2018-05-30

Custom Discovery Interface at the University of Michigan Library

Varnum, Kenneth J.

http://hdl.handle.net/2027.42/143857
Custom Discovery Interface
University of Michigan

New Developments in Discovery
BTAA Library Conference
30 May 2018

Ken Varnum
@varnum
I’m going to spend a few minutes this afternoon talking about the U-M library’s search interface; and by talking in a bit of detail about the underpinnings, I hope to illustrate the direction that we think discovery is taking. The interface is “in the wild” and you’re welcome to try it out yourself. The URL is at the end of the presentation.

As Roger and Bill both highlighted this morning, we Big Ten libraries suffer from an embarrassment of riches. We are fortunate to have a lot of resources to share with our faculty, staff, students, and the world. And that makes the starting points more complicated. It is not enough to send people to the catalog, or to a LibGuide, or to the one relevant subject database. Most of us have all of these aplenty, and more — with different breadths and depths depending on our institutions strengths and focuses.

For Michigan, we focused on a finite, but challenging, set of goals, as we thought about updating our aging discovery environments.
Goals

Improve user experience for ALL users
Reduce instruction complexities
Modernize infrastructure / reduce technical debt
Focus on the interface, not the underlying data

I’m going to focus on the technology and project management parts. Ben has talked some about our user experience process.

Ben has already covered our user experience work; we started with user needs, validated our designs along the way as we built out the site, and have exhaustively tested the site.
Teaching the Interface

Coordinating with library instruction staff

It matters much less where you start; you get all the options all the time

Reducing the cognitive load throughout the redesign
Technical Debt

A balloon payment was due

Pay-down old technical debt (retire multiple discovery layers built ca 2010)

Setting ourselves up for the next big project

Reducing the complexity of what we maintain

Catalog based on what was VuFind when it was born, and morphed into our own thing, far from the trunk from which it sprang

Article search, database/journal finder, and bento box discovery based on Drupal 6 (and updated w/out significant change) to Drupal 7.

Our website is also in debt. But we’re simplifying the loan — refinancing, in a way — to make that project easier. Cleaning up a lot of functions that were shoehorned into the “website” but that maybe shouldn’t have been, with hindsight.

Building with a single architecture — all our searches work the same way (both for users, and behind the scenes). We can add functionality to all at once, while preserving the distinctions our users have told us they value.
Solr (5.5) on the back end

Our own code in the middle — “Pride” & “Prejudice” — that act as brokers:

- User queries from the front end passed on to Blacklight
- User information to show appropriate content & offerings
  - “Get This” (page, copy, deliver) for catalog items
  - Correct scope for Summon-delivered content
- Everything is returned to the front end as JSON objects

Extensible -- could add more indexes, or more services
Our New Infrastructure

Solr (5.5) on the back end

Our own code in the middle — “Pride” & “Prejudice” — that act as brokers:

- User queries from the front end passed on to Blacklight
- User information to show appropriate content & offerings
  - “Get This” (page, copy, deliver) for catalog items
  - Correct scope for Summon-delivered content
- Everything is returned to the front end as JSON objects

Extensible -- could add more indexes, or more services
Front End
- Responsive
- Accessible
- Keyboard
- Screen-reader
Delivery Options

- Options organized by generally fastest to slowest
- Options are sensitive to the kind of item (circulating or not, available or not) and the user’s status.
- Options are presented from the holdings list; when there’s a physical item that could be obtained, the link is present.

<table>
<thead>
<tr>
<th>Description</th>
<th>Status</th>
<th>Call Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>v.1</td>
<td>On shelf</td>
<td>PQ 2631 .R63 472 1934</td>
</tr>
</tbody>
</table>

How would you like to get this item?

- **Find it in the library**
  Self Service (Immediate access to a physical copy when the library is open)

- **Pick it up at the library**
  Library-to-Library (Expected availability 1-3 days)

- **Have it Delivered**
  Document Delivery (Expected availability 2-4 days)

- **Request to have a small portion scanned**
  Document Delivery (Expected availability 1-3 days)

Frequently asked questions
Building for the Long Term

We are trying to future-proof, or at least future-insure, our architecture.

In fact, we already talk to three Solr indices: catalog data (data is managed by Aleph ILS), all the other locally-managed content (data is managed largely in Drupal, but also LibGuides), and Articles (data is managed by Summon).

If we should change ILS systems, as long as we can get data out of it, we should be OK for the UI. If we were to change article discovery platforms, we could do the same. And if our CMS changed, as long as we can get the data into d, we’re happy.

AND…. If we change the front end to something else, the back end wouldn’t care.
Thank You

Ken Varnum
varnum@umich.edu

https://search.lib.umich.edu/

Code Available at https://github.com/mlibrary