

The Birth of LDbase

Lessons Learned from Designing a Discipline-Specific Data Repository

Bryan J. Brown

- Repository Developer @ FSU Libraries
 - ~6 years (since 2014)
 - MLS + MIS from Indiana University
 - Main projects are DigiNole + LDbase
- Previous USM presentations:
 - IR Day 2016: <http://aquila.usm.edu/irday/2016/1/7>
 - SMIRC 2018: <http://aquila.usm.edu/smirc/2018/1/6>
- Get in touch if you have questions!
 - bjbrown@fsu.edu
 - @bryjbrown on Twitter & GitHub



Designing a Repository

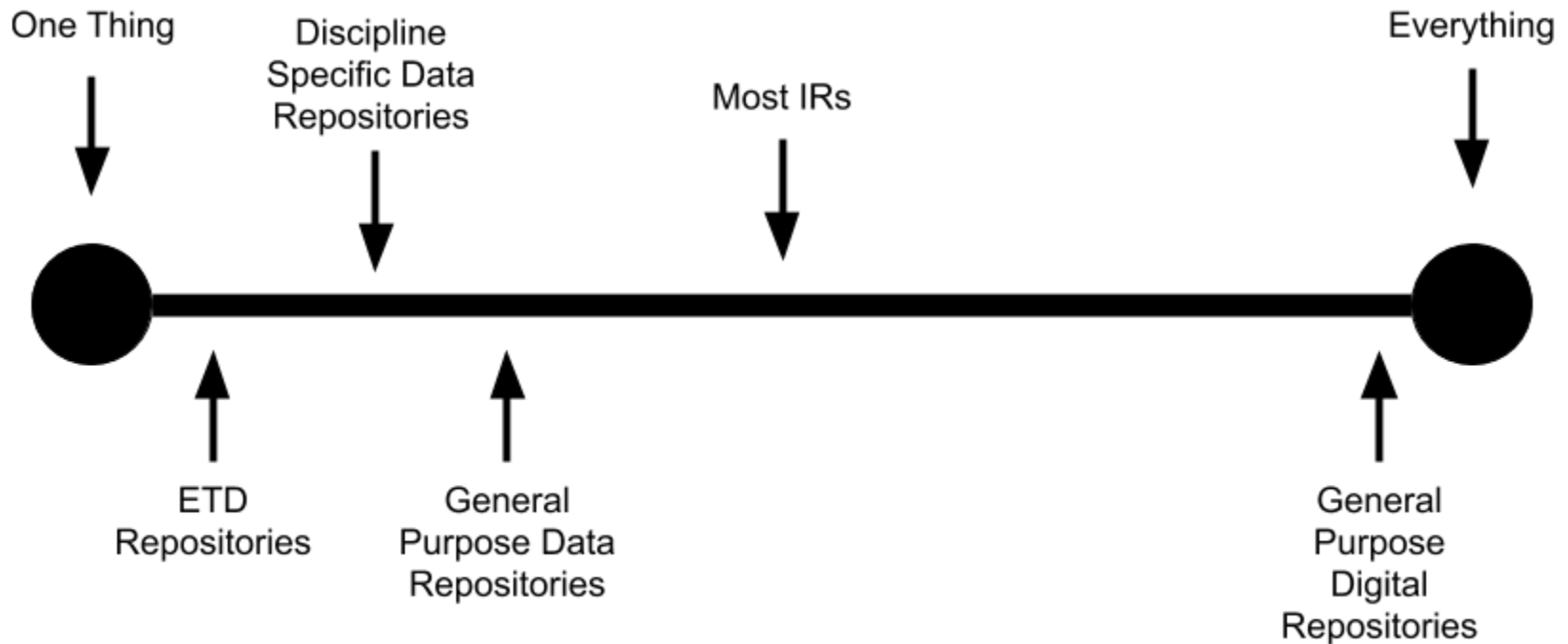
Designing a Repository

- What is a digital repository?
 - Software that enables preservation and access control on digital assets
- What is an institutional repository?
 - A digital repository designed for academic assets
- What are academic assets?
 - That's a great question...
- What's with all the questions?
 - Defining our vocabulary (words matter)
 - Defining our scope (what types of assets are we concerned with)

Scope Concerns

- What types of assets do we want to store?
 - Academic output?
 - Data?
 - Textbooks?
 - Student work?
 - ETDs?
 - Honors theses?
 - Undergraduate research?
- Who's assets do we want to store?
 - What communities are you serving?
 - Just your university?
 - The public in general?

Scope is a Spectrum



Designing DigiNole

History of DigiNole

- Islandora 7.x instance
 - Started as a Digital Library (2011)
 - Grew into an IR (2016)
 - Growing into a data repository (2020)
- Questions that have arisen:
 - Should authorities be part of an IR, or should they be in another system?
 - Should a data repository be part of an IR, or is it a separate thing?
 - Should a data repository be a workbench, or a graveyard?

Scope of DigiNole

- Philosophy: Everything FSU under one roof
 - Pros: One interface for everything
 - Users don't have to access multiple systems for different types of content
 - One search reveals all (serendipity!)
 - Cheaper for IT to host/maintain
 - Cons: One interface for everything
 - Thoroughly unoptimized search facets
 - Collision of different descriptive practices
 - Lots of accurate but irrelevant results
 - More stakeholders to consult before making changes

Designing LDbase

History of LDbase

- “Community repository for learning disabilities data”
 - “Data” = data repository
 - “Learning disabilities” = discipline specific
 - “Community” = additional social/sharing features, open to the public, emphasis on open data
- Funded by NIH grant received in early 2019
- Developed by a team of psychology faculty (domain experts) + library developers (repository infrastructure experts)
- Still being developed, shooting for public beta in early 2021

Developing LDbase

- February 2019 - August 2019 spent doing lots of planning / prototyping:
 - Establishing expectations
 - Creating a development timeline
 - Deciding on scope
 - Discovering & overcoming vocabulary hurdles
 - Creating use cases / user requirements
 - Creating user personas with different goals
 - Planning for required infrastructure
 - Building a prototype for internal testing

Developing LDbase

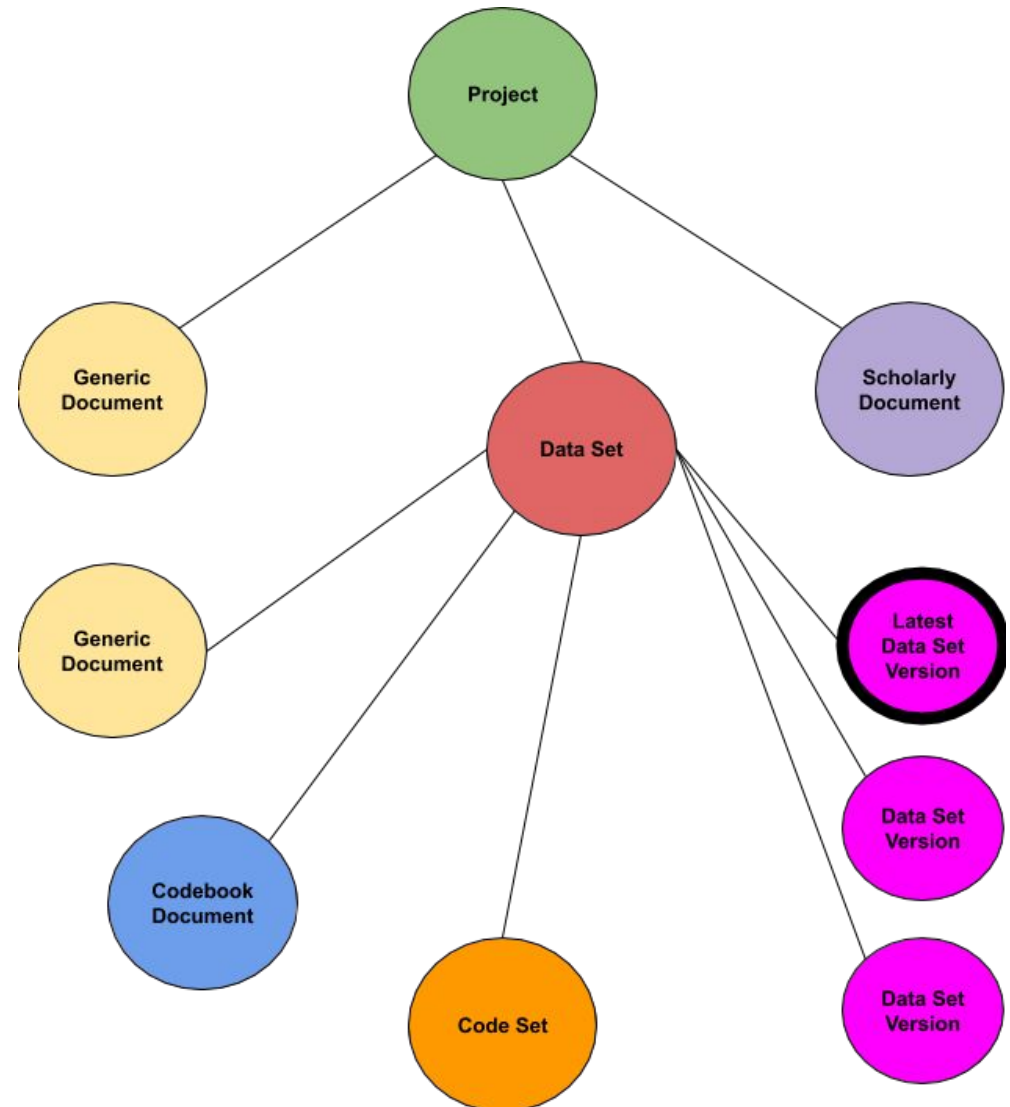
- February 2019 - August 2019 spent doing lots of planning / prototyping:
 - Establishing expectations
 - Creating a development timeline
 - Deciding on scope
 - Discovering & overcoming vocabulary hurdles
 - Creating use cases / user requirements
 - Creating user personas with different goals
 - Planning for required infrastructure
 - Building a prototype for internal testing

Vocabulary Issues

- “Metadata”
 - Asset description to the library developers
 - “codebooks” to the psychology faculty
 - Codebook = document explaining variable names and types in study data
- “Data repository”: workbench or graveyard?
- “Objects”:
 - A generic unit of content to library developers
 - Utter nonsense to psychology faculty
 - Just refer to them by the content type
- Pretty much all of the metadata fields needed to be explained to the library developers

LDbase Scope

- 3 content types:
 - Datasets
 - Versions
 - Code
 - Documents
 - Codebook
 - Scholarly
 - Generic
- 3 authority types:
 - Projects
 - People
 - Organizations



Lessons Learned

Narrow Scope = Good

- Restricting scope buys you interface optimizations
 - Navigation is easier
 - Searching is easier
 - Submission is easier
 - Managing content is easier
 - Migration will be easier
- Narrowly scoped repositories have less options, hence less display clutter
 - Users prefer simple / clean interfaces, even if they are less powerful
 - Think of Google vs OPAC

Build for Users

- Organize & set scope based on your target users
 - Library developers wanted lots of non-hierarchical generic “objects”
 - Psychology faculty wanted hierarchical named objects organized by project
 - This repository is for psychologists, not developers.
 - At some point even the psychology faculty have become steeped in LDbase development jargon, so external user testing is needed too.

When in Rome

- Learn the domain jargon of the community you are serving, and use it exclusively
 - Planning went much faster when we used psychology terms instead of librarian / developer terms in meetings
 - Using psychology terminology outside of meetings during development sprints solidified it as the standard
 - Once the developers started making an effort to use psychology terminology, the psychology faculty seemed to adopt developer terminology as well.

Takeaways

- When you are building a new repository...
 - Think from your users' perspective
 - Organize the way they would browse
 - Use labels that make sense to them
 - Optimize the interface for their tasks
 - Do usability testing as often as possible
 - Keep it as simple as possible (but no simpler)
 - Clean and simple > Powerful but cluttered
 - Be intentional about scope
 - Have an “elevator pitch” for your platform
 - How do new types of content affect it?

Questions?

Thank you!



BY