# Data Integration Using OWL and Rules

Petr Křemen

December 14, 2018

**Ex. 1** — Download the ZIP archive from the web site and unpack data1.ttl, data2.ttl and ontology.ttl.

**Ex. 2** — Take a look at each of the files in Protege.

**Ex. 3** — Create a new OWL ontology in Protege, import all three ontologies in it and save it along with the other as integration.ttl.

**Answer (Ex. 3)** — Two files containing the integration task are attached:

- integration.ttl – defensive mapping using mainly sub-classing (we infer data compliant with our integration ontology from the original sources)
- solution.ttl – strong mapping using mainly equivalent classes (we infer data compliant with our integration ontology from the original sources and also infer relationships from the other data source. For example, see the additional `dbo:parent` links for the `d1:wenceslas-iv`)

**Ex. 4** — Align the classes and Object Properties of data1.ttl and data2.ttl with ontology.ttl. For example, you might want to say that `d1:parent` is a subclass of (or equivalent class of) `o:parent`, or that `d1:is-child-of` is a subproperty of `inverse(o:has-child)`. Try to be as precise as possible.

**Ex. 5** — Define characteristics (transitivity, functionality, etc.) of the object properties.

**Ex. 6** — Define a SWRL rule that infers `o:has-mother` property assertions using the `o:woman` class and `o:has-parent` property.

**Ex. 7** — Define a SWRL rule that infers `o:has-step-mother` property assertions out of the existing data (e.g. `d1:wenceslas-iv` has three step mothers (we neglect that they need not have lived during Wenceslas' life).

**Ex. 8** — Define the class `mother-with-at-least-two-children`. Which instances belong to it?

**Ex. 9** — Open the SPARQL Query Tab (You will need to have SNAP Plugin installed) and construct a query that retrieves all pairs of step siblings (with the same logic as the SWRL rule in ontology.ttl).

**Answer (Ex. 9)** — The query might look like

```
PREFIX o: <http://onto.fel.cvut.cz/ontologies/osw2018/s11/ontology/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT *
WHERE {
  ?f o:has-child ?c1 ;
     o:has-child ?c2 .
  ?m1 o:has-child ?c1 ;
      o:has-spouse ?f .
  ?m2 o:has-child ?c2 ;
      o:has-spouse ?f .
  ?m1 owl:differentFrom ?m2 .
  ?c1 owl:differentFrom ?c2 .
  FILTER(str(?m1)>str(?m2))
}
```

**Ex. 10** — Take the resulting artifact and upload it into GraphDB. Compare the inferences to those in Protege.

## 1 References

1. https://www.w3.org/Submission/SWRL