Complexity Requirements for

Semantic BPM project (2023)

The project will be:

* created in teams of max. 2
* uploaded in a Moodle assignment as a ZIP file
* presented in person on the session exam date (both team members will attend on the same exam date)

Below are the requirements for maximum grade.

**Component A.** a *Modelling Tool* (4.5 p) from which you obtain models as RDF graphs. Your tool should customize BPMN (=> to avoid recreating the BPMN concepts, start from the BEE-UP source library, instead of the empty one)

**Component B.** a *Model-driven front-end* that is capable to access information from models created in Component A

(=> use GraphDB as a repository of models exported in RDF format) (4.5p)

* If you don't create Component A, create some models in BEE-UP and export them to RDF in order to be able to work on Component B. By doing this, you can get maximum 1.5p on Component A.
* If you don't create Component B try at least to demonstrate your ability to query the diagrams of your tool in the GraphDB query console. By doing this, you can get maximum 2.5p on Component B.

## Detailed requirements:

**A. Modelling Tool** = should be a BPMN customization, not a project started from scratch like the cooking tutorial (import the BEE-UP source in ADOxx to link your diagrams to BPMN). The customization should cover several customization layers:

* + **Semantic customization**:
    - at least one new type of model whose contents can be connected through hyperlinks to/from some BPMN elements
  + **Conceptual customization**:
    - at least 2 new concepts + 2 new relationships (either in the BPMN model type or in your own model type)
  + **Functional customization**:
    - at least one menu option that executes some functionality in the modeling tool; the functionality should both read AND write model contents
  + **Graphical customization**:
    - at least one graphical symbol should have an interactive hotspot (either on an existing BPMN concept or one of your own concepts)

Export to RDF at least two interlinked models created with the modeling tool.

.

**B. Model-driven Front-end** = a client component that demonstrates your ability to read diagram contents as graphs stored in GraphDB. You can implement this in any technology your prefer (any technology that can send queries via HTTP to the REST API of GraphDB will work).

This component will contain:

* + **The RDF Graph Database** containing
    - interconnected models exported from Component A
    - additional information imported via OntoRefine (at least two new properties)
    - a Lucene connector allowing search queries over some text property present in RDF database (from the models or from the OntoRefine import)
  + **The client demo** (Web page, mobile app, desktop app, whatever technology you prefer) that demonstrates the following query types:
    - one path query with minimum 3 slashes
    - one query with OPTIONAL or FILTER NOT EXISTS
    - one reasoning query (CONSTRUCT {...} WHERE {...})
    - one query that uses the Lucene connector
    - at least one of the queries should use somehow the external data imported via OntoRefine

## Moodle Assignment details:

**Deadlines for upload will be announced when the exam dates are fixed**

**You have to upload a ZIP file** containing the following, before the assignment deadline:

* **Documentation:** A Word document that includes:
  + Screenshots of the exported models created with component A
  + Screenshot of your metamodel customization (not the entire BEE-UP, only the parts that connect to BPMN concepts). The Metamodel must also contain all the attributes/links you created
  + The SPARQL queries and screenshots of their results
* **Sources:**
  + The ABL source of component A
  + The source code for component B
* The **full graph database** used by component B
  + Repository exported from GraphDB in TriG format, after everything works