Technologies du Web sémantique - TP 2

A Unified Knowledge Repository for Citizen Science Project Descriptions

A Citizen Science project is a scientific project that is carried out mostly be non-professional scientists. In general these project involve many participants who perform tasks such as observation (e.g. bird counting), image analysis (e.g. spotting specific astronomical objects on pictures), transcription of complex handwritten texts, etc.

Your goal is to build an ontology-based access to a repository of CS project descriptions.

Use cases

- import project descriptions from the JRC source (provided in RDF)
- import project descriptions from the WP source (provided in RDF)
- import zooniverse project descriptions (provided in RDF)
- find projects that match criteria such as
  - scientific domain
  - type of tasks
  - goals
  - ...

Todo

1. use Protégé to create an OWL ontology that
   - contains all the classes described in the provided CS Ontology diagram (CSO)
   - has axioms to map projects described with other vocabularies (JRC, WP, ZOO) to CSO (use equivalentClass, subClassOf, equivalentProperty, subPropertyOf)
   - has axioms to precisely describe the classes and their properties
   - has SWRL rules to infer facts that can be useful to answer search queries
2. test the ontology by loading triples from the provided data files and checking that the reasoner performs the expected inferences
3. create a SPARQL endpoint for your knowledge base

- transfer the ontology to a triple store (GraphDB)
  - since the triple store generally don't have a SWRL engine transform your SWRL rules into OWL-RL, when possible
- upload the data files
  - this may require some transformation of the initial RDF data, e.g. to replace list of objects represented by string values, as in `:p :area "France, Italy"`, by reference to objects (URIs): `:p :area geo:France, geo:Italy`.

4. develop a basic (but intelligent) search engine to demonstrate that
   - information from different sources have effectively been integrated
   - it can discover implicit facts (inferred by the triple store reasoner or through queries)
     - for instance, a query to find projects in biology should return the projects about invasive species, birds, trees, etc. . This will require some domain knowledge that you can add to the initial data.

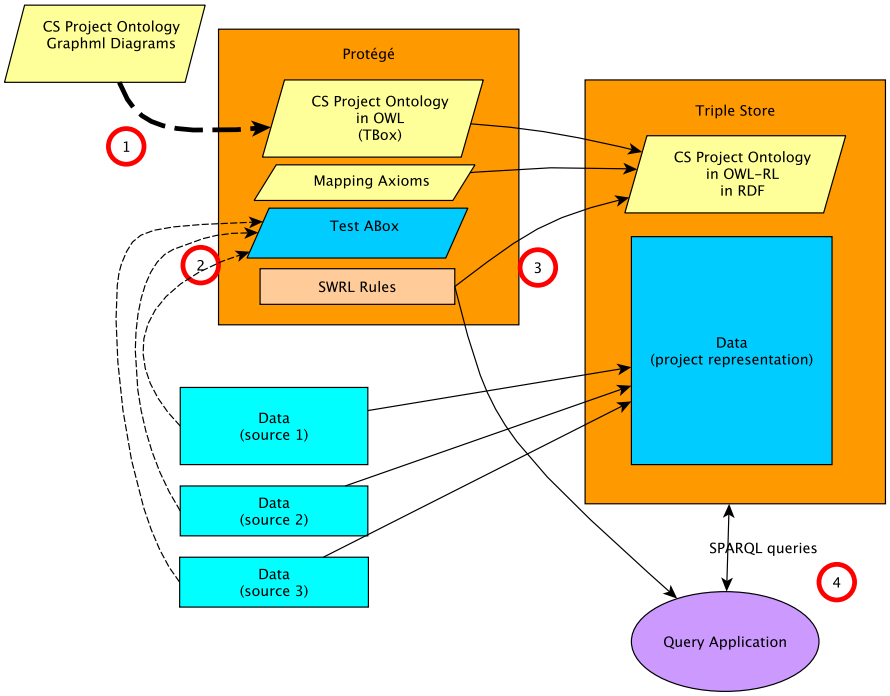

Figure 1: System development steps