This is a preprint of a paper to be published in the proceedings of RCDL’2013 “Digital Libraries: Advanced Methods and Technologies, Digital Collections” [«Электронные библиотеки: перспективные методы и технологии, электронные коллекции»], held on October 14–17, 2013 in Yaroslavl, Yaroslavl Oblast, Russia.

A Model for Integrating the Publication and Preservation of Journal Articles

Kevin S. Hawkins
University of Michigan, Ann Arbor

There are policy, technical, and workflow gaps in library efforts to preserve online journal literature. Since libraries are increasingly involved in journal publishing, HathiTrust, a shared preservation-quality digital repository, is a natural place to archive and provide access to journal literature to ensure its long-term preservation and discoverability. The U-M Library is funding the creation of mPach, an open-source, end-to-end publishing system in which archiving in HathiTrust happens as a byproduct of publication rather than being carried out after the fact. The architecture of mPach, its envisioned workflow, and plans for creating a shared infrastructure for publishing open-access journals are all summarized.

The deficit in journal preservation

Until quite recently, publishers produced documents on physical media, and libraries acquired and preserved copies of these documents. But in the era of the Internet, when publishers host content online, the library's role in acquiring and preserving the content is in jeopardy: without special licensing arrangements such as those often provided by open-access journals, a library has no legal right to make a copy of the content for preservation.

Various business models have evolved to address this situation, especially for journals, which are increasingly available only online. For non-open-access journals, research libraries often negotiate the right to create a digital copy of any content acquired during the period of subscription[1] and make this content available only to their patrons,[2] though few are equipped to provide this kind of restricted access and archiving with integrated browse and search functions. To address the more pressing concern of publishers going out of business without *any* libraries holding a copy of the content, libraries and publishers have collaborated in initiatives like LOCKSS,[3] CLOCKSS,[4] and Portico[5] in order to guarantee that one or more copy of the content will become available if it is no longer available from the publisher. Similarly, the Koninklijke Bibliotheek and Elsevier reached an agreement in 2002 whereby the KB will preserve Elsevier journals under terms similar to those governing journals that use LOCKSS, CLOCKSS, and Portico.[6] Still, there are problems with these models. LOCKSS and

---

[1] Sadie L. Honey. Preservation of electronic scholarly publishing: an analysis of three approaches. Portal: Libraries and the Academy, 5(1):59-75, Jan. 2005.
[2] NISO SERU Standing Committee, SERU: A Shared Electronic Resource Understanding: A Recommended Practice of the National Information Standards Organization. National Information Standards Organization (NISO), May 2012. http://www.niso.org/publications/rp/RP-7-2012_SERU.pdf.
[3] Lots of Copies Keeps Stuff Safe. http://www.lockss.org/.
[4] CLOCKSS. http://www.clockss.org/.
[5] Portico. http://www.portico.org/.
[6] National Library of the Netherlands and Elsevier Science make digital preservation history: permanent digital archive assures perpetual accessibility of scientific heritage. August 20, 2002. http://www.kb.nl/en/news/news-archive-2002/national-library-of-the-netherlands-and-elsevier-science-make-digital-preservation-history.

CLOCKSS use web crawling, which captures only the appearance of webpages but not their underlying structure or search functionality. Portico and the KB, on the other hand, rely on publishers to deliver journal articles in valid file formats, and not just the version first published but also any corrected versions of these articles.

One way to ensure that a library always has access to the latest content is for the library to operate the very system used to publish the journal. A survey in 2010 of a cross-section of North American academic libraries found that, of 144 responding institutions, 43 offered "operational publishing services" to their scholars at the institution.[7] Of these 43 institutions, most host publications using open-source software such as Open Journal Systems (OJS)[8] or DSpace,[9] while about a quarter use Digital Commons,[10] a hosted platform provided by bepress. Unfortunately, all of these platforms deliver to users only those files (primarily PDF files) created and uploaded by a journal editor. Since the library is not in a position to control the software and workflows used to create these files, the library can only provide bitwise preservation of the files, severely hampering future migration of the content.

A higher standard for preservation

Since libraries are increasingly involved in journal publishing, HathiTrust,[11] a shared preservation-quality digital repository, is a natural place to archive and provide access to journal literature to ensure its long-term preservation and discoverability. HathiTrust already archives and provides access to reformatted library holdings, but the University of Michigan Library, a founding member of HathiTrust, sees an opportunity to use HathiTrust for publishing born-digital journals as well. To develop an infrastructure in support of low-cost university-based publishing that addresses the needs and values of both content creators and librarians, the U-M Library is funding the creation of mPach,[12] an open-source, end-to-end publishing system in which the act of publishing and the act of archiving are unified. In other words, archiving in HathiTrust happens as a byproduct of publication rather than being carried out after the fact. mPach leverages existing components of HathiTrust and available open-source software where appropriate.

Archiving is not as simple as saving a copy of a file produced by a journal editor, as OJS and institutional repositories generally do. Instead, the content needs to be stored in a format that allows digital preservation. PDF/A, a non-proprietary variant of the PDF family standardized as ISO 19005, is often suggested for such needs, but even a PDF/A file is poorly suited for use with screen readers for the visually impaired and for any non-paginated display, and is suboptimal even for searching and data mining.

---

[7] James L. Mullins, Catherine Murray-Rust, Joyce L. Ogburn, Raym Crow, October Ivens, Allyson Mower, Daureen Nesdill, Mark Newton, Julie Speer, and Charles Watkinson. Library Publishing Services: Strategies for Success: Final Research Report. March 2012. http://wp.sparc.arl.org/lps/.

[8] Open Journal Systems. http://pkp.sfu.ca/ojs/.

[9] DSpace. http://www.dspace.org/.

[10] Digital Commons. http://digitalcommons.bepress.com/.

[11] HathiTrust Digital Library. http://www.hathitrust.org/.

[12] mPach. http://www.lib.umich.edu/mpach.

Rather than preserving the paginated appearance of a document, the text of the article needs to be stored in a format that reflects its structure and semantics, with associated media in formats that can be preserved and rendered. mPach has developed a specification for journal articles that uses the Journal Article Tag Suite (JATS), an application of NISO Z39.96-2012,[13] for the text and stores this with high-quality versions of media objects and with a METS record containing structural and preservation metadata.

An overview of mPach

There are three major parts of mPach (see also figure 1), each of which includes components in various stages of development at the time of writing:

- **the peer review and editorial system**: what authors and reviewers interact with
- **Prepper**: what prepares the article for ingest into HathiTrust for archiving and publication
- **modified HathiTrust components**: various modifications to existing components of the HathiTrust environment to support born-digital journal articles

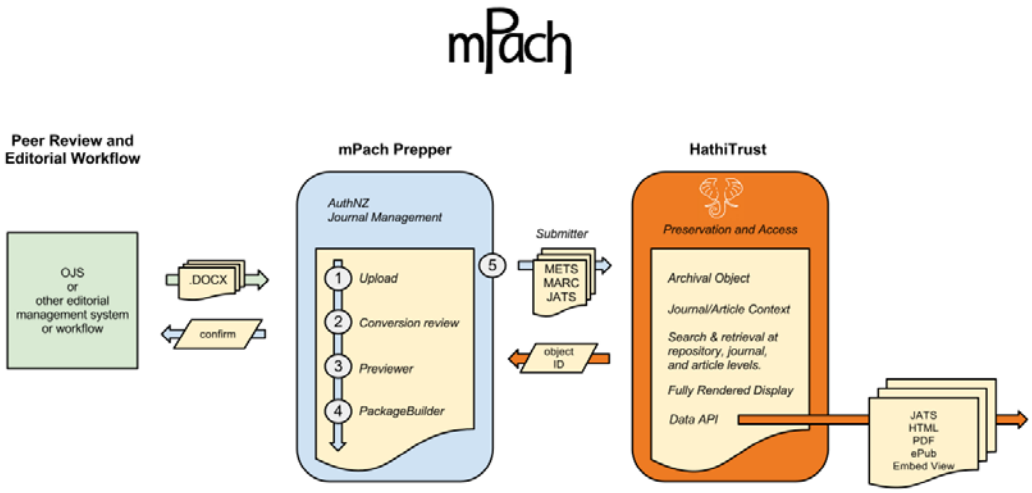

Figure 1: Major parts of mPach

As a modular system, mPach could be used with any peer review and editorial system that is capable of interacting with Prepper; however, the developers have chosen to provide OJS as the default option. Despite having no support for digital preservation, OJS is already widely used for library-based journal publishing, and mPach's integration with this software will allow for a smooth transition of journals already published using OJS into the HathiTrust repository. Integration with mPach requires that manuscripts that reach the "layout" stage in OJS be sent to Prepper, which prepares the HathiTrust Submission Information Package (SIP).

Prepper provides a user interface for the editor of a journal: a dashboard for administering the journal and putting manuscripts through a production process—akin to composition and typesetting—that prepares all content

[13] Journal Article Tag Suite. http://jats.nlm.nih.gov/.

according to the preservation standard developed for mPach content in HathiTrust. Prepper invokes Norm, a Python application developed to convert manuscripts from Office Open XML ("DOCX") format[14] into XML that conforms to JATS. DOCX is the default option because, like OJS, it is widely used in the editorial process of journals published by libraries. The Prepper interface also guides the staff member through a review of validation errors detected by Norm's conversion, uploading high-resolution figures, supplying "alt text" for figures, previewing the article as rendered using the default stylesheet (based on the Preview XSLT stylesheets[15]), uploading supplementary material,[16] and submitting for ingest into HathiTrust.

mPach requires a number of significant modifications to HathiTrust components and workflows originally designed to support reformatted print materials. The reading interface in HathiTrust, which previously supported only rendering of digitized page images, renders JATS XML in HTML and allows a user to download a dynamically generated PDF and EPUB, display metadata specific to articles (figure 2), and link to a special "collection" for the journal in HathiTrust's Collections application[17] that allows for browsing volumes and issues of the journal (figure 3).

[14] Office Open XML. Wikipedia. http://en.wikipedia.org/wiki/Office_Open_XML.
[15] NISO Journal Article Tag Set (JATS) version 1.0: Preview XSLT stylesheets. https://github.com/NCBITools/JATSPreviewStylesheets.
[16] Recommended Practices for Online Supplemental Journal Article Materials: a recommended practice of the National Information Standards Organization and the National Federation of Advanced Information Services. January 2013. http://www.niso.org/publications/rp/rp-15-2013.
[17] Collections. HathiTrust Digital Library. http://babel.hathitrust.org/cgi/mb.

Figure 2: Mockup of an article viewed in HathiTrust's user interface

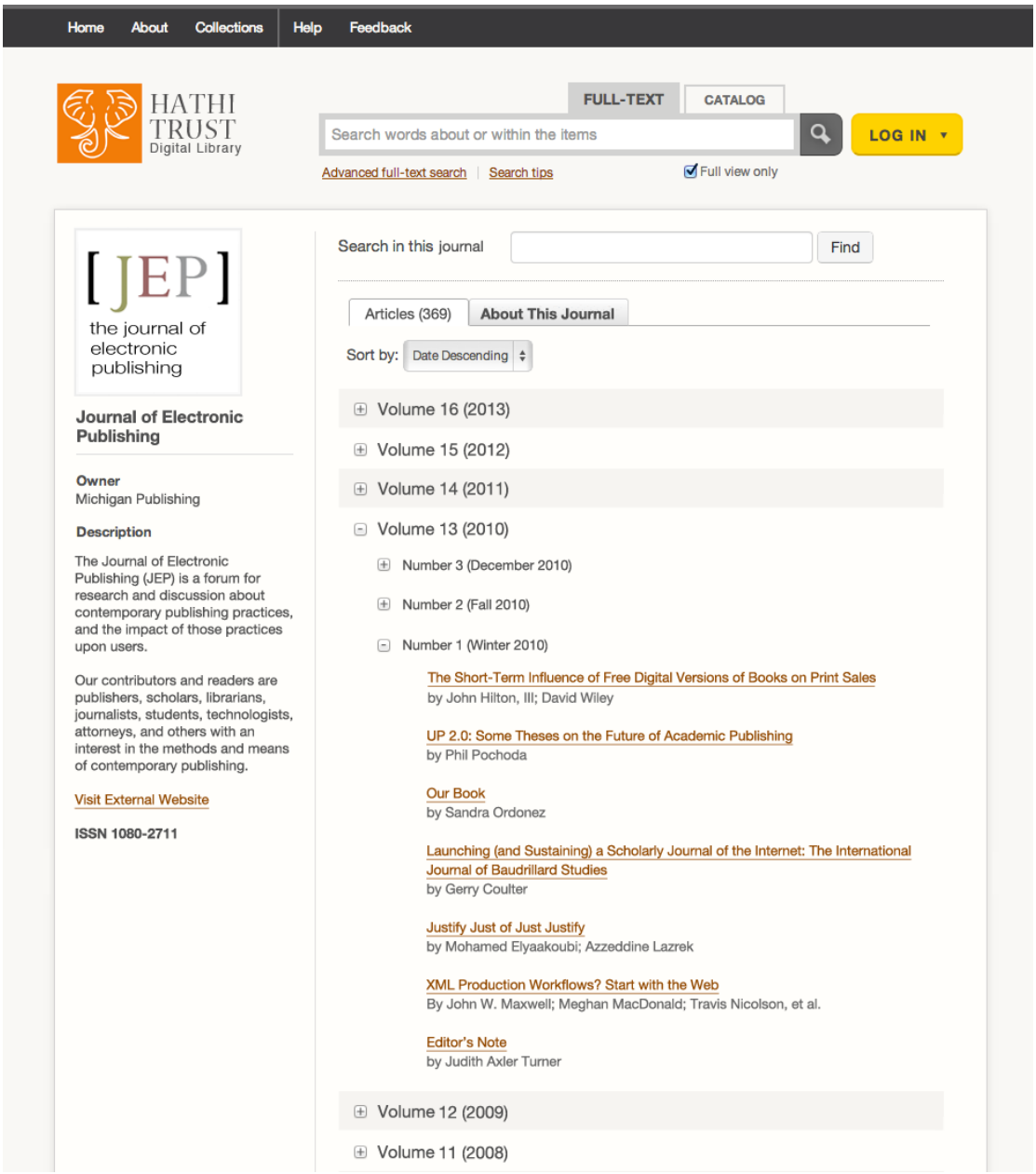

Figure 3: Mockup of a journal viewed in HathiTrust's user interface

Discovery of known items in HathiTrust using metadata like title and author is currently provided for by a catalog of MARC records, with one per item in the repository. For mPach, each article has its own analytic catalog record, tied to a monographic record for the journal as a whole. Finally, the HathiTrust Data API[18] allows for the content of each article to be retrieved for use outside of the native HathiTrust interface.

Note that by policy HathiTrust only closes access to content for legal reasons, not because a rightsholder wants to restrict access. Therefore, mPach only supports the publishing of open-access journals.

[18] HathiTrust Data API. http://www.hathitrust.org/data_api.

Workflow

In the typical workflow for publishing a journal using mPach, a journal editor uses OJS to manage submissions, peer review, and the editing process. Once an article reaches the "layout" stage (where a combination of composition and typesetting allows the article to be formatted in a consistent way), the journal editor formats it according to a predefined list of styles in Microsoft Word and submits the article in DOCX to mPach's Prepper, which guides the editor through conversion to JATS XML, preparation of the SIP, and submission for ingest. Prepper keeps track of articles so that a revised version can be submitted for ingest. Currently the ingest process overwrites any previous version of an item with the same identifier, but eventually HathiTrust will archive past versions and allow users to navigate among them.

mPach as a shared infrastructure

The U-M Library plans to host the Prepper system, including the submission module, to facilitate authorized deposit of content, and will make this system available for use by organizations wishing to publish journal literature in HathiTrust. The developers envision extending the Norm component to handle OpenDocument ("ODT")[19] and LaTeX as input formats, each of which is more commonly used in certain communities. Furthermore, if the Book Interchange Tag Suite[20] is adopted as a standard, the mPach architecture might be extended to support monograph publishing. While mPach is currently being developed to meet the needs of the U-M Library, the contribution of the sourcecode to the planned HathiTrust Development Environment should foster contributions from developers not at U-M and therefore lead to the creation of a truly shared infrastructure for publishing open-access scholarly journals.

[19] OpenDocument. Wikipedia. http://en.wikipedia.org/wiki/OpenDocument.
[20] Book Interchange Tag Suite (BITS) 0.2 DRAFT. http://jats.nlm.nih.gov/extensions/bits/.