Legacy No Longer:
Designing Sustainable Systems for Website Development

Dalziel, Karin; Dussault, Jessica; Tunink, Greg
University of Nebraska–Lincoln

Introduction

The Center for Digital Research in the Humanities[1] (CDRH) at the University of Nebraska–Lincoln is home to digital collections such as *The Walt Whitman Archive*, *The Willa Cather Archive*, *The Journals of Lewis and Clark*, and *O Say Can You See*[2]. These projects contain overlap between subjects, individuals, and locations, yet are siloed, and many are built in aging, unsupported technologies with no interoperability or common search. In order to address this, the Center has developed an API ("Henbit") as part of a modular software stack to index and display data and content.

Challenge

Over the past twenty years, the Center has created over 30,000 TEI files in addition to other data sets such as VRACore documents, spreadsheets, and databases. Sites showcase the content and metadata of these files using a variety of technologies, many of which are no longer maintained. In addition, some sites used commercial software which became unsustainable when costs went up, cementing a commitment to open source. This experience informed and reinforced our adopted design philosophy, which can be summed up as:

- Keep it simple, stable, and sustainable
- Embrace modularity by writing software for one purpose
- Avoid over-engineering solutions (i.e. graphical interfaces where command-line will do)
- Provide comprehensive documentation

The Center has been inspired to think bigger about what can be accomplished by including existing data in a new framework. An exciting next step is creating a site to search all Center data, find commonalities between projects, and read materials across sites for comprehensive research. This approach will also help solve accessibility issues of older project sites which do not meet modern requirements. As projects become unsustainable, the Center may retire them while keeping all content available.

While having one place to view and search the Center's data is important, it's also critical to allow the creation of independent sites which utilize unique organization and include special features requested by principal investigators for new and evolving projects. Quickly creating

[1] https://cdrh.unl.edu
[2] http://whitmanarchive.org, http://cather.unl.edu, https://lewisandclarkjournals.unl.edu, and http://earlywashingtondc.org

bare bones sites to view in-progress TEI is essential, as it allows metadata experts and PIs to refine their data and arguments. Such sites should be written for ease of maintenance, freeing future developer time to work on new projects rather than sustaining old ones.

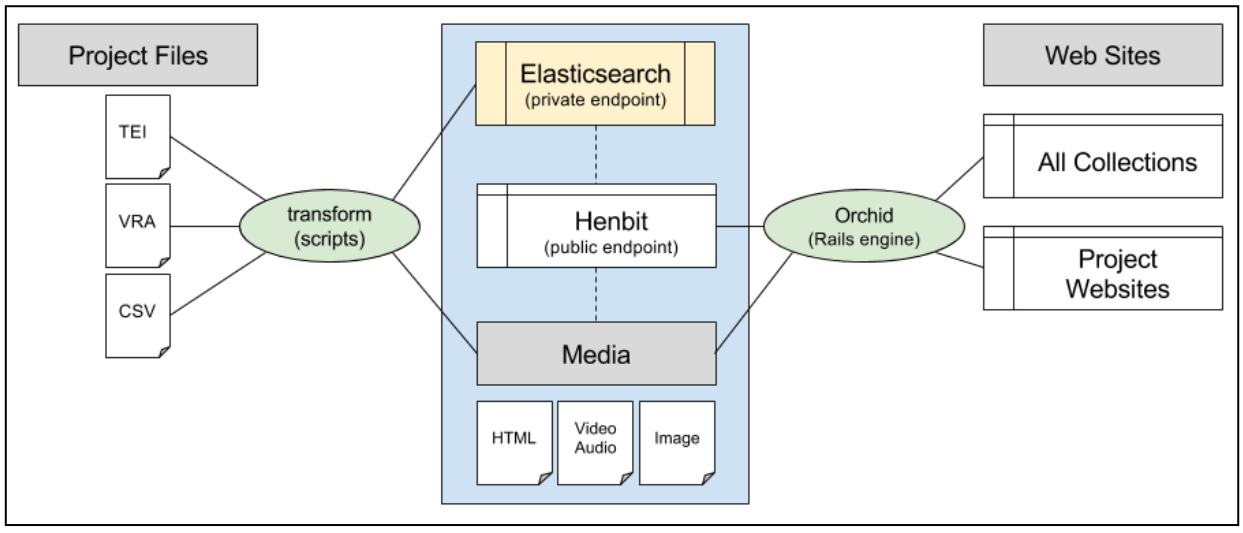
Solution

The Center explored the possibility of using existing software to address these challenges, such as XTF, Blacklight, and Fedora[3]. These packages did not fit the Center's needs; though comprehensive, they were not flexible enough to accommodate the variety of document types and project site requirements. Additionally, many solutions would lock the API into using Solr instead of allowing an interchangeable search engine (Blacklight, 2017; DuraSpace, 2017).

Instead of heavily customizing existing software, The Center decided to create a modular solution. The system consists of several components:

- data repository for project files and scripts for transformation
- document datastore and search engine (Elasticsearch)
- Ruby on Rails (Rails) API to serve data (Henbit)
- media retrieval system for associated images, audio, and video
- template generator for rapid website creation (Orchid)

With a modular software stack, future changes in technology and project needs can be accommodated with independent upgrades rather than massive redesigns and rewrites.



Project Files and Scripts

The data repository[4] houses original files for projects, such as TEI-XML, VRACore, CSV, and Dublin Core. The repository also contains CLI scripts which create HTML and populate

[3] https://xtf.cdlib.org, http://projectblacklight.org, and http://fedorarepository.org
[4] https://github.com/CDRH/data

search indexes with document content and metadata (CDRH, 2017a). New projects use generalized scripts, which are organized to allow overriding functionality in individual projects. Older websites may continue to use existing XSLT and populate legacy Solr indexes while their existing sites are supported, as well as populate Elasticsearch using the standardized script. Static HTML files derived from this process are used to create a document which can be viewed in a browser, regardless of the original data format.

Henbit (Public Endpoint)

Henbit[5] is a Rails powered API (application program interface) which creates appropriate requests for the backend index, and returns JSON. Currently, Henbit uses Elasticsearch as a backend, but most of its features (sorting, filtering, aggregating on ranges, etc) could be ported to a different backend. The OpenAPI specification[6] was used during Henbit's creation to fit current design practices (CDRH, 2017b).

Media Retrieval

In legacy sites, associated media lived inside the website directory. The Center has created a standard URL path for media files. It will be easier to optimize serving specific file types with this common retrieval structure. In the near future, the CDRH will be implementing a IIIF[7] image server to serve images of varying sizes and resolutions.

Orchid (Rails Engine)

Orchid[8] is a Rails engine which connects Rails 5 applications and Henbit. Orchid and a supporting gem, api_bridge[9], provide a template website that allows users to browse, search, filter, and view documents. This template is highly customizable, and can be altered to allow different URLs, search behavior, and anything possible in Rails (CDRH, 2017c).

Current Implementation and Future Plans

Beta versions of all components were released in 2017. In late 2017 the framework was used to build *The Complete Letters of Willa Cather*[10] (launched January 2018). *The Complete Letters* demonstrates the customization which can be accomplished with this modular system. The CDRH is currently developing another project, *Family Letters*, which will also take advantage of the data repositories, scripts, Henbit, and Orchid template.

In the meantime, older websites are being converted for the new system. Updated documents and original XSLT have been reorganized into the structure required by the data repository scripts and are being posted to the Elasticsearch index. Once a site for Centerwide

[5] https://github.com/CDRH/api
[6] https://github.com/OAI/OpenAPI-Specification
[7] http://iiif.io
[8] https://github.com/CDRH/orchid
[9] https://github.com/CDRH/api_bridge
[10] http://cather.unl.edu/letters

projects has been created, older sites can be retired as needed, replaced by content now available through the new API and supporting website.

The decision to use custom built software rather than an existing, out of the box solution, was not easy. Though at times it felt like reinventing the wheel, our highly customizable and flexible implementation prepares for future technological developments and enables flexibility in meeting project requirements.

References

Blacklight (2017). "Project Blacklight." http://projectblacklight.org.

CDRH (2017a). "CDRH Data Repository." *GitHub.* https://github.com/CDRH/data.

CDRH (2017b). "Henbit." *GitHub.* https://github.com/CDRH/api.

CDRH (2017c). "Orchid." *GitHub.* https://github.com/CDRH/orchid.

DuraSpace (2017). "Fedora Repository." http://fedorarepository.org.