Modern software applications need to manage and access increasingly complex data, which often does not conform to rigid schemas used in traditional databases. Graph databases and Description Logics (DLs) provide complementary tools for managing such loosely-structured information. For example, an ontology written in a DL can be used to capture knowledge about a given application domain in a machine readable format. This knowledge can then be used for answering ontology-mediated queries (OMQs): a user query is evaluated not directly over the available data, but a reasoning engine is employed to incorporate the knowledge in the ontology, possibly inferring facts that are not explicitly present in the data.

OMQs and queries over graph databases have many similarities, but they also have major differences that we aim to reconcile in this project. In particular, when answering OMQs one makes the so-called open-world assumption (information is assumed to be incomplete), while the closed-world assumption is made in graph databases (information completeness is assumed). It is acknowledged however that both assumptions are too strong, and that the users should have the ability to specify which part of the data is to be assumed as complete and which as incomplete. Second, OMQs do not support navigational features that are very common in languages for querying graph databases and other representations of loosely-structured data (e.g., the path expressions in the XPath query language for XML documents, or the property paths in the SPARQL query language for RDF data). In this project, we will consider extensions of standard OMQs with features that overcome the above two drawbacks, i.e. we will study OMQs that allow to mix the open-world and the closed-world assumptions, and that simultaneously support navigational features. For such rich OMQs, we will study the following fundamental database-theoretic properties:

- The first major goal is to compare the expressive power of the considered OMQs with the expressive power of classic database query languages. To this end, we will mainly develop translations from OMQs into suitable variations of Datalog. This will also enable us to transfer existing results from the setting of Datalog to the setting of OMQs, e.g., to reuse existing efficient Datalog engines for answering OMQs.

- The second major goal is to study the computational complexity of reasoning tasks to support the design of OMQs. In particular, we will investigate the query containment and the query emptiness problems for the OMQs introduced in the project. These static analysis problems lie at the core of query design and optimization techniques.