Adoption of RDA DTR and PID in Deep Carbon Observatory Data Portal

Stephan Zednik, Xiaogang Ma, John Erickson, Patrick West, Peter Fox, & DCO-Data Science Team

Tetherless World Constellation
Rensselaer Polytechnic Institute
Outline

• Background & Research questions
  – RDA-DTR, RDA-PIT, DCO Data Portal

• Nature of efforts
  – Basic data type vs. Specific data type

• Approaches
  – Integration architecture vs. Self-contained architecture

• Initial results

• Future works
Background

• RDA - Data Type Registry (DTR) working group
  – Addressed a core issue of data interoperability: to parse, understand, and reuse data retrieved from others

• RDA - Persistent Identifier Information Types (PIT) working group
  – Addressed the essential types of information associated with persistent identifiers (PID)

• Deep Carbon Observatory (DCO) Data Portal
  – Centrally-managed digital object identification, object registration, metadata management and knowledge graph curation.
  – http://deepcarbon.net
Research Questions

- Each defined data type needs a stable and resolvable PID
- Provide semantics - meaning and context - to the defined data types
- Annotate datasets with one or more defined data types
The DTR primitives are comparable to a list of **BASIC DATA TYPE CLASSES** in the DCO ontology, e.g. Dataset, Image, Video, Audio, etc.
A registered DCO dataset is asserted as an instance of one of those basic data type classes.
It is possible to further annotate the dataset with the **SPECIFIC DATA TYPES** defined within a DTR, and each data type has a unique PID.
Possible DCO-DTR Approaches

• An integration architecture
  – DCO Data Portal is built on the VIVO platform
  – DTR and DCO-VIVO as separate knowledge bases
  – DCO-VIVO uses DTR API to access data type information

• A self-contained architecture
  – To have the functionality of DTR completely within the DCO Data Portal
  – Need to modify the DCO Ontology, e.g. add a class dco:DataType and collect properties associated with it

*We are currently working on this*
Initial results

- Updates to the DCO Ontology:
  - A new class dco:DataType. Each specific data type is an instance of it
  - An object property dco:hasDataType linking a dataset and a data type
  - A collection of other classes and properties associated with dco:DataType
An example

The basic data type:
```xml
dco:dcoOntology
dco:hasDataType
dco:RDF.
```

The specific data type:
```xml
dco:dcoOntology
dco:hasDataType
dco:RDF.
```
Each registered object, such as a data type, has a unique DCO ID, which is resolvable by the global Handle System.
Future works

- More **use case analyses** relevant to data types in the DCO community
- **Refine the schema** for the annotation and provenance of specific data types
- A faceted ‘**data type browser**’ on the DCO Data Portal
- **Interoperability** between DCO specific data types and data types registered in other DTRs
  - The integration architecture between DCO-VIVO and DTR

**Thank you!**
• Backup slides
Update the DCO Dataset Browser

Data type as a facet in dataset searching and browsing