A DCMI constraint language for Description Set Profiles

Thomas Baker
Dublin Core Metadata Initiative

2013-09-11: RDF Validation Workshop, Cambridge, MA
From Record Format to RDF vocabulary

- **1995** Dublin Core as “simple metadata record”
- **1999** Dublin Core as an RDF vocabulary
- **2000** “Application profiles” mix-and-match
- **2003-2007** DCMI Abstract Model
  - Basis for templating validatable record formats
  - Formats straightforwardly mappable to RDF triples
Bridging two mindsets

- Oriented to *Record Formats*
  - Bounded sets of fields to be “filled in” with information

- Oriented to *Graphs*
  - Unbounded webs of information connected by statements
Let’s take a graph...
...view it in terms of a “record”...
...focusing on components that can be validated...
..and abstract those components to a generalized model.
...using grouping constructs with no equivalent in RDF.
Heuschrecken brauchen ökologische Ausgleichsflächen

Subject | Predicate | Object
--- | --- | ---
agris:CD2001000179 | dct:title | "Heuschrecken..."@de
agris:CD2001000179 | dct:creator | :PB
:PBS | foaf:name | "Peter, B."
Heuschrecken brauchen ökologische Ausgleichsflächen

Subject | Predicate | Object
--- | --- | ---
agris:CD2001000179 | dct:subject | agrovoc:c_4416k
agris:CD2001000179 | dct:title | "Heuschrecken..."@de
agris:CD2001000179 | dct:creator | :PB
:PBS | foaf:name | "Peter, B."
2008 draft Description Set Profile Constraint Language

- Constraints commonly required for validation:
  - `minOccurs/maxOccurs`
  - `resourceClass` (what is the Description about?)
  - `standalone` (describe Person without describing Book?)
  - `valueURI` (mandate a particular controlled vocabulary)
Templates for Description Sets

Constraints on Templates

- Data using this Description Set Profile describes:
  a Resource with *exactly one* [DC] *title*,
  the [DC] *subject* of which is a *URI from AGROVOC*.
  Authors, i.e. members of the [FOAF] *class Person* that have [FOAF] *names*.
Templates and Constraints

- Description Template constraints: Identifier, Standalone, Minimum occurrence, Maximum occurrence, Resource class membership
- Statement Template constraints: Minimum occurrence, Maximum occurrence, Type
  - Property constraints: Property list, Sub-property
  - Literal Statement Template constraints:
    - Literal value constraints: Literal list, Literal language, Literal language list, Syntax Encoding Scheme, Syntax Encoding Scheme list
  - Non-literal Statement Template constraints:
    - Non-literal value constraints: Description template reference, Class membership
      - Value URI constraints: Value URI occurrence, Value URI list
      - Vocabulary encoding scheme (VES) constraints: VES occurrence, VES list
      - Value string constraints: Minimum occurrence, Maximum occurrence (plus all other constraints that apply to literal values - see above)
Broad humor and bitter irony collide in this fictional autobiography of Rabo Karabekian, who, at age seventy-one, wants to be left alone on his Long Island estate with the secret he has locked inside his potato barn. But then a voluptuous young widow badgers Rabo into telling his life story—and Vonnegut in turn tells us the plain, heart-hammering truth about man’s careless fancy to create or destroy what he loves.

```
"hasInstance": "http://bibfra.me/instance/u2-1"
```

```json
{
    "id": "http://bibfra.me/instance/u2-1",
    "type": "HardcoverBook",
    "edition": "1st",
    "volumes": "1",
    "pages": "300",
    "date": "1987-09-02",
    "publisher": "http://bibfra.me/auth/org/delacorte_press",
    "ISBN": "9780385295901",
}
```
**Bluebeard**

A novel by Kurt Vonnegut, 2008

**Summary:**

Broad humor and bitter irony collide in this fictional autobiography of Rabo Karabekian, who, at age seventy-one, wants to be left alone on his Long Island estate with the secret he has locked inside his potato barn. But then a voluptuous young widow badgers Rabo into telling his life story—and Vonnegut in turn tells us the plain, heart-hammering truth about man's careless fancy to create or destroy what he loves.

**hasInstance:** [**http://bibfra.me/instance/u2-1**](http://bibfra.me/instance/u2-1)
Show the user readable/editable document...

```html
http://dublincore.org/documents/2008/10/06/dsp-wiki-syntax/DescriptionSetProfile-dist.zip
```
...with embedded constraints...

http://dublincore.org/documents/2008/10/06/dsp-wiki-syntax/DescriptionSetProfile-dist.zip
...extractable to a generic representation...

http://dublincore.org/documents/2008/10/06/dsp-wiki-syntx/DescriptionSetProfile-dist.zip
...usable to generate editor, validator, etc.

http://dublincore.org/documents/2008/10/06/dsp/wiki-syntx/DescriptionSetProfile-dist.zip
Singapore Framework for Dublin Core Application Profiles (2007)
An updated Description Set Profile templating language?

Application Profile

- Functional Requirements
- Domain Model
- Description Set Profile
- Metadata Vocabularies
- DSP Language
- Compatible Syntaxes

Domain Standards

- Community Domain Model
- Metadatas
- RDFS
- RDF

Foundation Standards

Wednesday, September 11, 2013
Base a Description Set Profile on specific requirements!
Model “reality” with a light touch!
Principle of Minimal Semantic Commitment (using DC and similar vocabularies)
Constrain the *data* *(not the vocabularies)!*
For consistency and quality control

---

**Application Profile**

- **Functional Requirements**
- **Domain Model**
- **Description Set Profile**
- **Data Format**

**Usage Guidelines**

- **annotates**

**Domain Standards**

- **Community Domain Model**
- **Metadata Vocabularies**
- **DSP Language**
- **Compatible Syntaxes**

**Foundation Standards**

- **RDFS**
- **RDF**

---

Wednesday, September 11, 2013
DSP design philosophy

• Design data for peaceful coexistence
• Pragmatic control of data quality
  – Go easy on sweeping ontological generalizations.
  – Constrain data produced.
  – Expect unexpected uses.
• Simple models have bigger audiences
DSP Requirements

- Authored in an idiom usable by normal people
- Constraints reusable for data automatic validation
September 4 discussion at DC-2013, Lisbon

• SPARQL a good candidate as a rules language for expressing constraints

• But users may need generic constraints language:
  – for authoring Application Profiles
  – for understanding Application Profiles

• Something like DC-DSP...?
  – using a syntax straightforwardly convertible into SPARQL queries?
  – as user-facing terminology for documenting Application Profiles?