Semantic Data

Introduction to project

Jean-Louis Binot

Agenda

- Objectives and topics
- Architecture and selection of tools
- Expected results
- Deadline
- Assessment criteria
- Groups
- Starting point and tips

Objectives

Develop an ontology-based knowledge base for a specific domain, and demonstrate its use and access for queries :

a) at the human level and

b) at the programming level, by 2 (Java) programs using ontology-related APIs.

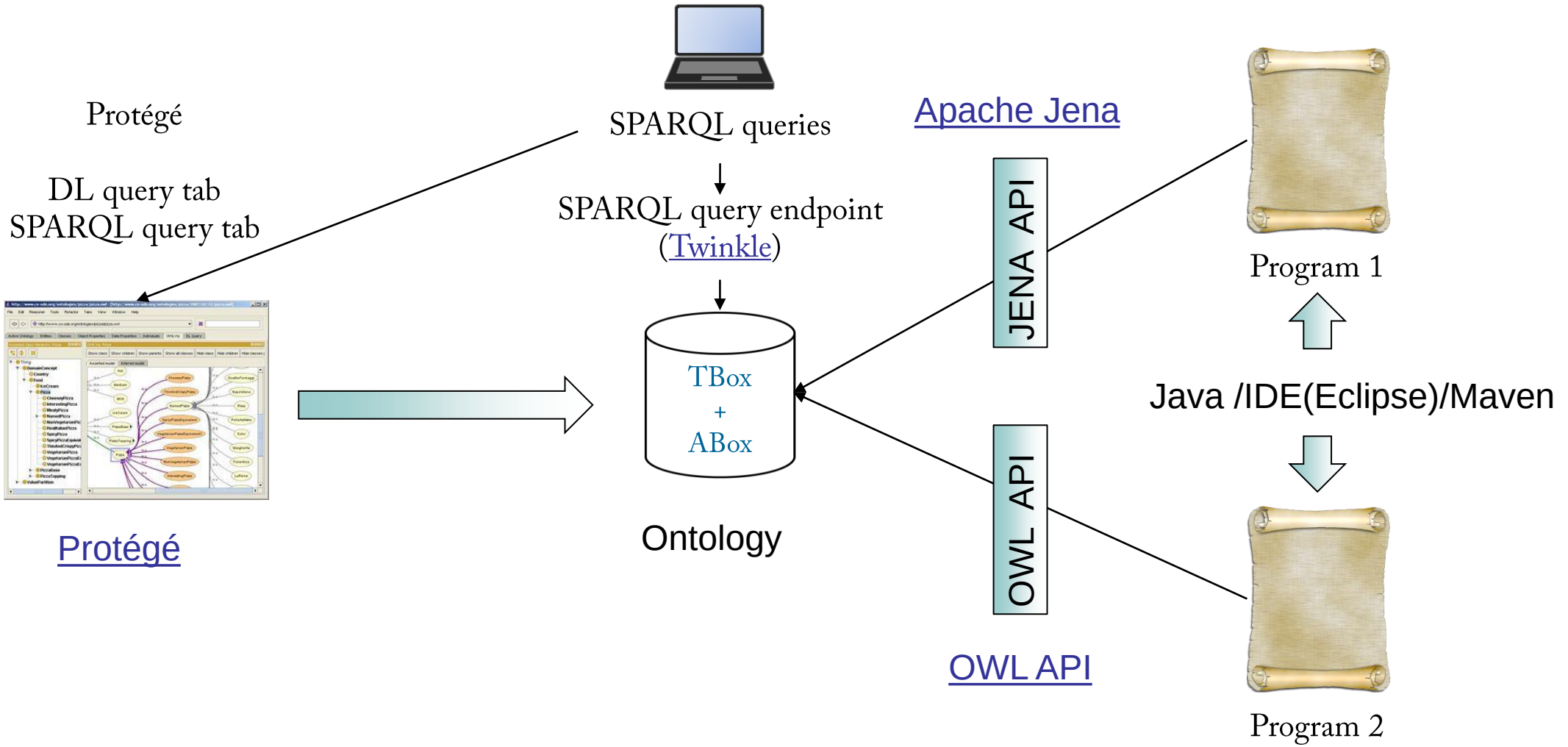
Cover the main steps of an entire process :

- From knowledge acquisition to ontology querying.
- Access by humans and manipulation by application programs.
- Based on a preselected architecture of tools.

Topics

- Topic 1: design and query an ontology about the tourism of province of Liège.
  - Typical information to cover : geography and topology (provinces, regions, cities, natural geography …), touristic information (natural attractions, human attractions …), types of activities (familial, sportive …) infrastructures (transports, hotels, restaurants …) …
- Topic 2 : design and query an ontology about ecology and endangered species.
  - Typical information to cover : ecological structure (biomes, ecosystems, environments, habitats…), classification of animals and plants, where they live (geography, ecology), food chains (producers, consumers, predators, preys…), threatened species and threats (classification, localization, causes …) …
- Topic 3 : design and query an ontology about historical dynasties.
  - Typical information to cover : dynasties (name, historical period, succession order, genealogy - dates, marriages, family relations…), ruled regions (location, nature - kingdoms, empires, duchies …), significant historical information (wars, conquests, discoveries, political relations …)…

Architecture and selection of tools



Expected results

Results will include :

- **An ontology** for the chosen domain, in OWL, developed with the editor **Protégé**.
- The demonstration of the capability of :
- **1. Manually querying the ontology** through :
  - Queries in Manchester syntax, entered in the DL Query tool of the editor Protégé.
  - SPARQL queries addressed through a SPARQL GUI (e.g., Twinkle, or Protégé SPARQL Query tab).
- **2. Using the ontology through applications,** through :
  - **A JAVA program using the JENA API**, able to interact with the ontology at the RDFS level in order to manage it and to query it with SPARQL queries.
  - **A JAVA program using the OWL API**, able to interact with the ontology at the OWL level in order to manage it, and to call a reasoner to perform inferences based on the ontology.

Expected results and Deadline

- Project results will be assessed from :
  - the resulting implementation;
  - a final defense, consisting of a presentation and demonstration, followed by Q&A.
- **Grade allocation**
  - Oral exam : 60%.
  - Project results : 40% (implementation 25%, defense 15%).
- Submission of project results is mandatory for presenting the exam !
  - Project notes can be kept or improved for September session.
- The results will be presented and defended during the session of 12/5/2021.
  - Submission platform : http://submit.montefiore.ulg.ac.be
  - Submission deadline for the code: Sunday 9/5/2021 end of day (23h59).
  - Submission content : the ontology (documented), the code of the API programs (documented) and the prepared example queries. The slides can be sent by email for the defense.

Assessment criteria

- Ontology
  - Loadable, executable - the ontology must be loadable and usable in Protege without infinite loops or crashes and must be consistent when launching the reasoner.
  - Overall quality and coverage - the coverage does not need to be exhaustive, as the domains are open ended, however the overall ontology structure must be complete enough and adequate to represent the main concepts and relations involved in the project domain. Time and space, where relevant, mut be appropriately covered. Some subparts of the model may be developed more in depth than others to demonstrate the possibilities of the ontology.
  - Inferences - the goal is to develop a knowledge base, relying as much as possible on the use of inferences based on appropriate definitions of concepts and properties. This aspect will be explicitly demonstrated in the project defense.
  - Use of concepts - the concept hierarchy must be well thought out for maximal expressivity and efficiency. Concept definitions and hierachy, including inclusion and equivalence axioms, should be well exploited.
  - Use of properties - the possibilities of properties, including object and data properties, property hierarchies, and property characteristics (functional, inverse, transitivity …) should be well exploited. A special attention will be put on the handling of locations, topologies, and other transitive relations.

Assessment criteria

- Ontology ./.
  - Instances - will be in sufficient number to illustrate the ontology and support queries.
  - URIs - URIs to useful informative external sites will be provided for the main concepts.
  - Documentation and reuse - the ontology should be sufficiently commented to understand its structure. Reuse is allowed but must be explicitly referenced.
- Queries (Manchester syntax and SPARQL)
  - Queries must be in sufficient number (at least 5 of each language), of sufficient complexity and variety to illustrate the possibilities of the ontology.
  - Queries must be prepared in advanced and executed correctly.

Assessment criteria

- Jena API demonstrator
  - The Jena program demonstrator will demonstrate the basic management of an ontology (load, modify, save) and the construction and use of SPARQL queries to query the added or modified information.
  - It will offer a decent interface allowing the user to manage and query the ontology interactively.
  - It must obviously be executable without bugs and correctly documented.
- OWL API
  - The OWL API program demonstrator will demonstrate the basic management of an ontology (load, modify, save) and the capability to call a reasoner to perform meaningful inferences.
  - It is not required to provide a user interface, as that was already done for the JENA API; the main purpose of this second demonstrator is to understand and demonstrate the difference between what can be done at the RDF level (JENA API) and at the OWL level (OWL API).
  - It must obviously be executable without bugs and correctly documented.

Assessment criteria ./.

- The defense will consist of
  - A presentation.
  - A demonstration of the ontology itself and the inferences it supports, using the ontology editor Protégé and its reasoner.
  - A demonstration of the Jena and the OWL API program demonstrators.
- The assessment of the defense will be based on :
  - The structure and content of the slides.
  - The execution of a working and convincing, well prepared, demonstration.
  - Respect of timing (to be communicated later).
  - The evidence of contribution from all group members (project + defense).

Group status

Group	Members	Topic
Group 1	Baudinet Dachet Dubart Langer	Ecology
Group 2	Lucas Mazur Perrin Poizat	Ecology
Group 3	Chapeau Duquenne Lievens Thielen	Tourism of Liège
Group 4	Cabay Lamay Polat Tilkin	Dynasties
Group 5	Latour Libert Michiels Ventat	Tourism of Liège

Group	Members	Topic
Group 6	Christians Farer Menouer Pirlet	Ecology
Group 7	Baré Bellefroid Gómez Herrera Wallon	Ecology
Group 8	Bissot Dumoulin Laurent Vrijens	Dynasties
Group 9	Alagov Jiang Koppula Korneev	Tourism of Liège
Group 10	Fiedler Harrik Itouchene Ouali	?

Starting point

- From a theory point of view, you need to understand enough about :
  - Inheritance (chapter 1).
  - Logic and especially Description Logics (chapters 2 and 4).
  - Ontology engineering (chapter 5).
  - RDF / RDFS / SPARQL (chapters 3 and 7).
  - To a lesser extend, Owl (chapter 6) (OWL is handled by the Protégé editor).
  - Optionally : rule extensions (chapter 10).
- From a practical point of view, you need to understand enough about :
  - How to express concepts in description logics (Practice 3).
  - How to edit an ontology with Protégé (Practice 4).
  - How to perform the setup of the tools (Practice 5).
  - How to handle specific design and modeling issues for your ontology (Practice 6).
  - How to query your ontology using SPARQL (Practice 7).

A last tip

Modeling an ontology and getting it right takes more time than one would expect. It is strongly advised to start in time and progress through regular work.

THANK YOU