

The PID Info Type Use Case

RDA P7 - Data Typing Working Group

Ulrich Schwardmann

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen
(GWDG)

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

01 March 2016, Tokyo

PID Information Type (PIT)

Ulrich
Schwardmann

DTR for PIT

- is very basic metadata of DO
- used also/primarily for automated processes
- needs as much disambiguity as possible
- use *context free grammar* to define the content of a PIT
- JSON schema is an obvious approach here

- gives a definition of a type in JSON
- provides schemas defining
 - how these definitions can be made, and
 - which fields need to be filled in.
- But the content of a type instance has to be validated
- and the **schema for a type itself is not defined** and cannot be derived in general
 - usual schema generators, that define schemas for gives json files derive schemas for how to define a type.
 - This is the *upward* schema, which is already known.

DTR consequences

Ulrich
Schwardmann

DTR for PIT

- Schemas for type instances need to be automatically derived from the type definition
- as an adaption and expansion for DTR one needs definitions for type
 - that are usable to derive schemas for these types
 - and these schemas need to have enough descriptive power:
 - the derivable schemas should cover a considerable part of the JSON-schema possibilities
 - test case: is it possible to define JSON as a type?
- DTR adaption: give schemas for the definition of types that
 - are suitable to derive a schema from a type definition for instance validation

Suggestion for DTR schemas

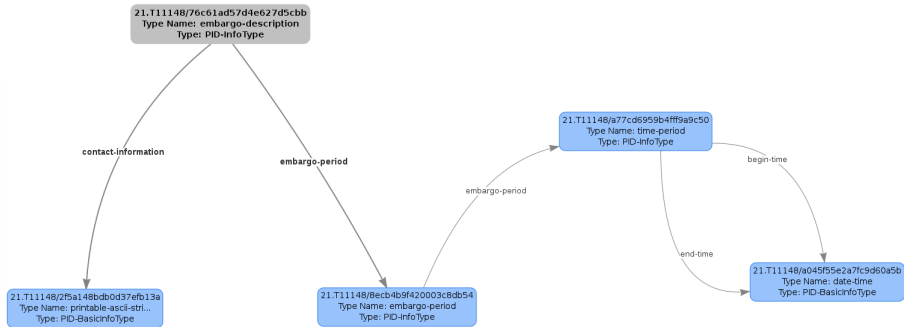
Ulrich
Schwardmann

DTR for PIT

- use **Basic Types**, that have a pattern description by a regular expression
 - it provides
 - **identifier**
 - **regular expression**
 - beside *name*, *description*, *standards*, *provenance*
- and **Derived Types**, that have
 - **identifier**
 - and **properties** defined just by
 - **Basic Types** referenced by their identifier
 - **Derived Types** referenced by their identifier

examples

Ulrich



examples

date-time ✕

Type: PID-BasicInfoType

[Full Object](#) [Digital Object View](#) [JSON View](#) [Verbose View](#) [Start Object](#) [Clone Object](#) [Show Relationships](#)

Identifier

21.11148:a04395e2a7c9b90a5b

Type Name *

date-time

please use printable ascii characters without blank

Description

combined date and time representations as string (ISO 8601)

Applicable Standards or Recommendations

Nature of Applicability	Standard Name *	Issued By *	Details
depends	8601 TID or standard number/name	ISO	

Provenance

Contributors of this Record

Identified Using *	Name *	Details
Text	Ulrich Schwardmann	GWDO

Creation Date

2010-02-23T11:47:17.663Z

Last Modification Date

2010-02-23T11:47:17.687Z

Regular Expression *

`^([0-9]{4})-([01]?[0-1]?[0-9]|10-12|0-9|30-31|T|0-13|0-9|20-31|0-5|0-9|0-5|0-9|0-9|P|Z|[-+]|00-11|0-9|20-31|0-5|0-9|P|Y|S`

Flavour of RegExp *

Ulrich
Schwardmann

DTR for PIT

examples

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "definitions": {
    "21.T11148_a045f55e2a7fc9d60a5b": {
      "description": "date-time",
      "type": "string",
      "pattern": "^(?([0-9]{4})-([0]?[1-9]|1[0-2])-([0-2][0-9]|3[0-1])(T(?([0-1][0-9]|2[0-3]):([0-5][0-9]):([0-5][0-9])(.[0-9]*)?)?(Z|([+|-]([0-1][0-9]|2[0-3]):[0-5][0-9]))?)?)?$"
    }
  },
  "type": "object",
  "properties": {
    "time-period@21.T11148_a77cd6959b4fff9a9c50": {
      "description": "time-period",
      "properties": {
        "begin-time": {
          "$ref": "#/definitions/21.T11148_a045f55e2a7fc9d60a5b"
        },
        "end-time": {
          "$ref": "#/definitions/21.T11148_a045f55e2a7fc9d60a5b"
        }
      }
    }
  }
},
"required": ["time-period@21.T11148_a77cd6959b4fff9a9c50"]
}
```

Ulrich
Schwardmann

DTR for PIT