

Knowledge Representation and Reasoning

Project outline and guidelines on deliverables

The project's goal is to develop an ontology for annotating autonomously developed (*) information systems and to demonstrate the meaningful exchange of information via those annotations. To this end, you form groups of three to develop an information system, and then they will develop the prototype ontology. You may collaborate with other groups for the creation of the ontology. Each group then proceeds with the annotation of their system and demonstrates the meaningful exchange made possible with a non-trivial demonstrator.

() In this project, you will pretend to be the owner, developer, or maintainer. We will thus simulate a knowledge graph project. Knowledge graph projects are supposed to integrate data from various sources. In this course of 5 ECTS, however, the transformation of your database to a knowledge graph will suffice. You may collaborate with other groups to render parts of this project more realistic. This will require groups to share some concepts, however. Such group collaboration is appreciated and will be considered for the final grade. Important, however, that collaboration across groups is optional to pass this course with flying colors. If done correctly, consider this a possibility to obtain bonus points.*

Table of Contents

Milestone 0: Groups	2
Milestone 1: Database	2
Milestone 2: Ontology Engineering	3
Milestone 3: Annotating your IS (i.e., creating mappings).	4
Milestone 4: Using the Ontology and Knowledge Graph	5
Presentation	6

Milestone 0: Groups

You form groups of three students and communicate the following:

- The names and email addresses of the group;
- The group's name (unless the group wants to be addressed by a number);
- The chosen domain and a description of their information system (or application).

You may draw inspiration from existing applications (and they are highly encouraged to do so), provided they cite their sources of inspiration.

The “deliverable” is thus an **email**.

Milestone 1: Database

The first step is developing the database underpinning their existing information system. The goal of this milestone is threefold: 1) familiarize yourself with the domain, 2) refresh your modeling skills, and 3) have a database schema and instances *simulating* a “legacy” information system. You should end up with a database whose semantics of its structure and contents are clear within your group.

The deliverables for this milestone are

- A **conceptual or database schema** using a modeling language of their choice (ERD, UML, ORM, etc.);
- An **implementation of the relational database** provided as an SQL dump file (i.e., DDL statements);
- A **couple of meaningful examples** to populate your database, either with the dump file or in a separate SQL file. You may initially restrict yourself to a handful of examples, but remember that the exercise becomes more interesting if you have at least a couple of dozen examples. Feel free to avail of existing datasets to populate your database, which you can cite in your technical report.
- A **technical report** that describes your application and elaborates on how your conceptual schema and database support said implementation. Do not merely include and refer to the schema and dump files in that report; describe them. Use this report to write down your assumptions and any integrity constraints and business rules that need to be implemented (i.e., constraints that cannot be modeled in your chosen modeling language).

You may create a database and its population from scratch; develop a schema and then populate the database with existing datasets; reuse a sufficiently complex database for which no ERD exists (requiring you to reverse-engineer the schema); ...

Milestone 2: Ontology Engineering

To achieve semantic interoperability between the different systems, organizations will eventually need to reach a common agreement on relevant concepts and relations in the form of an *ontology*. For this milestone, groups may assume that they have an external need to share (parts of their) data with unknown stakeholders. **Groups may also decide to work together on a common ontology (*)**.

The deliverables for this milestone are

- One or more **ontologies** (depending on your approach) with
 - A well-chosen namespace;
 - All concepts, relations, and -- if need be -- ontology individuals;
 - Human-readable labels and definitions. You do not need to provide such labels and definitions in multiple languages; English labels and English as default labels suffice. Human-readable definitions are also called *glosses*. For more information on how to create such glosses, please consult <https://dl.acm.org/doi/10.1145/1135777.1135850>.
- A **technical report providing** information on
 - Your design decisions, for instance. E.g., the choice and motivation for the OWL 2 profile you adopted. In the case of OWL 2 profiles, indicate how you stayed within a particular profile.
 - Your choice and motivation for naming and organizing concepts, instances, etc., in the ontology.
 - A description of the ontology-engineering process. Elaborate on the process, organization within and outside your group, the problems you encountered, and how you solved them. Highlight both technical and non-technical aspects. The former includes the tooling; the latter includes conflict resolution and the distribution of tasks.
- Optionally, you may generate **ontology documentation** with WIDOCO.¹ This software takes as input an ontology and some parameters (e.g., title, authors, abstract, etc.) to generate different representations (both for humans and software-based agents) according to Linked Data principles. You do not need to implement the content-negotiation rules for your ontology; WIDOCO takes care of that for Apache2 Web servers. This will come in handy if you intend to deploy your ontology (locally, that is) for Milestone 4.

(*) If you decide to work together as a group, there are two scenarios. Either you share the whole ontology or a common ontology that each group needs to extend or include in their ontology. Include all students involved in this exercise as authors in the ontology (using ontology annotations). Groups are expected to write their technical papers, though I recognize that, in this case, some paragraphs may be written together. If that is the case, include these parts in your technical report with a footnote "Written together with X, Y, Z."

¹ <https://github.com/dgarijo/Widoco>

Milestone 3: Annotating your IS (i.e., creating mappings).

For this part of the project, you will use the resulting ontology for committing and annotating your database with the ontology using mappings (with R2RML or RML) and using the RDF generated by these mappings.

Much like the choice and motivation for organizing and naming components of your ontology, you will describe your approach and motivation for:

- The organization and naming strategies of your mappings.
- The organization and naming strategies for the generated RDF.

The deliverables for this milestone are

- A **final technical report** on the development and use of the mappings. Remember to describe the process, both in technical and non-technical terms. Remember that you will need to think about knowledge organization strategies, IRI strategies, etc., while developing these mappings. Elaborate on your choices and their motivations.
- Your mappings.

Milestone 4: Using the Ontology and Knowledge Graph

The final milestone consists of three parts:

1. A deployment of your ontology and knowledge graph, which is necessary to support the next two parts. The knowledge will inform your deployment- and naming strategies you have proposed in the previous milestones.²
2. You must demonstrate that you can answer complex questions using the knowledge graph using some non-trivial SPARQL queries. With non-trivial, I mean queries that answer sufficiently interesting and complex questions that require (among others) subqueries, federated queries, paths, aggregates, and negation as failure.
3. As for the non-trivial demonstrator(s), you will seek an appointment with the professor to discuss the feasibility and scope. Groups have some liberty in choosing these demonstrators. Depending on the size of the group or the complexity of the chosen demonstrators, groups may develop one or two. You are allowed to
 - Look for topics of interest that you wish to explore (e.g., big data processing for knowledge graphs);
 - Look for (research) topics in knowledge graphs (e.g., investigating topics not covered in class, or try and compare other initiatives);
 - Apply topics we've covered in class and critically reflect on that in class (e.g., the use of SHACL or rule languages).

The non-trivial demonstrator is where groups actively look for information and conduct a small (and likely very applied) research study.

The deliverable for this milestone is a **technical report** consisting of

- Chapters based on revisions of prior technical reports (optional)
- A set of **SPARQL queries** that answer specific questions (with a description).
- A report on the non-trivial demonstrator. The nature of this “chapter” depends on the topic(s) chosen. Do not forget to include references, if any.

You can structure your report any way you want as long as all aspects are considered. Suppose you do not know how to structure your technical report. In that case, it may be structured as follows:

- Titlepage
- Abstract
- Introduction
- Information System
 - *Description of your IS and database*
- Ontology and Ontology Engineering
- Mapping Relational Databases to RDF
- Deployment and Demonstration
 - *How do you use or deploy the ontology and RDF?*
 - *Non-trivial queries.*
- Non-trivial Demonstrator(s)
- Conclusions
- References
- Appendix

Be careful, however: chapters that only contain one page are to be discouraged (except for the Introduction and Conclusions).

² You may, of course, revisit those if you were to encounter problems.

Presentation

The goal of the presentation is for the group to present their work and answer a series of questions. This presentation and “defense” will take place during the semester. You will use the questions and feedback to improve, clarify, and rectify elements in your project and report.

The presentation includes a demo of your KG and non-trivial demonstrator. **You may submit or present a video of the demo.**