

RIF & SPARQL

@AxelPolleres

Digital Enterprise Research Institute, National University of Ireland, Galway

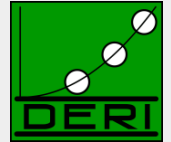


These slides are provided under creative commons
Attribution-NonCommercial-ShareAlike 3.0 Unported License!

© Copyright 2009 Digital Enterprise Research Institute. All rights reserved.



SPARQL1.1 Entailment Regimes

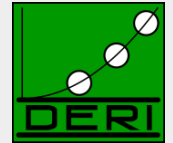


- SPARQL1.1 will define SPARQL query answering over RIF rule sets, RDFS and OWL:

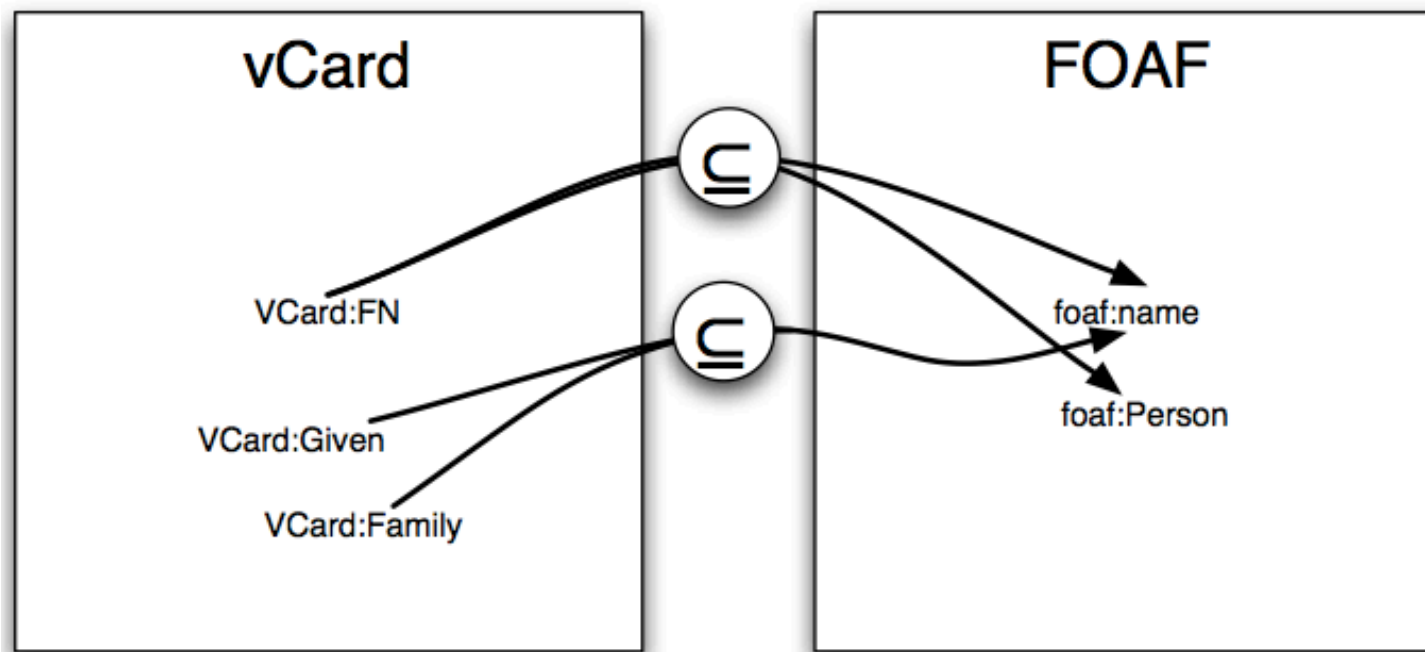
- <http://www.w3.org/TR/sparql11-entailment/>

- RDF Entailment Regime
 - RDFS Entailment Regime
 - D-Entailment Regime
 - OWL 2 RDF-Based Semantics Entailment Regime
 - OWL 2 Direct Semantics Entailment Regime
 - **RIF Core Entailment**

Why use RIF with SPARQL?



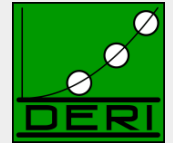
- RIF can express mappings between/within vocabularies not expressible by RDFS,OWL, e.g.



VCard:FN rdfs:subPropertyOf foaf:name .
VCard:FN rdfs:domain foaf:Person .

```
{?X foaf:name ?FullN } :- { ?X VCard:Given ?G. ?X VCard:Family ?F }  
AND ?FullN = fn:concat(?G, " ", ?F)
```

Combination of RIF, RDFS, OWL & SPARQL:



Digital Enterprise Research Institute

www.deri.ie

- Option 1: Use/implement combination semantics, as defined per <http://www.w3.org/TR/rif-rdf-owl/>
- Option 2: Use custom RIF rules encoding RDFS and OWL inference rules you want to apply
 - Advantage: declare explicitly which OWL2RL/RDFS inferences you do:
 - switch on/off the inference rules you do
 - and/or extend by own rules

■ RIF Core allows to encode RDFS, e.g.:

```
rdfs1: { ?S rdf:type ?C } :- { ?S ?P ?O . ?P rdfs:domain ?C . }
```

```
rdfs2: { ?O rdf:type ?C } :- { ?S ?P ?O . ?P rdfs:range ?C . }
```

```
rdfs3: { ?S rdf:type ?C2 } :- { ?S rdf:type ?C1 . ?C1 rdfs:subClassOf ?C2 . }
```

■ RIF Core allows to encode OWL2 RL, e.g. :

```
owl1: { ?S1 owl:SameAs ?S2 } :-  
      { ?S1 ?P ?O . ?S2 ?P ?O . ?P rdf:type owl:InverseFunctionalProperty }
```

```
owl2: { ?Y ?P ?O } :- { ?X owl:SameAs ?Y . ?X ?P ?O }
```

```
owl3: { ?S ?Y ?O } :- { ?X owl:SameAs ?Y . ?S ?X ?O }
```

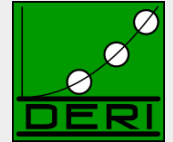
```
owl4: { ?S ?P ?Y } :- { ?X owl:SameAs ?Y . ?S ?P ?X }
```

■ Plus more (custom rules, including Built-ins):

```
{ ?X foaf:name ?FullN } :- { ?X vCard:Given ?G. ?X vCard:Family ?F }  
                          AND ?FullN = fn:concat(?G, " ", ?F)
```

<http://ruleset1.rif>

How to reference to a RIF Ruleset from SPARQL?



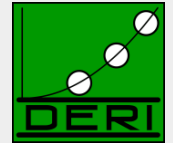
- RIF's so far only a normative syntax is RIF/XML
 - RIF encoding in RDF (RIF/RDF) underway:
http://www.w3.org/2005/rules/wiki/RIF_In_RDF
 - Will also provide a new RDF property `rif:usedWithProfile` to import RIF rulesets (in RIF/XML or RIF/RDF). e.g.

```
<http://ruleset1.rif> rif:usedWithProfile
  <http://www.w3.org/ns/entailment/Simple> .
foaf:homepage a owl:inverseFunctionalProperty.
<http://dblp.13s.../Tim_Berners-Lee>
  foaf:homepage <http://www.w3.org/People/Berners-Lee/> ;
  foaf:name "Tim Berners-Lee" .
<http://www.w3.org/People/Berners-Lee/card#i>
  foaf:homepage <http://www.w3.org/People/Berners-Lee/> ;
  VCard:Given "Timothy";
  VCard:Family "Berners-Lee" .
```

```
SELECT ?P ?N { ?P foaf:name ?N }
```

?P	?N
<dblp/Tim>	Tim Berners-Lee
<w3/B-Lee/card#i>	Tim Berners-Lee
<dblp/Tim>	Timothy Berners-Lee
<w3/B-Lee/card#i>	Timothy Berners-Lee

SPARQL1.1 + RIF Core Semantics



■ Semantics:

- Straightforward: **BGPs matching** defined as being RIF-RDF-entailed by RDF data graph in combination with the referenced ruleset.

■ Attention: Infinite answers possible

- (even though RIF Core has no function symbols):

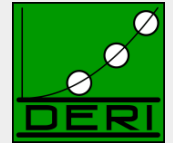
```
:a :b 1 .  
{ ?S ?P (?O + 1) } :- { ?S ?P ?O } .
```

```
SELECT ?O { :a :b ?O . }
```

■ No restrictions on finiteness in SPARQL1.1/RIF per se

- Finite answers up to the user or system
- E.g. restrict to strongly safe RIF Core (inspired by [Eiter et al. 2006] or
- Or system streams out answers (e.g. a la Prolog)

“Why not simply using SPARQL for publishing mappings?”



Digital Enterprise Research Institute

www.deri.ie

■ Why not use SPARQL1.1 CONSTRUCT?*

```
{?X foaf:name ?FullN } :-  
  { ?X VCard:Given ?G. ?X VCard:Family ?F }  
  AND ?FullN = fn:concat(?G, " ", ?F)
```

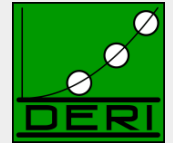
- Answer: RIF provides well-defined interacting of such a mapping with RDFS or OWL mappings, and adhoc approach based on CONSTRUCT doesn't.
- *But: No one holds you from defining a RIF syntax that looks “SPARQLish” SPARQL Rules... [Polleres WWW2007, Schenk-Staab WWW2008]*

* e.g. [Polleres et al. ODBASE2007], Bizer&Schulz COLDB2010]



Enabling **networked** knowledge.

“Why not simply using SPARQL for publishing mappings?”



■ Why not use SPARQL1.1 CONSTRUCT?*

```
CONSTRUCT { ?X foaf:name ?FullN }  
WHERE { ?X vcard: ?F. ?X foaf:lastName ?L  
        BIND ?FullN AS fn:concat(?F, " ", ?L) }
```

- Answer: RIF provides well-defined interacting of such a mapping with RDFS or OWL mappings, and adhoc approach based on CONSTRUCT doesn't.
- *But: No one holds you from defining a RIF syntax that looks “SPARQLish” SPARQL Rules... [Polleres WWW2007, Schenk-Staab WWW2008]*

* e.g. [Polleres et al. ODBASE2007], Bizer&Schulz COLD2010]