Research Data Collections WG

Current status

- Are the definitions of the conceptual model finally consistent and understandable?
- Does the revised conceptual model now adequately cover different collection approaches from user communities?
- Do the API methods match user needs, are they complete and can they be implemented also with legacy collection solutions in place?

Envisioned next steps of the WG work:

- Finalize the conceptual model and align it with the API specification
- Fine-tune the API methods, also in view of existing collection implementations
- Finalize default set of collection properties and models
- Improve communication channels with user communities and potential early adopters

Links

1. The WG homepage: https://rd-alliance.org/groups/pid-collections-wg.html
2. The collection of WG documents is hosted at: https://datashare.rzg.mp.de/index.php/s/W878ggdygLmxzXD
3. The WG GoTo Meeting (occurs the second and last Tuesday of every month at 13:00 GMT (9:00 Eastern, 15:00 Central European): https://global.gotomeeting.com/join/997874677
4. A model of the API can be found under: http://rdacollectionswg.github.io/apidocs/#/
5. Discover demo implementations of the API: http://dft-rda.esc.rzg.mp.de/reptor

Come and join us!

The group welcomes any input from user communities, particularly on the API and the concrete implementations.

Background

Several communities have expressed interest in leveraging aggregations of objects with particular focus on building such aggregations through PIDs and possibly providing identifiers for aggregation objects. There is however no unified cross-community approach to building and managing such collections and no common model for understanding them. Building collections within diverse domains and then sharing or expanding them across disciplines should enable common tools for end-users and e-infrastructure providers. Individual disciplinary communities can directly benefit if such tools are made widely available, and cross-community data sharing can benefit from increased unification between collection models and implementations. PID providers may benefit from marketing additional services on collections. The WG is working in close collaboration with scientific communities to come to a pragmatic and usable solution.

Definition

A collection is a digital object which is identified by a PID and consists of a set of PIDs/Ids, a set of additional pointers/links and metadata together with each PID/Id. The interaction options and expected behavior of a collection are defined by capabilities, which are part of a collection’s metadata. A collection can be given explicitly by naming all PIDs/Ids directly or implicitly by naming all PIDs/Ids directly or implicitly by a generation rule. A collection is called finite if the set of PIDs/Ids, generated by iteratively resolving its “sub-collections”, is finite.

Source: http://smw-rda.esc.rzg.mp.de/index.php/Collection

Current status and recent activities

Since the last plenary in Denver, the group has finished the conceptual model of a collection API. Based on the data structure below, REST-Style functions were discussed and defined. Members of the WG are working on concrete implementations of the definitions to prove their feasibility. The group is making steady progress through regular twice-monthly virtual working meetings, which are open for all.

Major discussion questions currently open:
Basic Data Model for Collections API

The underlying data model of the collections API can be defined by three submodels:

**Service**: defines the basic functionality an instance of the collections API provides. An instance can host (locally or virtually) an arbitrary number of collections

**Collection**: properties of a specific collection at the current instance. A collection can have an arbitrary number of members, including (sub-)collections

**Member**: properties of a specific member of a collection

### Service (S)

- **providesCollectionPids**: `true`|`false`  PID minting during collection creation
- **collectionPidProviderType**: string (uri)  External service for PID minting
- **enforcesAccess**: `true`|`false`  Enforcing access control on collections/members
- **supportsPagination**: `true`|`false`  Support paging through result sets
- **asynchronousActions**: `true`|`false` Indicates if actions such as update, delete occur synchronously or may be queued for later action
- **ruleBasedGeneration**: `true`|`false` Indicates if the service allows rule-based generation of new collections
- **maxExpansionDepth**: int (0)  Maximum recursion depth supported by service
- **providesVersioning**: `true`|`false`  Service provides versioned backend
- **supportedCollectionOperations**: [string]  List of supported /ops endpoints
- **supportedModelTypes**: [string (uri)]  List of supported collection models

### Collection (C)

- **id**: string (ideally pid)  Members in collection are indexed
- **Capabilities**
  - **isOrdered**: `true`|`false`  Members can be added, updated or removed
  - **appendsToEnd**: `true`|`false`  Appends or prepends newly added members
  - **supportsRoles**: `true`|`false`  Collection members have roles
  - **membershipIsMutable**: `true`|`false`  Members can be added, updated or removed
  - **metadataIsMutable**: `true`|`false`  Collection objects can be edited
  - **restrictedToType**: string  Constraints members to datatypes
  - **maxLength**: int (-1 = unlimited)  Limit collections size
- **Properties**
  - **ownership**: string (ideally pid)  Entity that holds rights to collection
  - **license**: string (uri)  License model
  - **modelType**: string (uri)  Collection model. Expected to use a controlled vocabulary, or PID of registered data type
  - **hasAccessRestrictions**: `true`|`false`  Access to collection members is restricted
  - **memberOf**: []  Memberships in other collections
  - **descriptionOntology**: string (namespace)  Ontology used for descriptive metadata. URI of a controlled vocabulary
  - **description**: {string: string}  Clear text description of collection

### Member (M)

- **id**: string (ideally pid)  Preferably a PID
- **location**: string (uri)  Link to data object
- **datatype**: string (uri)  Preferably defined in Data Type Registry
- **ontology**: string (uri)  URI of an ontology model class that applies to this item
- **mappings**: string (uri)  As defined in collection capabilities
- **role**: string  Role in the collection. Controlled Vocabulary term expected
- **index**: int  Position in ordered collection
- **dateAdded**: ISO datetime  Time that item has been added to collection