Building and Maintaining a Registry for PID Info Types

A DTR of the ePIC Persistent Identifier Consortium

Ulrich Schwardmann, GWDG

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG)

Am Fassberg, 37077 Göttingen
ulrich.schwardmann [at] gwdg.de

17 September 2016, Denver
see: http://dtr.pidconsortium.eu:8081
currently 75 PID-BasicInfoTypes and 57 PID-InfoTypes defined
What was still missing?

Type conformity and validation

- InfoType information has to be particularly reliable, because
  - the functionality of the data services is dependent on a correct preprocessing.
  - this kind of metadata is interpreted by automated services
  - therefore it is necessary to avoid each precondition of human interpretation.

- **Schemas** need to be part of the information type description
  - and have to be defined in a clearly determined and reproducible way.
  - only an automatic process can guarantee this.
Hierarchies of Types

- Information types are often referring to simpler types:
  - a geolocation contains longitude, given in sexagesimal or decimal form.
  - citation information contains an author, perhaps given by an ID in a certain ID system.
- They are eventually based on very basic types:
  - determined by regular expressions or other restrictions.

Suggestion:

- Information types are recursively built out of a finite combination of:
  - information types and
  - such basic information types.
- Possible advantage: reuse of schemas.
Hierarchies of Types

21.T11148/a77cd6959b4fff9a9c50
Type Name: time-period
Type: PID-InfoType

begin-time

end-time

21.T11148/a045f55e2a7fc9d60a5b
Type Name: date-time
Type: PID-BasicInfoType
Schemas, how to generate and maintain them?

- common practice: manually build a schema according to a given description
  - a lot of manual work for schema derivation and adaption
  - inconsistencies between description and the schema

Automatic schema derivation (in JSON):

- exactly describe the information type dependencies in the type description in the DTR
  - enable as much flexibility in the JSON framework as possible
  - the description of dependencies is a derivation from the canonical type description set

- exploit the hierarchy in an automated process
- basic information types have a (simple) schema as leaves in the dependency graph
- store the schema in the type description of the DTR
Hierarchies of Types

```json
```
Validation of instances

- all necessary information for type validation can be found in the PID and the DTR
  - and can be retrieved via the REST API of the Handle System or the DTR
- client function calls like `typeIsValid(PID, typeID)` are provided
Validation of DataCite Mandatory Properties

- all mandatory metadata entries for the DataCite MDS are also defined in JSON in the DTR
- types for all metadata entries for the DataCite MDS will follow
- additionally there is a collective type of these mandatory properties defined in the DTR
- for both metadata instances in JSON and XML exist schema
- there exists a crosswalk from the ePIC JSON to the DataCite XML
  - validation of the JSON type instance guarantees a valid DataCite XML

Conclusion: Types are a great medium to enable more interoperability between PID systems
Validation of tabular data with the Frictionless Data Schema

- Frictionless Data has a hierarchical schema description for necessary data to validate and process tabular data
The FDL types together with corresponding resulting schemas are described in DTR

```
"properties": {
    "CSV-table-description-FLD": {
        "additionalProperties": false,
        "description": "CSV-table-description-FLD@21.T11148_66e7993765837104ce3",
        "properties": {
            "$ref": "/definitions/21.T11148_bbbe2b8ca636e95eaffc",
            "fields": {
                "$ref": "/definitions/21.T11148_39c5c9e5b9a54b398562"
            },
            "foreignKeys": {
                "$ref": "/definitions/21.T11148_071c82c85172 Ved4a8a5"
            },
            "primaryKey": {
                "$ref": "/definitions/21.T11148_17ce43f9612c494cc3e3"
            }
        },
        "required": [
            "fields",
            "CSVDDF"
        ],
        "type": "object"
    }
},
"required": [
    "CSV-table-description-FLD"
],
"type": "object"
```
Validation of tabular data

Example: Validate the CSV file at http://techslides.com/demos/country-capitals.csv

- columns are country code, city name, latitude, longitude, region-code, continent-names
- for these columns exist (candidate) InfoTypes
- Validation by a simple python script
  - gets an array of the PITs of the column types
  - retrieves the CSV file above
  - gets the names and the schemas for the column types from the DTR
    - using the client library from above
  - validates the column entries with the schemas
  - and outputs each row with the offended column entry
Validation of tabular data

Example: Validate the CSV file at
http://techslides.com/demos/country-capitals.csv

- columns are country code, city name, latitude, longitude, region-code, continent-names
- for these columns exist (candidate) InfoTypes
- Validation by a simple python script
  - gets an array of the PITs of the column types
  - retrieves the CSV file above
  - gets the names and the schemas for the column types from the DTR
    - using the client library from above
  - validates the column entries with the schemas
  - and outputs each row with the offended column entry
Validation of tabular data

```

ERROR at line 0002 item 4: Somaliland, Hargeisa, 9.55, 44.050000, NULL, Africa
ERROR at line 0027 item 2: Bangladesh, Dhaka, 90.400000, 23.716666666666666, BD, Asia
ERROR at line 0229 item 2: United States, Washington, D.C., 38.883333, -77.000000, US, Central America
ERROR at line 0242 item 4: Northern Cyprus, North Nicosia, 35.183333, 33.366667, NULL, Europe
```
Endorsement and Deprecation of Types

- registration of PID InfoTypes is an endorsement process
  - currently all registered PIT are **in preparation**
  - after their description is sufficiently settled they become **candidates**
  - after there is no change request anymore they become **approved** and will be frozen
  - new version is necessary for the changement of approved types
    - in this case the former type becomes **deprecated**
- the endorsement of PID InfoTypes needs a **reviewing board**
Questions and Discussion

Thanks for your attention!

View also the ePIC Webpage:

http://pidconsortium.eu