Taking the Long Way Around: Improving the Display of HathiTrust Records in the Primo Discovery System

Jason Alden Bengtson and Jason Coleman

ABSTRACT

As with any shared format for serializing data, Primo's PNX records have limits on the types of data which they pass along from the source records and into the Primo tool. As a result of these limitations, PNX records do not currently have a provision for harvesting and transferring rights information about HathiTrust holdings that the Kansas State University (KSU) Library system indexes through Primo. This created a problem, since Primo was defaulting to indicate that all HathiTrust materials were available to KSU Libraries (K-State Libraries) patrons, when only a limited portion of them actually were. This disconnect was infuriating some library users, and creating difficulties for the public services librarians. There was a library-wide discussion about removing HathiTrust holdings from Primo altogether, but it was decided that such a solution was an overreaction. As a consequence, the library IT department began a crash program to attempt to find a solution to the problem. The result was an application called hathiGenius.

INTRODUCTION

Many information professionals will be aware of Primo, the web scale discovery tool provided by Ex Libris. Web scale discovery services are designed to provide indexing and searching User Experiences, not only for the library's holdings (as with a traditional Online Public Access Catalog), but also for many of a library's licensed and open access holdings. Primo offers a variety of useful features for search and discovery, taking in data from manifold sources and serializing them into a common format for indexing within the tool. However, such applications are still relatively young, and the technologies powering them have not fully matured. The combination of this lack of maturity and deliberately closed architecture between vendors leads to several problems for the user. One of the most frustrating is errors in identifying full-text access availability.

As with any shared format for serializing data, Primo's PNX (Primo Normalized XML) records have limits on the types of data they pass from the source records into the Primo tool. As a result of these limitations, PNX records do not currently have a provision for harvesting and transferring rights information about HathiTrust holdings that the K-State Libraries system indexes through Primo. This created a problem in the K-State Libraries' implementation, since Primo was defaulting to indicate that all HathiTrust materials were available to K-State Libraries patrons, when only a limited portion of them actually were. This disconnect was infuriating some library users, and creating difficulties for the public services librarians. There was a library-wide discussion about removing HathiTrust holdings from Primo altogether, but it was decided that such a solution was an overreaction. As a consequence, the library IT Services department began a crash program to attempt to find a solution to the problem.

Jason Bengtson (jbengtson@ksu.edu) is Head of IT Services for Kansas State University Libraries.
Jason Coleman (coleman@ksu.edu) is Head of Library User Services for Kansas State University Libraries.

HATHITRUST'S DIGITAL LIBRARY AS A COLLECTION IN PRIMO CENTRAL

HathiTrust was established in 2008 as a collaboration among several research libraries that were interested in preserving digital content. As of the beginning of March 2018, the collaborative's digital library contained more than sixteen million items, approximately 37 percent of which were in the public domain.[1] Ex Libris' Primo Central Index (PCI), which serves as Primo's built-in index of articles from various database providers, includes metadata for the vast majority of the items in HathiTrust's digital library, providing inline frames within the original Primo user interface to directly display full-text content of those items that the library has access to. Libraries subscribing to Primo choose whether or not to make these records available to their users. K-State Libraries, like many other Primo Central clients, elected to activate HathiTrust in its instance of Primo, which it has branded with the name Search It.

The unmodified version of Primo Central identified all records from HathiTrust's digital library as available online, regardless of the actual level of access provided to users. Users who discovered a record for an item from HathiTrust's digital library were presented with a conspicuous message indicating that full text was available and two links named *view it* and *details*. An example of the appearance of these search results is shown in figure 1. After clicking the "view it" tab, the center window would display the item's homepage from HathiTrust's digital library inside an iframe. Public domain items would display the title page of the item and present users with an interface containing numerous visual indicators that they were viewing an ebook (see figure 2 for an example). Items with copyright restrictions would display a message indicating that the item is not available online (see figure 3 for an example).

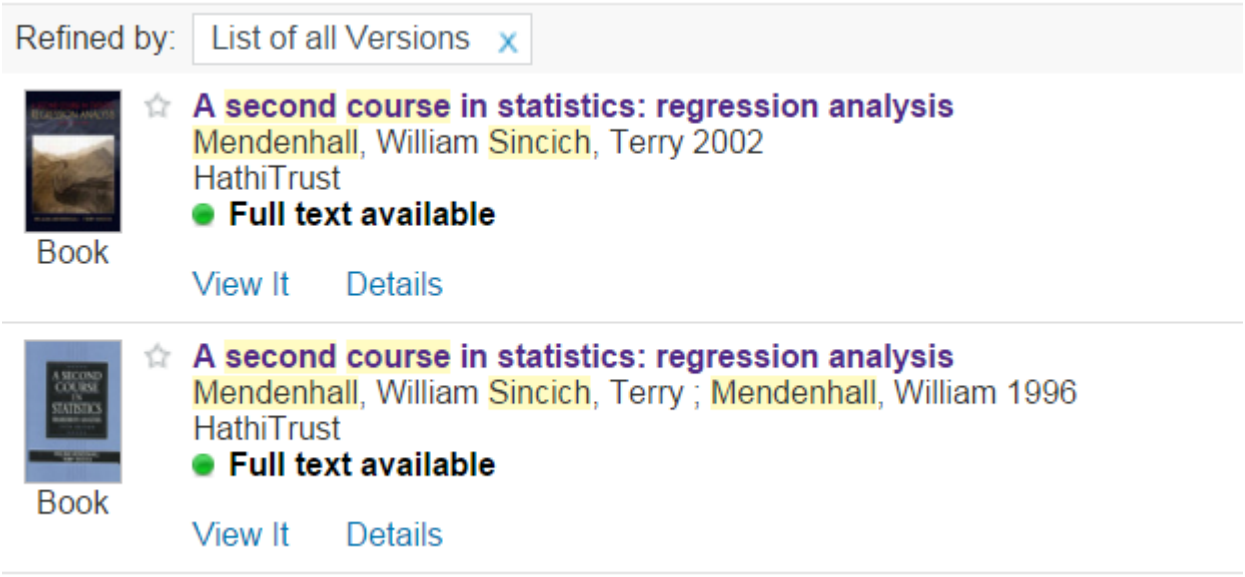

Figure 1. Two books from HathiTrust as they appeared in Search It prior to implementation of hathiGenius.

https://doi.org/10.6017/ital.v38i1.10574

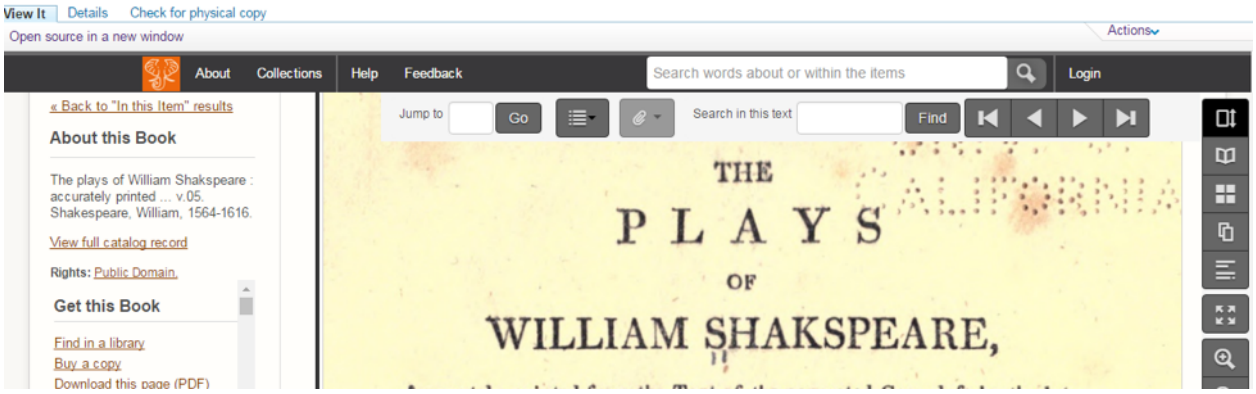

Figure 2. HathiTrust result for an item in the public domain.

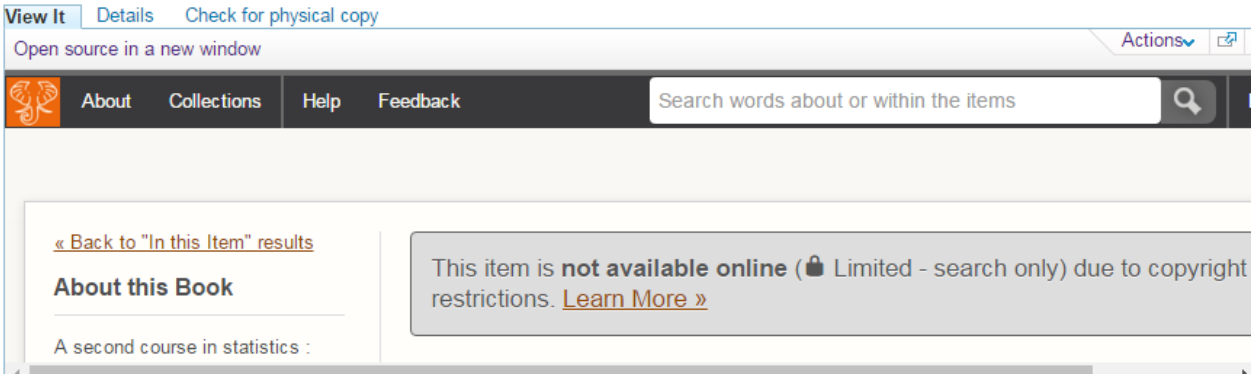

Figure 3. HathiTrust's homepage for an item that is not in the public domain.

Despite the intentions evident in the design of the Primo interface, availability of HathiTrust records was not being accurately reflected in the list of returns. The size of the indices underlying web scale discovery systems and the number of configurations and settings that must be maintained locally introduce a variety of failure points that can intercede when patrons attempt to access subscribed resources.[2] One of the failure points identified by Sunshine and Carter is inaccurate knowledgebase information. The scope of inaccurate information about HathiTrust items in Primo Central Index constituted a particularly egregious example of this type of failure.

PATRON REACTION TO MISINFORMATION ABOUT ACCESS TO HATHITRUST

Between the time HathiTrust's Digital Library was activated in Search It and the time the HathiGenius application was installed at least thirty patrons contacted K-State Libraries to ask why they were unable to access a book in HathiTrust when Search It had indicated that full text was available for the book. Many of these expressed frustration at frequently encountering this error (for an example, see figure 4).

1:08 263899577777596010930881 33 I find it misleading that the Search It function often finds a book I am interested in, but sometimes says it is available online; however, oftentimes it takes me to the Hathi Trust webpage for the book where I am told it is NOT available online. Is this because our library has had to give up their subscription to this service?

1:08 me Hi!

1:09 me That is definitely frustrating - and we are trying to find a way to correct it.

1:10 me It does not have to do with our subscription, but rather the metadata we receive from HathiTrust and its compatibility (or rather, incompatibility) with Search It

1:11 263899577777596010930881 33 Okay, so I guess I better ask for the book I am seeking (The Emperor's Mirror) through ILL.

1:11 me That'd probably be your best bet, but let me take a look - one moment

1:14 me Yes, ILL does look best. Please note that the ILL department will be closed after today until January

1:14 263899577777596010930881 33 Got it. Thanks. I hope the Hathi Trust issue is resolved soon. **(I have seen this problem all semester and finally got so frustrated to ask about it.)**

1:15 263899577777596010930881 33 Have a Happy holiday!

1:15 me You as well! And yes, I hope we can figure it out ASAP

1:15 me (it's frustrating for us, too!)

1:20 263899577777596010930881 33 has left the conversation

Figure 4. Chat transcript revealing frustration with inaccurate information about availability of items in HathiTrust.

STAFF REACTION TO MISINFORMATION ABOUT ACCESS TO HATHITRUST

Reference staff at K-State Libraries use a ticketing system to report electronic resource access problems to a team of librarians who troubleshoot the underlying issues. Shortly after the HathiTrust library was activated in Search It, reference staff submitted several tickets about problems with access to items in that collection. Members of the troubleshooting team responded quickly and informed the reporting librarians that the problem was one beyond their control. This message was slow to reach the entirety of the reference staff and was not always understood as being applicable to the full range of access problems our patrons were experiencing. Samples and Healy note that this type of decentralization and reactive orientation is common in electronic resource troubleshooting.[3] Like them, K-State Libraries recognized a need to develop best practices to obviate confusion. We also found ourselves pining for a tool such as that described by Collins and Murray that could automatically verify access for a large set of links.[4]

The extent of displeasure with the situation was so severe that some librarians stated they were loath to promote Search It to students since several million records were so conspicuously inaccurate.

https://doi.org/10.6017/ital.v38i1.10574

TECHNICAL CHALLENGES

THE K-State Libraries IT department wanted to fix the situation, in order to provide accurate expectations to their users, but doing so presented severe technical challenges, the most significant of which stemmed from the lack of rights information in the PNX record in Primo. Without more accurate information on availability, user satisfaction seemed destined to remain low. Research into patron use of discovery layers predicted this unsurprising dissatisfaction. OCLC's (2009) research into what patrons want from discovery system led the researchers to conclude that "a seamless, easy flow from discovery through delivery is critical to end users. This point may seem obvious, but it is important to remember that for many end users, without the delivery of something he or she wants or needs, discovery alone is a waste of time."[5] A later usability study reported: "Some participants spent considerable time looking around for features they hoped or presumed existed that would support their path toward task completion."[6] Additionally, the perceived need to customize discovery layers so that they reflect the needs of a particular research library is hardly new, or exclusive to K-State Libraries. The same issue was confronted by catalogers at East Carolina University, as well as catalogers at UNC Chapel Hill.[7] Nonetheless, the challenge posed by discovery layers comes with opportunity, as James Madison University discovered when their EBSCO Discovery Service widget netted almost twice the usage of their previous library catalog widget, and as the University of Colorado discovered when they observed users attempting to use the discovery layer search box in "Google-like" ways that could potentially aid discovery layer creators (as well as library IT departments) in both design and in setting expectations.[8]

As previously noted, Primo's results display is driven by PNX records (see figure 5 for an example). The single most fundamental challenge was finding a way to get to holdings rights information despite that data not being present in the PNX records, or, consequently, the search results that showed up in the presentation layer. There was no immediate option to create a solution that leveraged "server-side" resources, where the data itself resided and was transformed, since K-State Libraries subscribes to Primo as a hosted service, and Ex Libris provided no direct server-side access to K-State Libraries. Some alternative way had to be found to locate the rights data for individual records and populate it into the Primo interface.

Upon assessing the situation, the Assistant Director, IT (AD) decided that one potential approach would be to independently query the HathiTrust bibliographic Application Programming Interface (API) for rights information. This approach solved a number of fundamental problems, but also posited its own questions and challenges:

1. Some server-side component would still be needed for part of the query . . . where would that live and how could it be made to communicate with the Javascript K-State Libraries had injected into its Primo instance?
2. How to best isolate HathiTrust object identifiers from Primo and then use them to launch an API query?
3. How to keep those responses appropriately "pinned" to their corresponding entries on the Primo page?
4. How would the HathiTrust bibliographic API perform under load from Search It queries?

Answering these questions would require significant research into the HathiTrust bibliographic API documentation, and extensive experimentation.

```
▼<record xmlns="http://www.exlibrisgroup.com/xsd/primo/primo_nm_bib" xmlns:sear="http://www.exlibrisgroup.com/xsd/jaguar/search">
  ▼<control>
     <sourcerecordid>uc1.32106011231518</sourcerecordid>
     <sourceid>hathi_trust</sourceid>
     <recordid>TN_hathi_trustuc1.32106011231518</recordid>
     <sourcesystem>Other</sourcesystem>
     <pqid>EBL5174418</pqid>
   </control>
  ▼<display>
     <type>book</type>
     <title>A second course in statistics: regression analysis</title>
     <creator>Mendenhall, William</creator>
     <contributor>Mendenhall, William ; Sincich, Terry</contributor>
     <publisher>Upper Saddle River, N.J.: Prentice Hall</publisher>
     <creationdate>1996</creationdate>
     <identifier><b>ISBN: </b>0133968219</identifier>
     <description>xxi, 899 p.</description>
     <source/>
    ▼<subject>
       Regression Analysis ; Statistics ; Commercial Statistics
     </subject>
     <version>2</version>
   </display>
  ▼<links>
     <openurl>$$Topenurl_article</openurl>
     <thumbnail>$$TPCamazon_thumb</thumbnail>
     <thumbnail>$$TPCgoogle_thumb</thumbnail>
     <openurlfulltext>$$Topenurlfull_article</openurlfulltext>
    ▼<linktorsrc>
       $$Uhttp://hdl.handle.net/2027/uc1.32106011231518$$EView_full_text_in_HathiTrust
     </linktorsrc>
   </links>
```

Figure 5. A portion of the PNX record for http://hdl.handle.net/2027/uc1.32106011231518 (the second item shown in figure 1).

BUILDING THE APPLICATION

Of these four questions, the first was easily the most challenging: where would the server-side component live and how would it work? The K-State Libraries IT Services department had, in the past, made a number of significant modifications to the appearance and functionality of the Primo application by adding JavaScript to the static HTML tiles used in the Primo interface. However, generally speaking, JavaScript cannot successfully request data from outside of the domain of the web document it occupies. Requesting data from an API across domains requires the mediation of a server-side appliance. The AD constructed one for this purpose, using the PHP programming language. This script would serve as an intermediary between the JavaScript in Primo and the HathiTrust API. The appliance accepted data from the Primo JavaScript in the form of the contents of http variables (encoded in the URL of the GET request to the PHP appliance), then used those values to query the HathiTrust API. However, since this server-side appliance did not reside in the same domain as K-State Libraries' Primo instance, the problem of getting the returned API data from the PHP appliance to the JavaScript still remained.

This problem was solved by treating the PHP appliance as a JavaScript file for purposes of the application. While JavaScript cannot load data from another domain, a web document may load actual JavaScript files from anywhere on the web. The hathiGenius appliance takes advantage of this fact by calling the PHP appliance programmatically as a JavaScript file, with a JavaScript Object Notation (JSON) version of the identifiers of any HathiTrust entries encoded as part of the URL used to call the file. The PHP script runs the queries against the API and returns a JavaScript file consisting of a single variable containing the JSON data encoding the availability information for the HathiTrust entries as supplied from the bibliographic API . . . essentially appearing to the browser as a standard JavaScript file.

https://doi.org/10.6017/ital.v38i1.10574

The second and third problems were intrinsically interrelated, and essentially boiled down to finding a unique identifier to use in an API query from the HathiTrust entries. The most effective way to handle these queries was to use the "htid" identifier, which was largely unique to HathiTrust entries, could be easily extracted from any entries that contained it, and would form the basis of the PHP script's request to the HathiTrust RESTful API to obtain rights information. In the process of harvesting the htid, hathiGenius also copies the id for the object in the webpage that serves as the entry in the list of Primo returns containing that htid. As the data is moved back and forth for processing, the htids, and later the resultant JSON data, remain paired to the object id for the entry in the list of returns. When hathiGenius receives the results of the API query, it can then easily rewrite those entries to reflect the rights data it obtained.

The fourth question has been fully answered with time. To this point, well over a year after hathiGenius was activated in production, Library IT has not observed any failure of the API to deliver the requested results in testing, and no issues to that effect have been reported by users. Log data indicates that, even under heavy load, the API is performing to expectations.

FURTHER MODIFICATIONS

Originally, the hathiGenius application supplied definitive states of available or unavailable for each entry. However, some experimentation showed this approach to be less than optimal. Since the bibliographic API cannot be queried by Kansas State University as a specific user, but rather was being queried for general access rights, the possibility still existed for false negatives in the future, if Kansas State University's level of access to HathiTrust changed. The data returned from the API queries, when drilled down, just consisted of the usRightsString property from the API, which corresponded to open-source availability, and did not account for any additional items available to the library by license in the future.

After the application had been active for a short time, to mitigate this potential issue, the "not available" state (consisting of an application of the "EXLResultStatusNotAvailable" class to the HathiTrust entry) was "softened" into an application of the "EXLResultStatusMaybeAvailable" class and verbiage asking users to check the "View It" tab for availability.

A few weeks after deployment, IT received a ticket indicating hathiGenius was failing to work properly. The source of the problem proved to be detailed bibliographic pages for items in a search results list, which were linking out from the search entries. These pages used a different class and object structure than the search results pages in Primo, requiring that an additional module be built into hathiGenius to account for them. Once the new module was added to the application and put into place, the problem was resolved.

A second issue presented itself some weeks later, when a few false negatives were reported. At first, the assistant director assumed that licensing had changed, creating a disparity between the access information from the usRightsString property and the library's actual holdings. However, upon investigation it was clear that hathiGenius was dropping some of the calls to the HathiTrust bibliographic API. The API itself was performing as expected under load, however, and the failure proved to be coming from an unexpected source. The PHP script used by hathiGenius to interface with the API was employing the cURL module, which, in turn, was using its own, less secure certificate to establish a Secure Socket Layer (SSL) connection to the HathiTrust server. Once the

script was refactored to employ the simpler file_get_contents function, which relied upon the server's main SSL certificate, the problem was fully resolved.

hathiGenius also had a limited vulnerability to bad actors. While the internal script's destination hardwiring prevented hathiGenius from being used as a generic tool to anonymously query APIs, the library did encounter a situation in which a (probably inadvertently) malicious bot repeatedly pinged the script, causing it to use up system resources until it interrupted other services on the host machine. Modifications were added to the script to provide a simple check against requests originating from Primo. Additionally, restrictions were placed on the script so that excessive resource use would cause it to be intermittently deactivated. While not perfect solutions, these measures have prevented a repeat of the earlier incident.

K-State Libraries has recently finished work on its version of the new Primo User Interface (Primo New UI), which was moved into production this year. The new interface has a completely different client-side structure, requiring a very different approach to integrating hathiGenius.[9]

APPEARANCE OF HATHITRUST RESULTS IN PRIMO AFTER HATHIGENIUS

When the hathiGenius API does not find a usRights property, we configured Primo to display a yellow dot and the text "Please Check Availability with the View It Tab" (see figure 6 for an example). As noted earlier, we originally considered this preferable to displaying a red dot and the text "Not Available Online," because there might be instances in which the item is actually available in full view through HathiTrust despite the absence of usRights in the record.

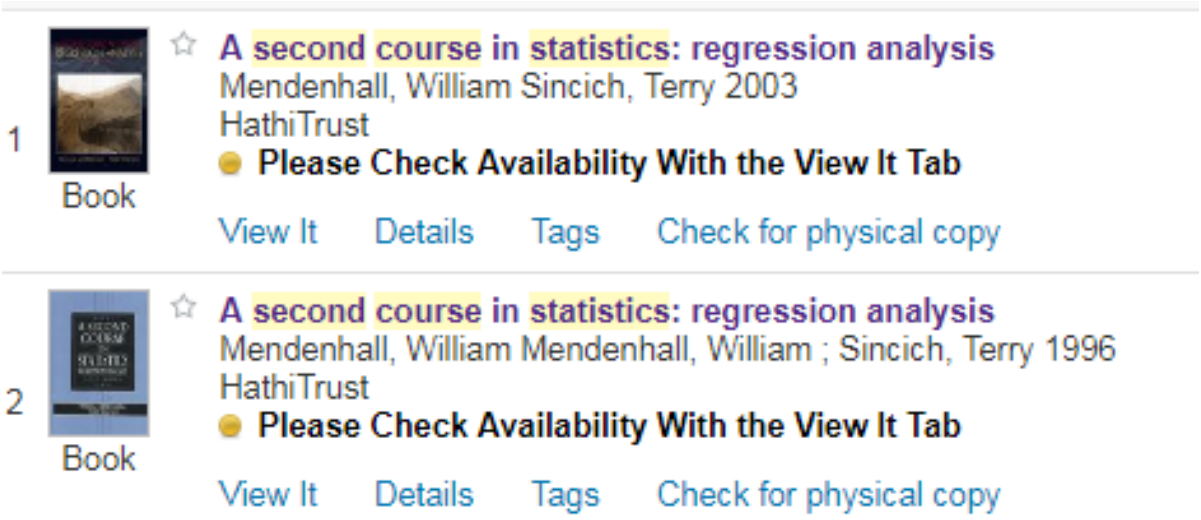

Figure 6. Two books for which hathiGenius found no usRights in HathiTrust.

When the hathiGenuis API finds usRights, we configured Primo to display a green dot and text "Available Online" (see figure 7 for an example).

https://doi.org/10.6017/ital.v38i1.10574

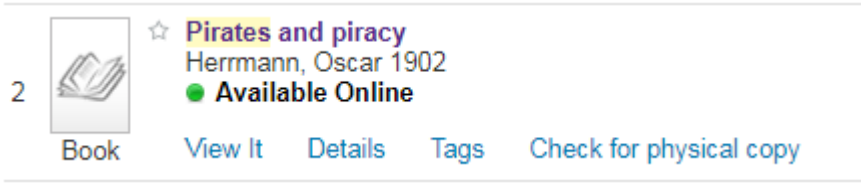

Figure 7. A book for which hathiGenius found usRights.

PATRON RESPONSE

Since the beginning of 2017, the reference staff at K-State Libraries have received no reports of patrons encountering situations in the original user interface in which Primo indicates that full text is available but HathiTrust is only providing a preview. However, a small number of patrons (at least four) expressed confusion at seeing a result in Primo and discovering that the full-text is not available. Some of those patrons noted that they saw the text "Please Check Availability with the View It Tab," and inferred that this was meant to state that the full-text was available. Others indicated that they never considered that we would include results for books that we do not own. These responses add to the body of literature documenting user expectations that everything should be available in full-text in an online library and that systems should be easy to use.[10]

INTERNAL RESPONSE

In order to gauge the feelings of K-State Libraries' staff who regularly assist patrons with reference questions, the authors crafted a brief survey (included in appendix A). Respondents were asked to indicate whether they had noticed a positive change following implementation of hathiGenius, a negative change, or no change at all. They were also invited to share comments. The survey was distributed to thirty individuals. Twelve (40 percent) of those thirty responded to the survey.

The survey response indicated a great deal of ambivalence by reference staff toward the change, with four individuals (33 percent) indicating they had not noticed a difference, and another four (33 percent) indicating that they had noticed a difference, but that it had not improved the quality of search results. Only two (17 percent) of the respondents revealed that they had noticed an improvement in the quality of the search results. One (9 percent) respondent indicated that they felt that the HathiTrust results had gotten noticeably worse since the introduction of hathiGenius, although they did not elaborate on this in the survey question which invited further comment. The remaining respondent stated that they did not have an opinion.

Four comments were left by respondents, including one which indicated displeasure with the new, softer verbiage for hathiTrust "negatives," and one who claimed that the problem of false positives persisted, despite such feedback not being seen by the authors through any of the statistical modalities currently used for recording reference transactions. One user praised hathiGenius, while another related broad displeasure with the decision to include HathiTrust records in Search It. That individual claimed that almost none of the results from HathiTrust were available and stated that the hope engendered by the presence of the HathiTrust results and the corresponding suggestion to check the View It tab was always dashed, to the detriment of patron satisfaction.

THE NEW UI

As previously mentioned, in late 2018, K-State Libraries adopted the Primo New UI created by Ex Libris. This new user interface was built in Angular, and changed many aspects about how hathiGenius had to be integrated into Primo. The K-State Libraries' IT department completed a refactoring (reworking application code to change *how* an application works, but not what it does) of hathiGenius to integrate it with the new UI and released it into production in September 2018.

As an interesting aside, the IT department did not initially prioritize the reintegration of hathiGenius, due to the ambivalence of the response to the application evidenced by the survey conducted for this paper. However, shortly after Search It was switched over to the new UI, complaints about the HathiTrust results again displaying inaccurate availability information began to come in to the IT department via both email and tickets from reference staff. As the stridence of the response increased, the project was reprioritized, and the work completed.

FUTURE DIRECTIONS

As previously mentioned, hathiGenius currently uses the very rough "usRightsString" property value from the HathiTrust bibliographic API. However, the API also delivers much more granular rights data for digital objects. A future version of the app may inspect these more granular rights codes and compare them to rights data from K-State Libraries in order to more definitively provide access determinations for HathiTrust results in Primo should the licensing of HathiTrust holdings be changed.

Similarly, since htid technically only resolves to the volume level, a future version may additionally harvest the HathiTrust record number, which appears to be extractable from the Primo entries.

Based on feedback from the survey, the "soft negative" verbiage used in hathiGenius was replaced with a firmer negative. This decision proved especially sagacious given that, once the early issues with certificates and communication with the HathiTrust bibliographic API were sorted out, the accuracy of the tool seemed to be fully satisfactory. Another problem with the "soft negative" was the fact that it asked users to click on the View-it tab, when many users simply chose to ignore the tabs and links in the search results, instead clicking on the article title, as found in a usability study on Primo conducted by the University of Houston Libraries.[11]

It is also worth noting the one survey respondent who is apparently not seeing an improvement in HathiTrust accuracy. If the continued difficulties they have indicated can be documented and replicated, the IT department can examine those complaints to investigate where the tool may be failing.

DISCUSSION

One interesting feature of this experience is the seeming disconnect between library reference support staff and users in terms of the perception of the efficacy of the tool. This disconnect is all the more curious given the negative reaction displayed by reference support staff when hathiGenius became unavailable temporarily upon introduction of the Primo New UI. Part of this perceived disconnect may be a result of the fact that staff were given a survey instrument, while the reactions of users have been determined largely via null results (a lack of complaints to, or

https://doi.org/10.6017/ital.v38i1.10574

requests for assistance from, service point staff). However, given the dramatic drop in user complaints compared to the ambivalent reaction to the tool by most of the survey respondents, it appears that the staff had a much less enthusiastic response to the intervention than patrons. A few possibilities occur to the authors, including a general dislike for the discovery layer by reference librarians, a general disinclination toward a technological solution by some respondents, or the initial perception by at least part of the reference staff that the problem was not significant. As noted by Fagan et al., the pivot toward discovery layers has not been a comfortable one for many librarians.[12] Until further research can be conducted on this, and reactions to similar customization interventions, these possibilities remain speculation.

One particular feature of note with hathiGenius is the use of what one of the authors refers to as "sidewise development" to solve problems that seem to be intractable within a proprietary, or open source, web-based tool. While not a new methodology in and of itself, the author has mainly encountered this type of design in ad-hoc creations, rather than as a systematic approach to problem-solving. Instead of relying upon the capabilities of Primo, this type of customization made its own query to a relevant API and blended that external data with the data available from Primo seamlessly within the application's presentation layer in order to facilitate a solution to a known problem. The solution created in this fashion was portable, and unaffected by most updates to Primo itself. Even the transition to the New UI required changes to the "hooks" and timing used by the JavaScript, rather than any substantial rewrite of the core engines of the application. This methodology has been used repeatedly by K-State Libraries IT Services to solve problems where other interventions would have necessitated the creation of specialized modules, or the rewriting of source code; both of which would be substantially affected by updates to the product itself, and which would have been difficult to improve or version without down time to the affected product. Similar solutions have seen tools independently query an application's database in order to inject the data back into the application's presentation layer, bypassing the core functionality of the application.

CONCLUSION

Reactions at this point from users, and at least some library staff, have been positive. While not a perfect tool, hathiGenius has improved the user experience, removing a point of frustration and an area of disconnect between the library and its users. The application itself is fully replicable by other institutions (as is the general model of sideways development), allowing them to improve the utility of their Primo instances. As with many possible customizations to discovery layers, hathiGenius provides fertile ground for additional work, research, and refinement, as libraries struggle to find the most effective ways to implement discovery tools within their own environments. Beyond hathiGenius itself, the sideways development method provides a powerful tool for libraries to improve the tools they use by integrating additional functionality at the presentation layer level. Tackling the problem of inaccurate full-text links in discovery layers is only one application of this approach, but it is an important one. As libraries continue to strive to improve the results and usability of their search offerings, the ability to add local customizations and improvements will be an essential feature for vendors to consider.

APPENDIX A. FEEDBACK SURVEY

Q1 In January 2017, the library began applying a tool (called hathiGenius) to the HathiTrust results in Primo in order to eliminate the problem of "false positives." In other words, Primo would report that all of the HathiTrust results it returned were available online as full text, when many were not. We would like your feedback about the impact of this change from your perspective.

Q2 Which of the following statements best describes your opinion about the impact of hathiGenius?

- I haven't noticed a difference.
- I feel that Search It's presentation of HathiTrust results in Search It has become noticeably better since hathiGenius was implemented.
- I feel that Search It's presentation of HathiTrust results in Search It has become noticeably worse since hathiGenius was implemented.
- I have noticed a difference, but I feel that Search It's presentation of HathiTrust results is about the same quality as it was before hathiGenius was implemented.
- No opinion.

Q3 Please share any comments you have about hathiGenius or any ideas you have for improving the display of HathiTrust's records in Search It.

https://doi.org/10.6017/ital.v38i1.10574

REFERENCES

[1] HathiTrust Digital Library, "Welcome to HathiTrust!" accessed March 4, 2018, https://www.hathitrust.org/about.

[2] Sunshine Carter and Stacie Traill, "Essential Skills and Knowledge for Troubleshooting E-Resources Access Issues in a Web-Scale Discovery Environment," *Journal of Electronic Resources Librarianship* 29, no. 1 (2017): 7, https://doi.org/10.1080/1941126X.2017.1270096.

[3] Jacquie Samples and Ciara Healy, "Making It Look Easy: Maintaining the Magic of Access," *Serials Review* 40, no. 2 (2014): 114, https://doi.org/10.1080/00987913.2014.929483.

[4] Maria Collins and William T. Murray, "SEESAU: University of Georgia's Electronic Journal Verification System," *Serials Review* 35, no. 2 (2009): 80, https://doi.org/10.1080/00987913.2009.10765216.

[5] Karen Calhoun, Diane Cellentani, and OCLC, eds., *Online Catalogs: What Users and Librarians Want: An OCLC Report* (Dublin, Ohio: OCLC, 2009): 20, https://www.oclc.org/content/dam/oclc/reports/onlinecatalogs/fullreport.pdf.

[6] Rice Majors, "Comparative User Experiences of Next-Generation Catalogue Interfaces," *Library Trends; Baltimore* 61, no. 1 (Summer 2012): 191, https://scholarcommons.scu.edu/cgi/viewcontent.cgi?article=1132&context=library.

[7] Marlena Barber, Christopher Holden, and Janet L. Mayo, "Customizing an Open Source Discovery Layer at East Carolina University Libraries "The Cataloger's Role in Developing a Replacement for a Traditional Online Catalog," *Library Resources & Technical Services* 60, no. 3 (July 2016): 184, https://journals.ala.org/index.php/lrts/article/view/6039; Benjamin Pennell and Jill Sexton, "Implementing a Real-Time Suggestion Service in a Library Discovery Layer," Code4Lib Journal, no. 10 (June 2010): 5, https://journal.code4lib.org/articles/3022.

[8] Jody Condit Fagan et al., "Usability Test Results for a Discovery Tool in an Academic Library," *Information Technology and Libraries* 31, no. 1 (March 2008): 99, https://doi.org/10.6017/ital.v31i1.1855.

[9] Dan Moore and Nathan Mealey, "Consortial-Based Customizations for New Primo UI," *The Code4Lib Journal*, no. 34 (October 25, 2016), http://journal.code4lib.org/articles/11948.

[10] Lesley M. Moyo, "Electronic Libraries and the Emergence of New Service Paradigms," *The Electronic Library*, 22, no. 3 (2004): 221, https://www.emeraldinsight.com/doi/full/10.1108/02640470410541615.

[11] Kelsey Brett, Ashley Lierman, and Cherie Turner, "Lessons Learned: A Primo Usability Study," *Information Technology and Libraries*, 35, no. 1 (March 2016): 20, https://ejournals.bc.edu/ojs/index.php/ital/article/view/8965.

[12] Fagan et al., "Usability Test Results for a Discovery Tool in an Academic Library," 84.