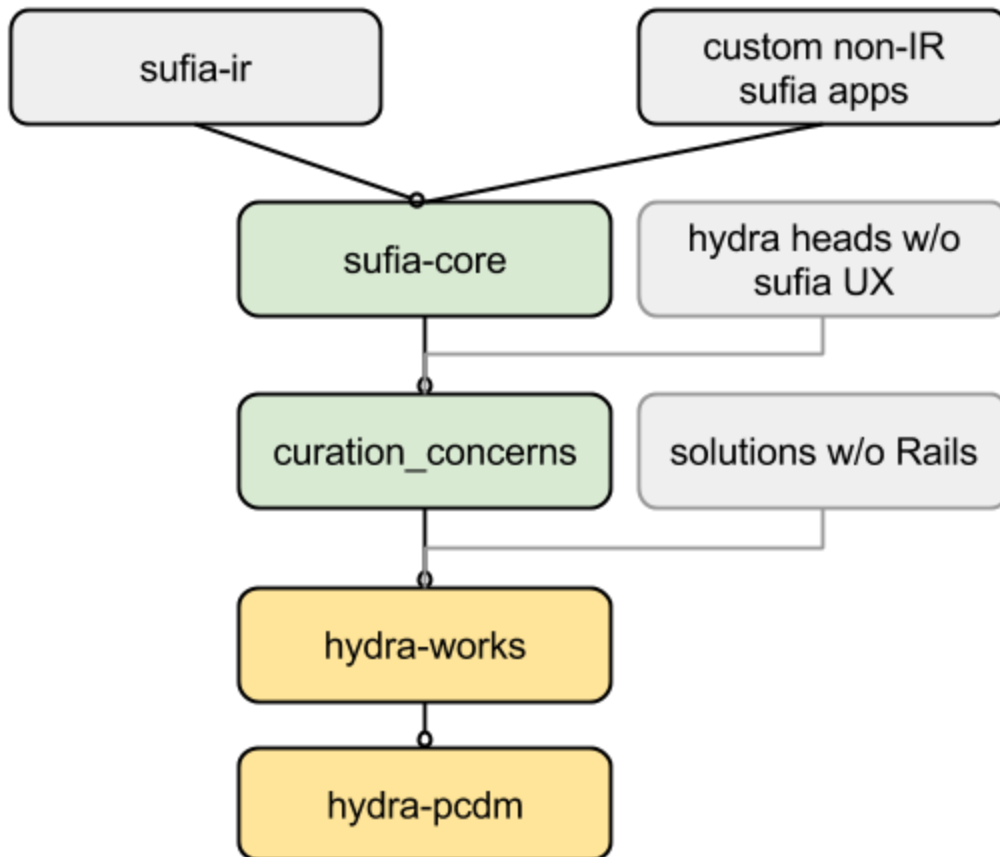


CurationConcerns: Breaking up Sufia and Worthwhile

Overview

Proposed End-Goal

We are implementing PCDM and breaking up existing functionality across new gems.



- **Sufia-IR** will be a rails engine that relies on **sufia-core**. It extends the functionality of **sufia-core** to provide a pre-fabricated, customizable Institutional Repository. It's basically the small portion of Sufia that will not be extracted into **sufia-core**.
- **sufia-core** (extracted from Sufia and Worthwhile/Curate) will be a rails engine that relies on **CurationConcerns**. It provides all of the functionality from Sufia that is not unique to the IR use case.
- **CurationConcerns** (extracted from Worthwhile/Curate) will be a Rails engine that relies on **Hydra::Works** and provides the models, controllers, views, routes and generators for creating and managing Work, File and Collection objects.

- **Hydra::Works** supports creation of PCDM-based Works, Files and Collections according to the [Hydra::Works PCDM Diagram](#). Note that in Hydra::Works, a GenericFile is not a PCDM File. Instead it is a PCDM Object that contains PCDM Files.
- **Hydra::PCDM** provides the ability to create PCDM Objects, Files and Collections

Differentiating between the Gems

Guidelines for deciding where code belongs.

If it ...	... then it goes in
Implements <i>only</i> the PCDM model	hydra-pcdm
Implements the Hydra::Works PCDM Model AND Does not depend on Rails	hydra-works
Implements the Hydra::Works Model AND Depends on Rails AND Does not depend on Blacklight* AND Does not make assumptions about your overall application (layout, access controls, etc)	curation_concerns & curation_concerns-models
Depends on Blacklight OR Makes assumptions about your overall application (layout, access controls, etc) AND Does not assume that your application is an Institutional Repository	sufia-core & sufia-core-models
Makes your application behave (specifically) like an Institutional Repository	sufia-ir

* We tried removing this dependency, but hydra-collections relies on Blacklight and there is still some Blacklight-related code in Curation Concerns for retrieving works and collections.

See [Breakdown of Features to Extract](#) and [Code Shredding: Sufia, Worthwhile and Hydra-Works](#) to determine where, based on the above guideline, basic support for the following features belongs:

- Characterizing files
- Basic support for Generating Derivatives of a File
- Configuration for Generating Derivatives (ie. dimensions for thumbnails)
- Virus Checking
- Full Text Indexing
- File Upload

- Proxy Deposit
- System Statistics
- “Featured Works”
- Breadcrumbs
- UI for Adding Works to a Collection
- UI for Viewing a Collection and searching through its members
- Sufia Dashboard

CurationConcerns

What is a Curation Concern?

The Curate gem (and the Worthwhile gem, which is a fork of the Curate code) has a very useful notion of a CurationConcern. Any type of PCDM object (ie. GenericWork, Dissertation, ScannedBook, or even GenericFile) can be treated as a CurationConcern. Based on this pattern, we can provide default Rails Controllers, Views, Helpers, and Routes for performing CRUD operations on those CurationConcern objects. We can also provide a scaffold generator that gives you a starting point for customizing the workflow and views associated with each CurationConcern type.

We will be extracting this functionality into a [curation-concerns](#) gem. It will be an engine that *extends an existing Rails application*, adding support for creating and managing CurationConcerns. It does not provide things like users, application layout, access controls, search, etc. because it assumes that the host application will provide them.

See the diagrams in [CurationConcerns in Worthwhile and in CurationConcerns](#)

Example: Dissertations

I’m building an application that handles Dissertation objects. In the future, I will also be adding ScannedBook objects and GISDataset objects. I want my Dissertations to have Hydra::Works::WorkBehaviors. I want the submission workflow to be relatively similar between all of the Work types, I want the process of uploading files into a Works to be the same across Work types, and I want to be able to associate Works with Hydra::Works::Collections.

However, I want to customize the workflow for creating & editing Dissertations and I also want to customize how they appear in search results.

The CurationConcerns engine makes it easy for me to do this. It provides:

- Convenience methods for manipulating works
- Methods registering which types of CurationConcern that your application cares about
- Helper methods for listing links to create CurationConcerns
- Routing for CurationConcerns (automatically includes all registered CurationConcern types)
- Search logic for including the desired CurationConcern types in search results

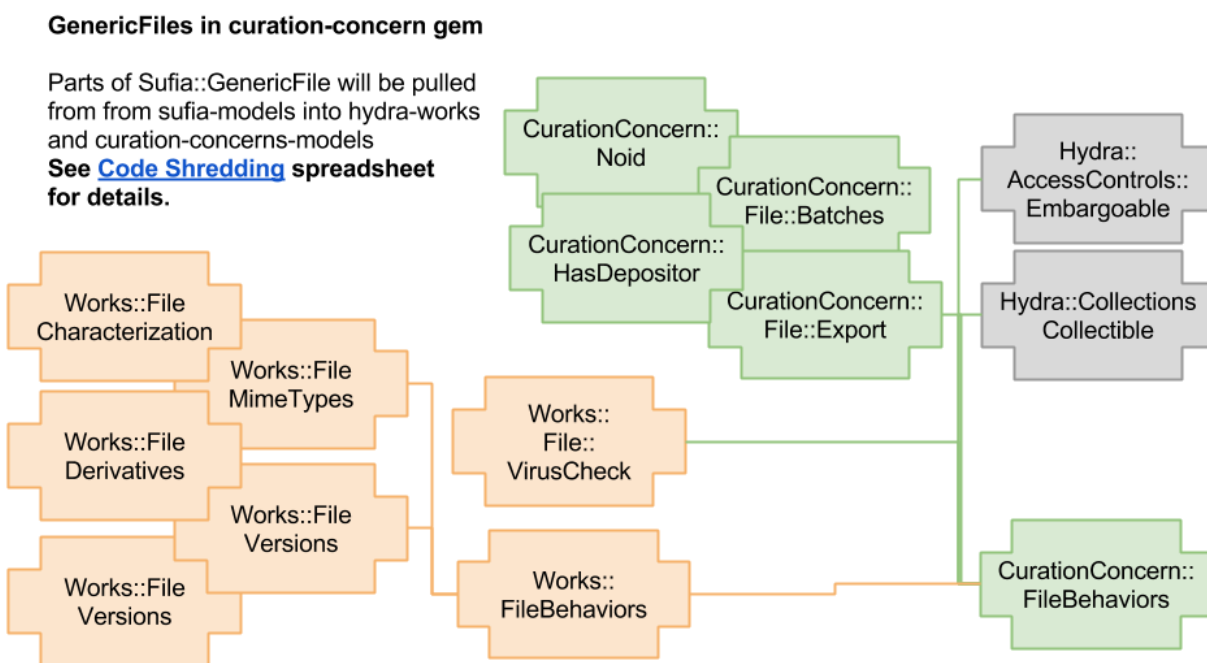
- Default views (inherited from the engine) for CRUDing CurationConcern objects

Using the `curation_concern:scaffold` generator, you will get

- Update to `config/initializers/curation_concerns.rb` so that Dissertation is included in your registered CurationConcern types (so it's included in routing, etc.)
- Dissertation model, which includes `CurationConcern::WorkBehaviors`
- `DissertationsController`, which includes baseline CurationConcern Controller Behaviors
- View partial for customizing how a Dissertation appears in search results (the rest rely on defaults from the engine)
- RSpec tests which, by default, ensure that your Dissertation objects behave like CurationConcerns

GenericFiles in Hydra::Works, CurationConcerns and Sufia

This is from a slide in the deck titled [CurationConcerns in Worthwhile and in CurationConcerns](#). The specific, file-by-file breakdown of where the code will go is included in [Code Shredding: Sufia, Worthwhile and Hydra-Works](#) (referred to by “Code Shredding” link in the diagram).



Reference/Links

Modeling of [CurationConcerns in Worthwhile and in CurationConcerns](#)

Sufia-Worthwhile Alignment Factors

[Breakdown of Features to Extract](#) from Sufia and Worthwhile -- where the features are, where they should go.

[Code Shredding: Sufia, Worthwhile and Hydra-Works](#) all of the Classes and Modules defined in Sufia and Worthwhile (and some relevant related stuff). Where they should be moved to, what they should be named, etc.

Proposed Work Plan

Note: *Phase 1 is the Sufia-PCDM sprint (already in progress May 4 -15). Phase 2 and Phase 3 could each be done with a small team, and might be easier to complete with a small team -- they do not require another big, coordinated group sprint.*

Phase 1: Sufia with Hydra::Works & Hydra::PCDM

Implement Hydra::PCDM and Hydra::Works. Refactor Sufia to rely on Hydra::Works, so that Sufia is able to create GenericWork, GenericFile and Collection content that is persisted using PCDM and the Hydra::Works model reflected in the [Hydra::Works PCDM Diagram](#).

1. Implement **hydra-pcdm** and **hydra-works** to support creation of PCDM-based Works according to the [Hydra::Works PCDM Diagram](#)
2. Refactor sufia to rely on **hydra-works**
3. As a stopgap measure, copy any necessary CurationConcern functionality from **Worthwhile** into Sufia so that Sufia can have GenericWorks. *This code will be deleted from Sufia after hydra-curation-concerns gem is implemented (see Phase 2)*

Note: Some of the Sufia::GenericFile functionality will ultimately reside in Hydra::Works. (see [Code Shredding: Sufia, Worthwhile and Hydra-Works](#)) If necessary, move that functionality into hydra-works now. Otherwise, wait until the next sprint.

Phase 2: CurationConcerns

Create a gem that implements the [CurationConcern pattern](#) (originally from Curate). Make it rely on hydra-works, so that it creates Work, File and Collection objects according to the Hydra::Works model. Refactor Sufia to use it, throwing away redundant code from Sufia. In the process, pull GenericFile functionality out of Sufia into CurationConcerns and Hydra::Works so that it can be used & maintained more broadly.

1. Extract the *CurationConcern* Engine code (models, controllers, views, routes and generators) from Worthwhile (originally from Curate) and put it in a **hydra-curation-concerns** gem.

2. Extract *Sufia::GenericFile* and the associated code (characterization, derivatives, etc.) out of the *sufia* and *sufia-models* gems, splitting the code between **hydra-works** and **hydra-curation-concerns** (See [Diagram: GenericFiles in Hydra::Works](#), [CurationConcerns and Sufia](#) and [Code Shredding: Sufia, Worthwhile and Hydra-Works](#)).
3. Refactor **sufia** and **sufia-models** to get *GenericFile* and *GenericWork* functionality from **hydra-curation-concerns** (and **hydra-works**) rather than having that code repeated in *sufia*.
4. Remove redundant placeholder Works/CurationConcern code that was added to Sufia in Sprint 1

Phase 3: **sufia-core**

Merge the relevant Engine functionality from Sufia and Curate/Worthwhile (mostly Sufia) into a Rails Engine that generically supports building full-featured Hydra-based Applications for creating & managing CurationConcerns. See [Breakdown of Features to Extract](#) from Sufia and Worthwhile, which outlines where the features are & where they should go.

1. Fork *sufia*, rename it to **sufia-core**, and remove any code/functionality/assumptions that are unique to the IR use case
2. Refactor **sufia** to rely on **sufia-core**, so that all *sufia* does is add the functionality (workflows, layout & business logic) that is unique to the IR use case