**Dualism of Appellations**

The purpose of this is to provide an **RDF based technical solution** for representing and querying a property which can be at the same time Data and Object type regardless of the fact that it violates the respective constraints or rules.

Practically we can have three options of representing appellations. By taking the example of Alexander the Great with supposed URI: <http://example.com/person/alexander_the_great> we can do the following:

1)      Use the “P1 is identified by” property and an instance of E41 Appellation class:

[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great)

crm:P1\_is\_identified\_by

[<http://example.com/appellation/alexander\_the\_great>](http://example.com/appellation/alexander_the_great) .

[<http://example.com/appellation/alexander\_the\_great>](http://example.com/appellation/alexander_the_great)

rdfs:label

"Alexander the Great" .

2)      Use directly the rdfs:label:

[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great)

rdfs:label

"Alexander the Great" .

3)      Use the “P1 is identified by” property as a data property (violating the rdfs definitions):

[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great)

crm:P1\_is\_identified\_by

"Alexander the Great" .

Based on these examples the following steps were followed to test the practical application of such cases to a triple store. **Virtuoso triple store** was used for the following examples.

1.       The cidoc\_crm.rdfs was altered to include the following:

<rdf:Property rdf:about="P1\_is\_identified\_by">

<rdfs:label <xml:lang=>"en">is identified by</rdfs:label>

<rdfs:domain rdf:resource="E1\_CRM\_Entity"/>

<rdfs:range rdf:resource="E41\_Appellation"/>

</rdf:Property>

<rdf:Property rdf:about="P1\_is\_identified\_by">

<rdfs:label <xml:lang=>"en">is identified by</rdfs:label>

<rdfs:domain rdf:resource="E1\_CRM\_Entity"/>

<rdfs:range rdf:resource=["http://www.w3.org/2000/01/rdf-schema#Literal"](http://www.w3.org/2000/01/rdf-schema#Literal)/>

</rdf:Property>

So, an is identified property was added to the initial schema but with rdfs:Literal as a range.

2.       The cidoc crm schema was uploaded in virtuoso and the following query (give me the range of P1\_is\_identified\_property) was executed to be sure that the changes have been applied:

prefix crm: [<http://www.cidoc-crm.org/cidoc-crm/>](http://www.cidoc-crm.org/cidoc-crm/)

prefix rdfs: [<http://www.w3.org/2000/01/rdf-schema#>](http://www.w3.org/2000/01/rdf-schema)

select \* where { crm:P1\_is\_identified\_by rdfs:range ?range}

**result:**

|  |
| --- |
| **range** |
| <http://www.cidoc-crm.org/cidoc-crm/E41_Appellation> |
| <http://www.w3.org/2000/01/rdf-schema#Literal> |

So, it is confirmed that the two ranges have been added. I repeat at this point that Virtuoso **does not apply** any semantic validation. The purpose of this test is to prove that this exercise is possible even though conceptually it may not be correct.

3.       The ttl data that was presented previously has been added in virtuoso:

@prefix rdfs: [<http://www.w3.org/2000/01/rdf-schema#>](http://www.w3.org/2000/01/rdf-schema) .

@prefix rdf: [<http://www.w3.org/1999/02/22-rdf-syntax-ns#>](http://www.w3.org/1999/02/22-rdf-syntax-ns) .

@prefix crm: [<http://www.cidoc-crm.org/cidoc-crm/>](http://www.cidoc-crm.org/cidoc-crm/) .

[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great)

crm:P1\_is\_identified\_by [<http://example.com/appellation/alexander\_the\_great>](http://example.com/appellation/alexander_the_great) .

[<http://example.com/appellation/alexander\_the\_great>](http://example.com/appellation/alexander_the_great)

rdfs:label "Alexander the Great" .

<<http://example.com/person/alexander_the_great>>

rdfs:label "Alexander the Great" .

[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great)

crm:P1\_is\_identified\_by "Alexander the Great" .

4.       A query to return all the “identifiers” of alexander the great using the is identified property was applied:

prefix crm: [<http://www.cidoc-crm.org/cidoc-crm/>](http://www.cidoc-crm.org/cidoc-crm/)

prefix rdfs: [<http://www.w3.org/2000/01/rdf-schema#>](http://www.w3.org/2000/01/rdf-schema)

select \* where

{ [<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great) crm:P1\_is\_identified\_by ?identifier }

**result:**

|  |
| --- |
| **identifier** |
| http://example.com/appellation/alexander\_the\_great |
| Alexander the Great |

So, it is obvious that with the same query both the literal and the uri values are returned.

A version of the above query to return also the appellation’s label (but not the uri) is the following:

prefix crm: [<http://www.cidoc-crm.org/cidoc-crm/>](http://www.cidoc-crm.org/cidoc-crm/)

prefix rdfs: [<http://www.w3.org/2000/01/rdf-schema#>](http://www.w3.org/2000/01/rdf-schema)

select ?identifier

where {

{[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great) crm:P1\_is\_identified\_by ?identifier }

UNION

{[<http://example.com/person/alexander\_the\_great>](http://example.com/person/alexander_the_great) crm:P1\_is\_identified\_by ?identifier\_uri .

?identifier\_uri rdfs:label ?identifier }

FILTER (!isURI(?identifier))

}

**With the following result :**

|  |
| --- |
| **Identifier** |
| Alexander the Great |
| Alexander the Great |

The next question is, if P1 can be declared superproperty of rdfs:label, so that the query for P1 returns everything CRM regards as Appellation. It works:

It was tested by altering the cidoc-crm rdfs file, importing it in virtuoso and asking for the subproperties of rdfs:label as follows:

<rdf:Property rdf:about="P1\_is\_identified\_by">

<rdfs:label <xml:lang=>"en">is identified by</rdfs:label>

<rdfs:label <xml:lang=>"ru">идентифицируется посредством</rdfs:label>

<rdfs:label <xml:lang=>"fr">est identifiée par</rdfs:label>

<rdfs:label <xml:lang=>"pt">é identificado por</rdfs:label>

<rdfs:domain rdf:resource="E1\_CRM\_Entity"/>

<rdfs:range rdf:resource="E41\_Appellation"/>

**<rdfs:subPropertyOf rdf:resource=**[**"http://www.w3.org/2000/01/rdf-schema#label"**](http://www.w3.org/2000/01/rdf-schema#label)**/>**

</rdf:Property>

Query (Give me all the subproperties of rdfs:label) :

select \* where {

?p rdfs:subPropertyOf rdfs:label

}

Result from Virtuoso:

p:

<http://www.cidoc-crm.org/cidoc-crm/P1_is_identified_by>