BoF PID Collections: Introduction

Bridget Almas (Perseus Digital Library / Tufts)
Tobias Weigel (DKRZ / University of Hamburg)
Frederik Baumgardt (Perseus Digital Library / Tufts)
Tom Zastrow (MPCDF)
Agenda for today

- Introduction
- Use Case overview
  - DKRZ
  - Perseus DL / Open Philology Project, U Leipzig
  - PECE
  - BCO-DMO
  - Ocean Data Interoperability Platform
  - Hathi Trust / SEAD project
  - Harvards Astronomy Abstract Service (?)
  - ... and we are open for more use cases!
- Discussion, possible adoption pathways and other relevant efforts
Current status

- Maturing case statement
  - Comments from TAB and OAB
  - Additional comments welcome
- Initial use case providers and possible adopters
- BoF at P6, head for endorsement after P6 and first WG meeting at P7
Collection models – what, why and how
- Extract common characteristics for understanding collections
- Devise a set of categories, constraints and requirements
- Collection life cycle
- Also include features like fragment identification and queries

Unified API for CRUD+L on collections
- Create, Read, Update, Delete, List
- Based upon the requirements
- Independent from a particular PID solution
- Implement a demonstrator
- Derive core PID Information Types
- Examples for added-value tools/services
- **WG PID Information Types**
  - Use of types and profiles to describe collections and membership

- **WG Type Registries, BoF Data Typing**
  - Use of the registry as an automation facilitator where appropriate
  - Do we have key collection data types?

- **WG Data Citation**
  - Citable collections: Requirements, constraints, impact on collection models
A bag of things with a price tag persistent identifier.
Motivation and larger context

- More automated object management
  - Improving scalability; object numbers and variety
  - Sharing objects and services across infrastructures

- PID(s) and collections as a Data Fabric component

from Larry Lannom / EUDAT

from the Data Fabric Paris document
What is a collection?

- A collection provides a way to bag objects together.
- A collection bears a PID.
- Items may bear PIDs.
- Collections may have a life cycle.
- Collections may be implemented in many ways.

API: I do not want to know how your collection is implemented, but I want to know how to work with it.

- Create one in your system. Add items. Iterate over them.
- Get elemental metadata (size, author, timestamps, version, status)
- Possibly modify and remove elements – if that is allowed. Is it?
The items in a collection may be of same type, but often it is a mix.
- Texts, diagrams, images, audio/video, metadata objects, ...
- The Data Typing WG should help in defining these.

Collections may also contain collections.
- What are the requirements?
- Is it possible for every collection model?
- BagIT
- OAI-ORE
- CITE Collection Services protocol
- CLARIN virtual collections
- HTTP / REST / URIs
- ...

- But there is no unified CRUD approach with PIDs as primary anchors and there is no comprehensive work on different collection models.
Some interesting questions...

- How many of you know what the difference between an array and a linked list is?
- Do you know what an abstract data type is?
Collections in a nutshell

- Draft primer document available from the session page

- What is a collection?
- What can you do with it?
- What are the technical requirements?
- How does all of this translate to the scope of user communities?

- Work in progress – comments appreciated.
  - This may perhaps become another deliverable.
How will this session continue?

- Next on the agenda: Use cases!
What

Application Layer

PID / Collection Layer

Data Access Layer
How

- ADS (abstract data structure)
- Interface (technical specification)
- Reference implementation
  - Centralization?
Complexity
- Basic set vs. more complex collection type
- Mutability (affects citability)

Ordering
- Set vs. Vector

Recursiveness
- Set/Vector -> Graph, Tree, Matrix, Map, Object

Heterogeneity
- allow different kinds of data types?
- provide information on data types?
- typed relations between PID and collection?
CRUD+LS
Query type information and other metadata
  generalized: map, reduce, zip, flatten, group, filter, sort
Hypermedia response
  enable type agnostic processing