for all the information on a physical file making up a component of an FU.

In the original content model (see Figure 2), the distinction between an Archival Unit and Content Unit was not well articulated. As implemented, the ConPrep system generated Content Units, which could be understood as a logical unit of content made up of one or more content files and a metadata file that captures all the relevant preservation metadata. As these Content Units were ingested into the Archive, they were renamed (

In the new content model, we refined the concepts as follows:

- Archival Unit: the abstract intellectual object
- Content Unit: a particular version (original, revision, update etc.) of the content

In effect, the presence of multiple content units within an archival unit means that the content has been sent to the archive in multiple versions by the content provider.

These versions can represent changes to the intellectual content, or technical changes such as repair of damaged files or migration to new formats by the provider. This kind of versioning is not under the control of, or initiated by, the archive, and requires maximum flexibility about the granularity and purpose (intellectual content, technical repair) of the change. In such a scenario, all versions (CUs) of an archival unit (AU) are preserved. Each version is represented by a different Content Unit, as shown in Figure 3:

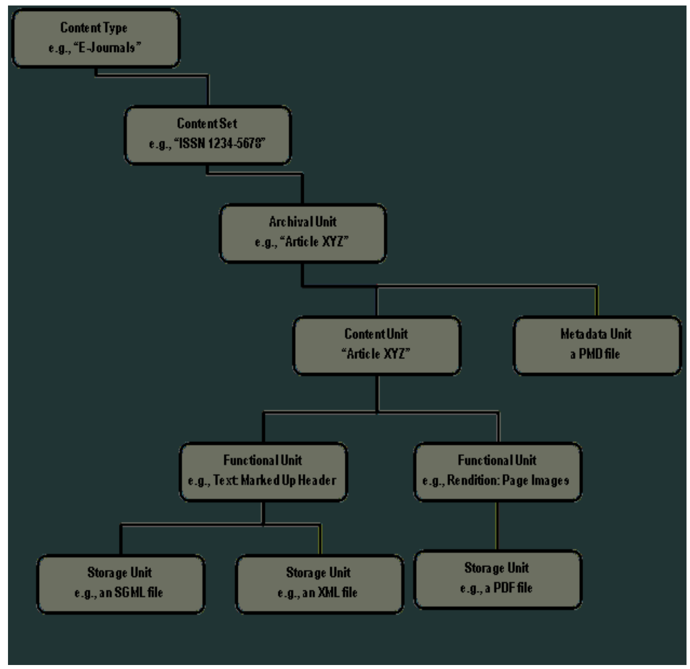

Figure 2 Portico PMETS 1.x Content Model

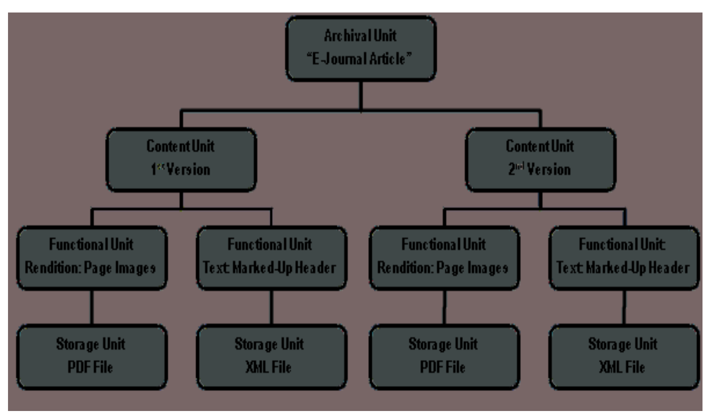

Figure 3 PMD 2.0 Content Model: Archival Unit with 2 Versions of Content Unit

In the content model, we can describe groups of Storage Units (SUs) that are "intellectually" identical but "technically" different by grouping the SUs together in one Functional unit (FU). We can use this grouping both to capture "use" information (see Figure 4), and to indicate migrated content (see Figure 5).

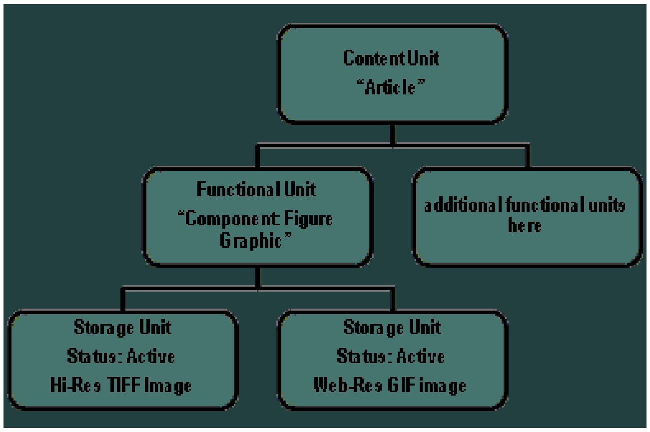

Figure 4 PMD 2.0 Content Model: Multiple Storage Units for Multiple Uses in Same Content Unit

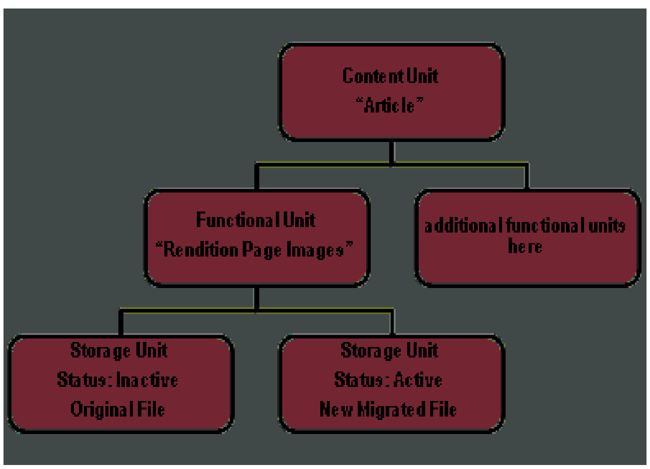

Figure 5 PMD 2.0 Content Model: Content Versioned Within Single Content Unit

Finally, in the new content model, we have extended this concept of grouping with two new components: the Storage Unit Set and the Storage Unit Pointer. These components allow us to describe, in a fairly compressed way, two new kinds of structural relationships: objects that simultaneously belong in more than one group, and relationships between sets of objects. Both are illustrated in Figure 6 below. In this example, a digitized book, each page image exists in multiple resolutions (the dotted arrows) and the entire set of high-res page images has been converted into a single PDF file (the curved red arrow). These new relationships can also be used to describe an XML text that consists of multiple files (e.g., chapters of a book).

2.1.3 Goals and Context

The goals of the new preservation metadata project were to

- Support new requirements and processes described in the previous section
- Incorporate the latest thinking from the preservation community, including from the now mature PREMIS model

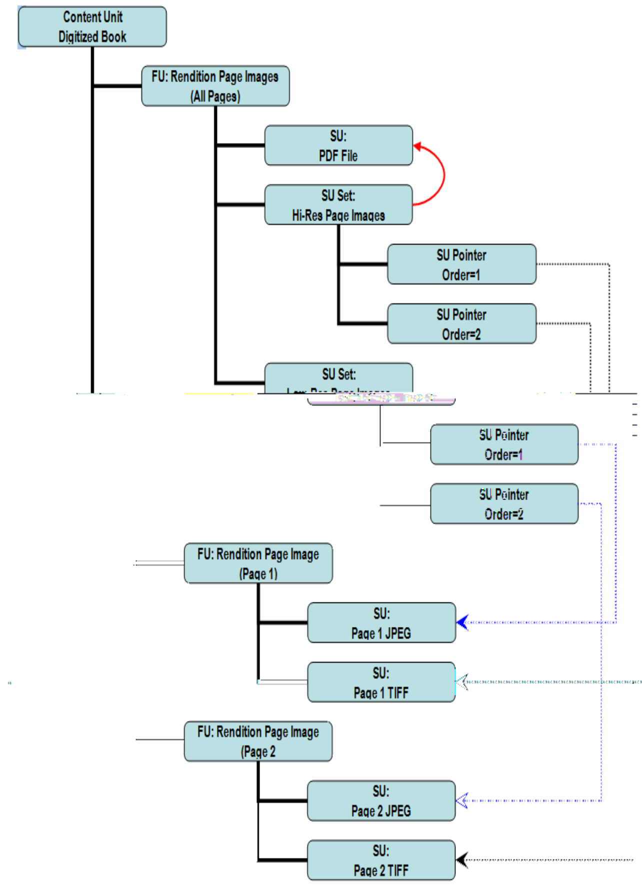

Figure 6 PMD 2.0 Content Model: Complex Component Relationships

- Develop a well-documented design for the new content model, and implement that design cleanly and consistently across all our applications. Design goals included [12]
    - Making explicit all data constraints not currently explicitly expressed in our schemas
    - Eliminating redundant information where possible
    - Establishing a clean base line for future expansion of events metadata
    - Clarifying what event goes with which object and why
    - Employing consistent editorial/coding practices (capitalization, verb tenses, etc.)

The project was undertaken as the archive continued its normal processes, including on-going incremental changes to the ConPrep system itself (deployment of new tools, facilities, etc.). It was undertaken as well in the context of a major institutional transition, as Portico, which had originally moved from a proof-

of-concept project of JSTOR to a free- (
of the newly created Ithaka Harbors, in 2003, became an integrated service, along with JSTOR and Ithaka Strategy and Research, of the newly created ITHAKA, in 2009.

An additional consideration is the key role that preservation metadata plays in the archive (

2.2.2 *Requirements and Design: Events Review*

A key component of PMETS 1.x and its underlying data model was the Portico event model. When the migration project was initiated, approximately one billion events had already been recorded in the processing of the approximately 15 million archival units and their 150 million component files. These events were associated with items in the PMETS file at both the CU and the SU level.

The event model was instantiated in the Portico Events schema. It was primarily modifications to (i.e. new versions of) the Events schema that necessitated new versions of the PMETS schema. These modifications were made incrementally, as new use cases were created by new workflow steps or other changes to the system. The event schemas defined each event separately, with different attributes and sub-elements for each event. A new design would simplify the existing data structures into a generic event that is typed with properties not specified in the schema itself, thereby allowing extensions without new versions. This in turn would obviate the need for regenerating the corresponding JAXB classes for marshalling and unmarshalling files in ConPrep.

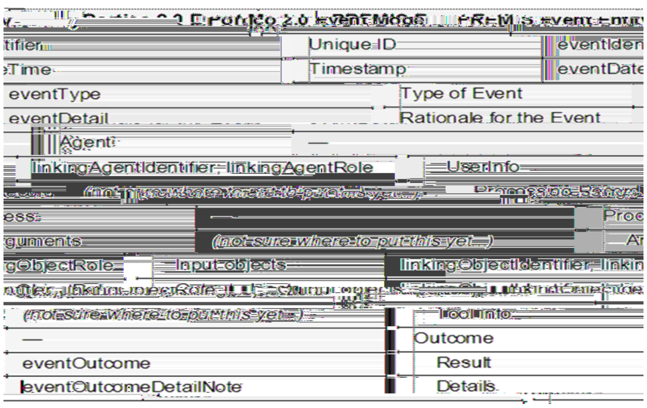

Figure 8 Mapping New Event Model to PREMIS

We reviewed each version of the Events schema, developing tables indicating, for each activity in the ConPrep workflow, what events could result, and the element and attribute values assigned by the system. Informed by the analysis of key components of the PREMIS event model (see Figure 8), we abstracted out simple event types that describe the event itself. Those basic event types would then be qualified or sub-classed by assigning values the *Rationale* attribute. The controlled list of those values, however, would not be defined in the schema, thus allowing for extension without a new version of the schema.

2.2.3 *Information Architecture*

The data model having been constructed, the next steps were to review the ConPrep and archive server management Java code, and the relational database used to store and manage data object and event information during the ConPrep workflow, to determine what changes would be required to employ the new data model, and create and manage instances of the new PMD 2.0 XML format for preservation metadata. Changes included:

- New relational schema for the relational database, conforming to the new information model (see Figure 9)

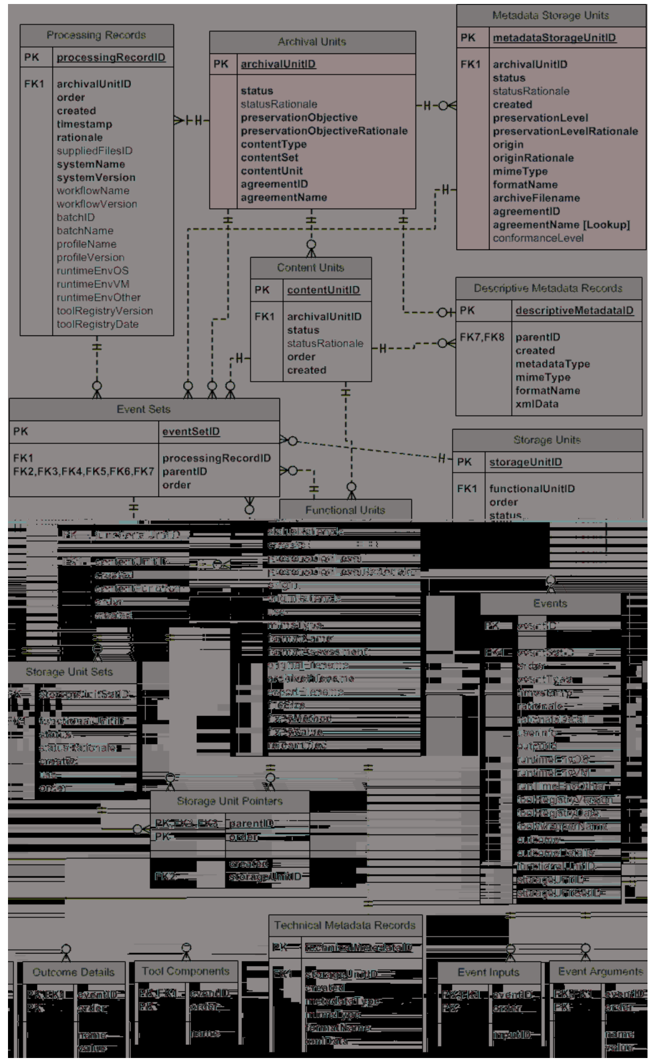

Figure 9 New Relational Schema

- New code to create, read, and write PMD 2.0 files
- New workflow step to create AU-level, Dublin Core descriptive metadata that could be employed across all content types (each CU would have content-type appropriated descriptive metadata as well)
- New code for creating instances of the new event types, with the appropriate new attribute values in managed lists, including new validator code at event creation time
- New code for the Portico delivery and audit sites for handling the new metadata files
- New tool wrapper code to employ new streamlined schema for preserving tool information
- New code for the ConPrep GUI for viewing new metadata formats, and to adapt user-defined reports to the new AU/CU hierarchy

- New Schematron validator for the new PMD 2.0 format, to enforce, among other things, controlled lists of values for event attributes
- New archive server management code to handle new PMD 2.0 format

There were other tasks associated with performing the actual migration and validation of existing PMETS files.

The first task was to create a detailed information map of the elements and attributes in the new schema (see Figure 10). This map provided a definition of the meaning of each element or attribute; its data type and constraints on values, with an indicator as to whether the constraint was to be enforced by the schema or by the Schematron validator; and its place in the relational database, in the new schema, and the corresponding element or attribute, if one existed, in the PMETS file to be migrated.

Name	Definition	Data Type and Constraints	Oracle Implementation	PMD 2.0 Implementation	PMETS 1.x Implementation
xmlSchemaVersion	major + minor version of PMD when exported as XML	✓ fixed value of "2.0" for now; will increment in future	–	PMD / @xmlSchemaVersion	–
conformanceVersion	Name of file used to verify conformance (i.e., Schematron) of this file to enumeration specifications.	String.	Metadata Storage Units . conformanceLevel	PMD / @conformanceVersion	–
objID	Unique ID for this XML metadata file; will be the same ID as the active Metadata Storage Unit	ARK	Metadata Storage Units . metadataStorageUnitID	PMD / @objID	/ PorticoMETS / @objID
created	Date/time this record [illegible] Storage Unit	✓ Timestamp	Metadata Storage Units . [illegible]	PMD / @created	/ PorticoMets / metsHdr / @CREATEDATE
ARCHIVAL UNIT					
archivalUnit	The basic unit of archived content.	✓ Children: 1..n Processing Record 1..n Event Set 1..1 Descriptive Metadata 1..n Content Unit 1..n Metadata Storage	Archival Units	PMD / ArchivalUnit /	/ PorticoMets

Figure 10 Information Mapping

The next task was to develop the transformation and validation pipeline for the existing 15 million PMETS files. This entailed

- Extracting a copy of the files from the Portico archive
- Developing an XSL transformation from PMETS to PMD, using the information mapping table
- Developing the Schematron assertions to test the data types and constraints in the information mapping table (this is the same Schematron that would be used in ConPrep, going forward, to validate new PMD files)

The pipeline was to be run via an application called ConprepLite is a light-weight façade over the Conprep workflow and tool wrapper classes. It was devised to enable the Portico Data Team to test their transformation and validation tools against thousands of files, while using the same code invoked by the ConPrep runtime to run those tools. Because we were scaling up the use of ConprepLite from thousands to millions of files, it was also necessary to refactor the ConprepLite software to be multithreaded, and to streamline the reading and processing of the XML configuration files (which listed input files, and the workflow steps to be executed) from a document object model to a streaming model.

Finally, we would require one set of scripts to extract samples of the newly created AU-level descriptive metadata, for review and approval by the Portico Archive Service Product Manager, and another set of scripts to import the new PMD 2.0 files into the archive, and update the archive management database to reflect the presence of these new assets, and their relationship to the existing content and metadata files.

2.3 Execution

2.3.1 Technical Challenges

One of the lessons learned from scaling up the ConPrep system ([11]. That expectation was amply met when we revved up the pipeline. We found that processing such a large number of (often very) large XML files stressed both hardware and almost every layer of software in the pipeline stack.

Tuning of all sorts was an issue. With multiple threads running on multiple machines, it took some tuning to settle on reasonable batch sizes, so that any failure of a single batch would not result in the waste of days or even weeks of run time. It took some trials to determine the optimum thread count to employ on each instance of ConprepLite that was running on multiple, and different, hardware and operating systems configurations.

Both the PMETS files and the XSL files designed to transform them were quite large and complex (the transform files run to rather than the kernel. The ConprepLite instances were then moved to heavier-duty machines with an NFS mount to the file system with the extracted PMETS files.

Additionally, inspecting the logs, we saw that nearly two-thirds of the time was being spent on the Schematron validation. Our first thought was that the heavy use of regular expressions was consuming a lot of processing time. This however proved not to be the case. We then recollected that Schematron essentially is a code generator, taking as input user assertions, and transforming (run against the file being validated. We had already optimized Conprep and ConprepLite to cache compiled XSL transformations, including the XSL transform g (. Outside the ConprepLite workflow, we serialized the XSL transform generated by Schematron, so that we could inspect the generated code to see what actually was being run. What we found was that (generated code was using a ((through each of the (very large) PMD2 files. We tuned the code to minimize passes through the PMD2 files.

2.3.2 *Quality Assurance*

Although the transformation was tested against many sample files as it was developed, we expected to encounter, in a transformation of such complexity, dealing with input of such complexity, errors of one sort or another, as we in fact did. Key to catching such errors was the capability for large-scale automated validation, both via schema validation and Schematron.

We also performed extracts of the newly generated descriptive metadata for manual review, to verify the correctness of the newly created metadata.

As a matter of policy, Portico retains the original PMETS file along with the new PMD file (which references the now-inactive earlier version) associated with the archival unit. This enables us to re-run the transform as needed, should we discover, at a later time, any errors in our transformation process.

3. REFLECTIONS

It is important to consider the process of migration, not just from the perspective of issues raised by specific file formats, but also in the larger context of the life cycles of systems and software themselves, and in the new use cases for repository content that emerge from ever-evolving expectations of an archive s community of use ((metadata would seem to indicate, it is reasonable to expect over the long term that changes in the large-scale infrastructure of archives, their capabilities, and their communities of use, will themselves necessitate the ability to manage, manipulate, move, and migrate content at very large scales.

Archives and repositories will need to make their own assessments of the necessity, feasibility, and usefulness of such large-scale asset migrations as Portico undertook. They will need to balance the tradeoffs between just-in-time versus large scale pre-emptive migration. And they will need to make these assessments not only about both assets conventionally understood (-generated artifacts such as preservation metadata, which also constitute content, albeit of a less conventional kind, in need of stewardship and preservation.

Preservation institutions

of such migrations. It is comparatively easy to determine the significant properties [6] to be tracked in an XML-to-XML migration such as the one described in this paper. Nevertheless, it is important to articulate that mapping in advance of the transformation, so that the success of the transformation can be tested. This is crucial for the construction of automated tests of the correctness of the transformation another key capability for migration at scale.

Fifteen million of anything is a lot. It is no surprise that it takes a lot of work to manipulate content at that scale, whether that manipulation is a migration, or some other operation. In this case, in terms of elapsed time, Portico spent approximately three to four months planning the migration, and another nine months in its development and execution.

Given the scale at which this was happening, the importance of the content itself, and the many other activities of the staff involved in accomplishing a migration or any similar large-scale, cross-corpus manipulation of content, it is crucially important carefully to analyze, document, plan, and track such efforts. An important part of the planning will be to expect and to allow time and resources for --the unexpected.

4. ACKNOWLEDGEMENTS

The authors would like to acknowledge Evan P. Owens, formerly CTO of Portico, and Vice President for Content Management, ITHAKA, who directed the migration project, the project documents of which were key source materials for this paper.

5. REFERENCES

[1] Beck, Jeff. Report from the Field: PubMed Central, an XML-based Archive of Life Sciences Journal Articles. Presented at International Symposium on XML for the Long Haul: Issues in the Long-term Preservation of XML, Montréal, Canada, August 2, 2010. In *Proceedings of the International Symposium on XML for the Long Haul: Issues in the Long-term Preservation of XML*. Balisage Series on Markup Technologies, vol. 6 (2010). DOI=10.4242/BalisageVol6.Beck01.

[2] Caplan, Priscilla. The Florida Digital Archive and DAITSS: a Working Preservation Repository Based on Format Migration. *International Journal on Digital Libraries* 6.4 (2007): 305 311.

[3] CCSDS. *Reference Model for an Open Archival Information System (OAIS). CCDS 650.0-B-1 Blue Book Issue 1* (2002)

[4] Digital Library Federation. Metadata Encoding and Transmission Standard (METS) Version 1.7. 2008 Web 06 June 2012 from http://www.loc.gov/standards/mets/version17/mets.xsd

[5] Heslop, H., Davis, S. & Wilson, A. *An approach to the preservation of digital records* (2002) Web 08 June, 2012, from http://web.archive.org/wed(h)-6(tt)-3(5192.86 Tm 0.021 Tc[(An)] TJETBT1

fo26 217.94 Tma85192.ico.8( )-189(it)-3(2(l(b)182 Tm[85[7)-2(a)4 0 5l(u)-r5[8519 T(0)-6(0)7>60003>8260570052>8]‹TJ 0 0 1 406.18 264.10JET0052>8]‹TJ 0 0 1 406.18

[7] ISO/IEC 19757-3:2006 Information technology -- Document Schema Definition Language (DSDL) -- Part 3: Rule-based validation Schematron ISO/IEC 2006

[8] Declaration ISO/IEC JTC 1/SC 29 N 3971 Information Technology Multimedia Framework Part 2: Digital Item

[9] Kunze, J. and Rodgers, R. *The ARK Identifier Scheme*. 22 May 2008. Web 06 June 2012 from https://confluence.ucop.edu/download/attachments/16744455/arkspec.pdf?version=1&modificationDate=1261036800000

[10] PREMIS Editorial Committee. PREMIS Data Dictionary, Version 2. Library of Congress March 2008 Web 06 June 2012 from http://www.loc.gov/standards/premis/v2/premis-dd-2-0.pdf

[11]

Ma