Haplo Repository

Open Source Plugins

Haplo Repository is a modern academic repository designed for researchers and repository teams to work together.

Developed in a highly modular manner, institutions can pick the functionality they need, integrate with institutional systems and the wider academic infrastructure, and provide a highly customised public repository interface to showcase the institution's research.

Key out-of-the-box features

- Flexible schema, allowing multiple types of outputs (including research data) to be managed in the same repository
- Support for practise based research
- REF compliance monitoring and reporting
- Flexible ingest processes with excellent UX, delivering high levels of self-deposit from researchers
- "Request a copy" workflow from internal and external researchers, including file preparation stages for research data (eg anonymisation).
- ORCiD integration
- DOI minting
- Impact recording
- Research web profiles
- Multiple per-file embargo support
- Standard protocol and metadata support, including OAI-PMH
- Metadata import and compatibility with legacy EPrints repositories
- Flexible toolkit for customised public interfaces to the repository

Getting started

You'll need the Haplo Repository plugins
https://github.com/haplo-org/haplo-repository

which run on top of the Haplo Platform
https://github.com/haplo-org/haplo

To try it out quickly with Docker, follow the instructions in the README file. Or contact hello@haplo.com to request a demo.

All the code is available on GitHub under the Mozilla Public License v2.

Haplo Repository Plugins

Haplo Repository's highly modular nature is delivered through multiple co-operating plugins. Institutions choose the functionality they need by installing the features they need, and making any per-institutional schema changes.

Haplo's mature plugin API https://docs.haplo.org/dev/plugin guarantees that plugins will work perfectly together, by providing mediated communication between them, hooks for plugins to apply institutional policy, and a user interface that seamlessly combines plugin functionality into a consistent and coherent whole.

Each of the plugins offers an opportunity to swap out standard functionality with institution specific implementations, allowing a high level of customisation.

A "demonstration" repository is provided as a starting point for an institutional repository, which will work out-of-the box for most institutions requiring a standalone repository.

Schema

hres_repository
Essential Haplo Repository functionality. This contains essential shared schema, reporting, and permissions required for any installation of the Haplo Repository

hres_repo_datacite
Support for the Datacite metadata schema, and translation to present data stored in Haplo in this format, suitable for presentation through machine interfaces such as OAI-PMH.

hres_repo_eprints
Translates to and from a typical configuration of EPrints EP3 XML. This allows for simple migration from a legacy EPrints repository, as well as ensures Haplo presents metadata to the world in a backwards-compatible way.

hres_repo_ethos
Adds a translation to the uketd_dc metadata format so that Theses in the repository can be harvested over OAI-PMH by the British Library's EThOS service.

hres_repo_rioxx
Provides a mapping to expose data in the RIOXX metadata format to external systems, and to report to repository editors whether the metadata on an item meets the RIOXX quality standards.

hres_repo_schema_collections
Adds the schema for collections of outputs, and UI to aid users in grouping together outputs already in the repository into a collection.

hres_repo_schema_research_data
Schema for Research Data output types. The schema added by this plugin is required if the repository is to include research data.

hres_repo_schema_data_recommended
Additional schema for research data types, which is optional but recommended.

hres_repo_schema_outputs
Schema for traditional research output types, such as Journal Articles and Books, as well as non-traditional outputs such as Exhibitions, Artefacts, and Software. This plugin contains the minimal requires schema if the repository is to be used for research outputs.

hres_repo_schema_outputs_recommended
This plugin adds additional, recommended, schema for research outputs. It could be replaced if an alternative schema is desired. The schema added by this plugin includes sophisticated support for practice-based research types, and for accreditation of different kinds of author contributions (eg. Lyricist, Director, Producer. etc).

hres_repo_schema_recommended
Recommended but optional schema that can be used for all research output types, including datasets and collections.

hres_repo_metadata_dc
Produces a mapping from Haplo's schema to the Dublin Core schema, and produces XML suitable for exposing to other interfaces, such as through OAI-PMH.

hres_author_citation_value
Custom data type and user interface for recording author citations, linking to the author record to ensure consistency and accurate reporting, while allowing custom citation formats and per-output citations.

User interface

hres_repo_navigation
UI to promote browsing and discovery of outputs in the internal interface. Providing filterable lists of outputs in a department, research group, faculty, or by a given researcher.

hres_bibliographic_reference
Implements the default repository citation style, forming consistent citations for output records from the metadata recorded within the system.

hres_repo_duplicate
Adds the option for administrative staff to copy duplicate an existing repository item, copying over metadata onto a new record to reduce the amount of manual data entry for related items (eg. research presented at a conference and also written up as a Journal Article).

hres_repo_harvest_claim
Contains the implementation for the internal processes, UI, and management of claiming items harvested from external sources. Provides tasks and notifications sent to authors, and the workUnits and UI for managing updates and edits to records.

hres_repo_harvest_sources
Provides infrastructure for registering integrations as external sources of output metadata, collecting information from those configured sources (each source integration implemented as a separate plugin) and matching harvested outputs to records already within the system.

hres_repo_impact
Draft implementation of support for Impact and Evidence of Impact records within Haplo Repository. Under development, with the schema based on the recommendations from CASRAI's working group - https://forum.casrai.org/t/june-2017-impacts-data-collection-review-now-closed/383

hres_repo_impact_factor
Adds the Impact Factor attribute to Journal objects, and repots to allow institutions to track the Impact Factor scores of their research

hres_repo_impact_ui
Default UI for the recording of Impact and Impact evidence in Haplo. Aim is to promote the recording of evidence for Impact through UX guiding researchers to that option.

hres_repo_ingest_start_ui
The default UI options for the start of the researcher self-deposit process. Includes inbuilt guidance for the depositing author, aiming to guide them to input the best quality data possible. Pluggable so universities can have different UI if they so wish.

hres_repo_ingest_workflow
The default ingest workflow for the mediated self-deposit process.

hres_repo_object_editor
Implements a custom form for adding and editing research outputs, to include per-field guidance to depositing authors as they add their research.

hres_repo_list_management
Adds an and intuitive interface for repository editors to review and manage the controlled list values configured within the application. By default this covers Journals, Funders, and Publishers, but a configuration option is provided so this can be extended to other schema types.

Open Access and REF

hres_ref_schema
Schema related to the REF, including Units of Assessment. Reporting on REF.

hres_ref_repository
Support for REF Open Access compliance checking and exception recording. The plugin reports on each item, and also provides overall compliance dashboards so Repository Editors can monitor the REF readiness of the repository.

hres_repo_open_access
This plugin provides most of the schema for tracking the Open Access status of items in the repository (see hres_repo_oa_green_* plugins for more information), as well as reporting dashboards to allow repository editors to get an overview of the OA footprint of research at the institution.

hres_repo_oa_green_inferred
The first of two methods for recording an output as Green Open Access. If this plugin is installed the repository will assume that this repository is the primary system that publications would be uploaded to for a researcher. It therefore infers Green OA status based on whether the Accepted Author Manuscript is attached to the output record.

hres_repo_oa_green_stored
The second method of recording Green OA status is for cases where it's likely that AAMs are being uploaded to other repositories, such as a subject-specific repository, and so inferring Green OA status is not appropriate. This plugin adds an explicit metadata field to outputs where Green OA status can be recorded.

hres_repo_apc
Initial draft of recording and tracking Article Processing Charges within Haplo Repository. This is still under development, providing minimal features that we intend to expand on in partnership with our development institutions.

hres_ref_process_management
Overall management of a REF assessment process.

Researcher Profile

hres_researcher_profile
Framework and UI for managing a self-service researcher profile, gathering data from information stored elsewhere in the application, and self-service editing. Profile can be show in the internal user interface, and published in the public repository interface.

hres_researcher_profile_biography
Standard profile fields for researcher biography.

hres_researcher_profile_extended
Extended research profile fields.

hres_researcher_profile_photo
Photo upload and management in researcher profiles.

Access control

hres_file_mediated_access
Framework for managing access to sensitive files.

hres_repo_access_level_policy
Schema and policy to allow for restricting sensitive files uploaded to outputs, such as research data or items with personally identifiable information.

hres_repo_access_request_components
A set of reusable components for to support requests for access to non-public files held within the repository.

hres_repo_access_request_management
UI to enable requests for access to restricted files to be managed on output records.

hres_repo_embargoes
Implements embargoes for the Haplo repository. Multiple embargoes can be applied to an output, restricting each kind of file (eg. publisher's version vs accepted manuscript), allowing you to upload all available documents to Haplo securely.

Integrations

hres_oai_pmh
The framework for Haplo systems to act as an OAI-PMH data provider.

hres_oai_pmh_standalone
Provides a standalone OAI-PMH endpoint, for repositories without a human-usable public interface to act as a 'dark archive'.

hres_digital_object_identifier
Datatype and schema support for DOIs.

hres_repo_doi_minting
Layer on top of the support for the DOI data type, that allows systems to be configured to automatically mint and update DOIs for new outputs.

hres_repo_doi_retrieve_for_new_object
Development implementation of an interface to harvest metadata from data stores using a DOI entered by the user. This version harvests the basic from the Datacite metadata store.

hres_repo_irus_uk
Sends tracking requests to IRUS-UK and IRUSdata-UK for file downloads from the repository.

hres_orcid
Full integration with ORCiD, including pushing publications to ORCiD. Schema and implementation of a custom ORCiD identifier data type.

hres_orcid_obtain
User interface to guide researchers to signing up with ORCiD, with reporting on institutional progress to full adoption.

hres_orcid_researcher_profile
Standard management of ORCiD on researcher profile.

Public repository interface

hres_repo_publication_common
Provides a set of common features and components for repository publications - the public interface to the repository - such as html metadata tags, search by fields functionality, components such as styled button links, and restricted file rendering. These can be flexibly reused in many different places in the public interface, or not at all, to allow it to be fully configurable for your needs.

hres_repo_publication_page_parts
“Page parts” are sections of a page in repository publications that can be reused between pages. For example, a “related items” page part lists outputs in the system by one of the authors of the item you’re viewing. This plugin provides a collection of useful page parts to put together to build a customised public interface.

hres_repo_web_profiles
Supports researchers having control over the display of their publications on their public web profile - allowing them to reorder their publications builds engagement with the system and so encourages deposits

Demonstration repository

These are plugins that bring together the repository components to demonstration the functionality, and as the basis for a customised deployment in an institution.

hresrepodemo_application
This sets up application-specific system configuration such as the homepage layout, any institution-specific roles and permissions, and lists the common plugins to be installed in this system as dependencies.

hresrepodemo_access_request
This is a demonstration implementation of the access request process for a restricted item. This could either be Research Data that has not been made available publicly, or research outputs containing potentially sensitive information.

hresrepodemo_repository_publication
This plugin uses the standard publication components in hres_repo_publication_common and hres_repo_publication_page_parts to produce a simple example public interface for the repository.

hresrepodemo_request_a_copy
This is a demonstration implementation of the Request a Copy process, intended for items that are currently under embargo. This uses many of the same components as the access request process described by hresrepodemo_access_request - reusing common components in this way allows sophisticated processes to be described accurately with very little effort in Haplo Repository systems.

https://www.haplo.com/repository
https://github.com/haplo-org/haplo-repository