Examples of Provenance Query Rewriting (draft)

Authors: *Nikos Minadakis, Michalis Mountantonakis, Yannis Marketakis, Pavlos Fafalios, Yannis Tzitzikas*

Last update: *Nov 7, 2014*

Contact: *marinetlo@ics.forth.gr*

Contents

Introduction

The objective of this document is to demonstrate the automatic method that is currently supported that takes as input a query and rewrites it in a way that makes evident the source of each row in the answer. The queries that follow are expressed using the ontology MarineTLO which is described in http://www.ics.forth.gr/isl/MarineTLO . The method for query rewriting is described in [ESWC'2014], while the tool *SPARQL Query Rewriting Tool* (also accessible through the web site of MarineTLO) implements this rewriting. Furthermore the document demonstrates the usage of this SPARQL query enrichment in order to detect conflicts between sources, to filter the results according to the data sources and to achieve normalization of numeric values. The usage is being demonstrated through a set of SPARQL examples. For each example a short description is given, its corresponding SPARQL expression and the rewritten SPARQL query with their results.

Example Query 1

Query Description: Find the FAO codes of the water areas in which "Thunnus Albacares" is native

Original SPARQL query expression:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT distinct ?waterareacode
WHERE {
  eco1:thunnus_albacares tloimarine:LX13_is_native_at ?waterarea.
  ?waterarea rdf:type tloCore:BC15_Water_Area .
  ?waterarea  tloCore:LC1_is_identified_by ?x .
  ?x tloimarine:assignedCode ?waterareacode .
}
```

Original Query Results:

waterareacode
34
27
61
31
41
47
21
57
51
81
48
88
77
71

87

SPARQL query with provenance support:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT distinct ?waterareacode ?source1 ?source2 ?source3
WHERE {
  graph ?source1 {eco1:thunnus_albacares tloimarine:LX13_is_native_at ?waterarea}.
  graph ?source2 {?waterarea  tloCore:LC1_is_identified_by ?x} .
  graph ?source3 {?x tloimarine:assignedCode ?waterareacode} .
}
```

Provenance Query Results:

waterareacode	g0	g1	g2	g3
41	Fishbase	Fishbase	Fishbase	Fishbase
31	Fishbase	Fishbase	FLOD	FLOD
88	Fishbase	Fishbase	FLOD	FLOD
57	Fishbase	Fishbase	Fishbase	Fishbase
41	Fishbase	FLOD	FLOD	FLOD
81	Fishbase	FLOD	Fishbase	Fishbase
31	Fishbase	FLOD	Fishbase	Fishbase
57	Fishbase	FLOD	FLOD	FLOD
31	Fishbase	Fishbase	Fishbase	Fishbase
81	Fishbase	Fishbase	Fishbase	Fishbase

Example Query 2

Query Description: Find the scientific name, the authority name and the authority date for the species "Thunnus albacares"

Original SPARQL query expression:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT ?sname ?aname ?date
WHERE{
  eco1:thunnus_albacares tloCore:LX6_type_was_attributed_by ?sbn .
  ?sbn tloimarine:assignedName ?sname .
  ?sbn tloCore:LC13_is_carried_out_by ?sbn2 .
  ?sbn2 tloimarine:name ?aname .
  ?sbn tloimarine:assignedDate ?date
}
```

Original Query Results:

sname	aname	date
thunnus albacares	Bonnaterre	1788
Thunnus albacares	Bonnaterre	1788
Thunnus albacares	Bonnaterre	1788

SPARQL query with provenance support:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT ?sname ?aname ?date ?source1 ?source2 ?source3 ?source4 ?source5
where{
  graph ?source1 { eco1:thunnus_albacares tloCore:LX6_type_was_attributed_by ?sbn }.
  graph ?source2 { ?sbn tloimarine:assignedName ?sname }.
  graph ?source3 { ?sbn tloCore:LC13_is_carried_out_by ?sbn2 }.
  graph ?source4 { ?sbn2 tloimarine:name ?aname }.
  graph ?source5 { ?sbn tloimarine:assignedDate ?date }
}
```

Provenance Query Results:

sname	aname	date	source1	source2	source3	source4	source5
thunnus albacares	Bonnaterre	1788	http://www.ics.forth.gr/isl/Fishbase	http://www.ics.forth.gr/isl/Fishbase	http://www.ics.forth.gr/isl/Fishbase	http://www.ics.forth.gr/isl/Fishbase	http://www.ics.forth.gr/isl/Fishbase
Thunnus albacares	Bonnaterre	1788	http://www.ics.forth.gr/isl/Worms	http://www.ics.forth.gr/isl/Worms	http://www.ics.forth.gr/isl/Worms	http://www.ics.forth.gr/isl/Worms	http://www.ics.forth.gr/isl/Worms
Thunnus albacares	Bonnaterre	1788	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia

Example Query 3

Query Description: Find the codes of the species “Thunnus Albacares”

Original SPARQL query expression:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT distinct ?codev
WHERE {
  eco1:thunnus_albacares tloCore:LX1_is_identified_by ?code .
  ?code tloimarine:assignedCode ?codev .
  ?code tloimarine:LX_has_code_type ?codeType
}
```

Original Query Results:

codev
YFT
yft
00000491
2497
oek
sbz

00000366
2371
00002733
11080
WoRMS:127027

SPARQL query with provenance support:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT distinct ?codev ?source1 ?source2 ?source3
WHERE {
  graph ?source1 { eco1:thunnus_albacares tloCore:LX1_is_identified_by ?code }.
  graph ?source2 { ?code tloimarine:assignedCode ?codev }.
  graph ?source3 { ?code tloimarine:LX_has_code_type ?codeType }
}
```

Provenance Query Results:

codev	g0	g1	g2
2497	TLObasedDataWarehouseV3	FLOD	FLOD
2371	TLObasedDataWarehouseV2	FLOD	FLOD
11080	TLObasedDataWarehouseV3	FLOD	FLOD
00002733	TLObasedDataWarehouseV3	FLOD	FLOD
sbz	FLOD	FLOD	FLOD
2497	FLOD	FLOD	FLOD
sbz	TLObasedDataWarehouseV3	FLOD	FLOD
11080	FLOD	FLOD	FLOD
00000491	TLObasedDataWarehouseV2	FLOD	FLOD

Sources Filtering

If someone wants to retrieve results that come from specific sources, the produced query has to be formulated. Let's consider the case that demands Provenance Query 2 to be changed so as to return results coming from Fishbase and Worms. The required changes are in red color:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX eco1: <http://www.ecoscope.org/ontologies/ecosystems/>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT ?sname ?aname ?date ?source1 ?source2 ?source3 ?source4 ?source5
WHERE{
  graph ?source1 { eco1:thunnus_albacares tloCore:LX6_type_was_attributed_by ?sbn }.
  graph ?source2 { ?sbn tloimarine:assignedName ?sname }.
  graph ?source3 { ?sbn tloCore:LC13_is_carried_out_by ?sbn2 }.
  graph ?source4 { ?sbn2 tloimarine:name ?aname }.
  graph ?source5 { ?sbn tloimarine:assignedDate ?date } .
FILTER
(regex(?source1,'Fishbase')||regex(?source2,'Fishbase')||regex(?source3,'Fishbase')||regex(?source4,'Fish
base')||regex(?source5,'Fishbase')
||regex(?source1,'Worms')||regex(?source2,'Worms')||regex(?source3,'Worms')||regex(?source4,'Worm
s')||regex(?source5,'Worms'))
```

```
}
```

Detecting Conflicts

Let's consider the case that there are inconsistencies between the sources. An example on how to recognize the problematic case of attaching two different scientific names to a species follows:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT ?sname1 ?sname2 ?species ?source1 ?source2
WHERE{
 { graph ?source1 { ?species tloCore:LX6_type_was_attributed_by ?sbn .
   ?sbn tloimarine:assignedName ?sname1 }.
 }.
 { graph ?source2 { ?species tloCore:LX6_type_was_attributed_by ?sbn2 .
   ?sbn2 tloimarine:assignedName ?sname2
 }
}
FILTER(!SAMETERM(?sname1,?sname2))
}
```

Indicative Results:

sname1	sname2	species	source1	source2
Anoplogaster brachycera	Anoplogaster cornuta	http://dbpedia.org/resource/Fangtooth	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Anoplogaster cornuta	Anoplogaster brachycera	http://dbpedia.org/resource/Fangtooth	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus platypterus	Istiophorus albicans,	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus platypterus	Istiophorus albicans,	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus platypterus	Istiophorus albicans,	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus platypterus	Istiophorus albicans,	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus albicans,	Istiophorus platypterus	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus albicans,	Istiophorus platypterus	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus albicans,	Istiophorus platypterus	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Istiophorus albicans,	Istiophorus platypterus	http://dbpedia.org/resource/Sailfish	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Galaxias fuscus	Galaxias olidus	http://dbpedia.org/resource/Mountain_galaxias	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia
Galaxias olidus	Galaxias fuscus	http://dbpedia.org/resource/Mountain_gala	http://www.ics.forth.gr/isl/DBpedia	http://www.ics.forth.gr/isl/DBpedia

		xias		

Implementing Conflict Resolution Policies at Query Level

This examples demonstrates how one could implement a conflict resolution policy (for the form min/max/avg) using the query language. The next query **returns the length** of Cottunculus microps species and the provenance of this information:

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT  ?length ?source
WHERE{
 graph ?source {
 ?bn rdf:type tloCore:BC44_Attribute_Assignment .
 ?bn tloCore:LX6_assigned_attribute_to_type
 <http://www.fishbase.org/entity#cottunculus_microps>.
 ?bn tloCore:LC9_assigned ?bn2 .
 ?bn2 rdf:type tloCore:BC62_Statistic_Indicator .
 ?bn2 tloCore:LC19_has_dimension tloimarine:Lenght .
 ?bn2 tloCore:LC20_has_value ?length .
}
}
```

Query Results:

length	source
30	http://www.ics.forth.gr/isl/Fishbase
35	http://www.ics.forth.gr/isl/Worms
25	http://www.ics.forth.gr/isl/Ecoscope

According to the query results there are three different values for the length of Cottunculus microps species coming from three different sources. To select the average of these values the following query has to be executed.

Query that **returns the average length** of Cottunculus microps species :

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

SELECT avg(xsd:integer(?value)) as ?average_length
WHERE{
 ?bn rdf:type tloCore:BC44_Attribute_Assignment .
 ?bn tloCore:LX6_assigned_attribute_to_type <http://www.fishbase.org/entity#cottunculus_microps>.>.
 ?bn tloCore:LC9_assigned ?bn2 .
```

```
  ?bn2 rdf:type tloCore:BC62_Statistic_Indicator .
  ?bn2 tloCore:LC19_has_dimension tloimarine:Lenght .
  ?bn2 tloCore:LC20_has_value ?value .
}
```

Query Results:

average_length
30

Apart from selecting and checking the conflicted values (e.g. using the average example above), one might want to create a dataset that contains only not conflicting values. This is demonstrated by the following query.

```
DEFINE input:inference 'http://www.ics.forth.gr/isl/Schema'
DEFINE input:same-as 'yes'
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tloimarine: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetloimarine.owl#>
PREFIX tloCore: <http://www.ics.forth.gr/isl/MarineTLO/v4/marinetlo.owl#>

INSERT into <graph5> { ?bn2 tloCore:LC20_has_value ?average}
SELECT ?bn2 ?average
WHERE{
 {
  SELECT avg(xsd:integer(?value)) as ?average
  WHERE{
   ?bn rdf:type tloCore:BC44_Attribute_Assignment .
   ?bn tloCore:LX6_assigned_attribute_to_type ?species.
   ?bn tloCore:LC9_assigned ?bn2 .
   ?bn2 rdf:type tloCore:BC62_Statistic_Indicator .
   ?bn2 tloCore:LC19_has_dimension tloimarine:Lenght .
   ?bn2 tloCore:LC20_has_value ?value.
  }
 }
 ?bn rdf:type tloCore:BC44_Attribute_Assignment .
 ?bn tloCore:LX6_assigned_attribute_to_type ?species.
 ?bn tloCore:LC9_assigned ?bn2 .
 ?bn2 rdf:type tloCore:BC62_Statistic_Indicator .
 ?bn2 tloCore:LC19_has_dimension tloimarine:Lenght .
}
```

References

[**ESWC'2014**]: Y. Tzitzikas, N. Minadakis, Y. Marketakis, P. Fafalios, C. Alloca, M. Mountantonakis and I. Zidianaki. MatWare: «***Constructing and Exploiting Domain Specific Warehouses by Aggregating Semantic Data***», 11th Extended Semantic Web Conference (ESWC'14), Anissaras Hersonissou, Crete, Greece, May 2014.