Department of Computer Science and Information Systems

Salvatore Rapisarda srapis01@dcs.bbk.ac.uk

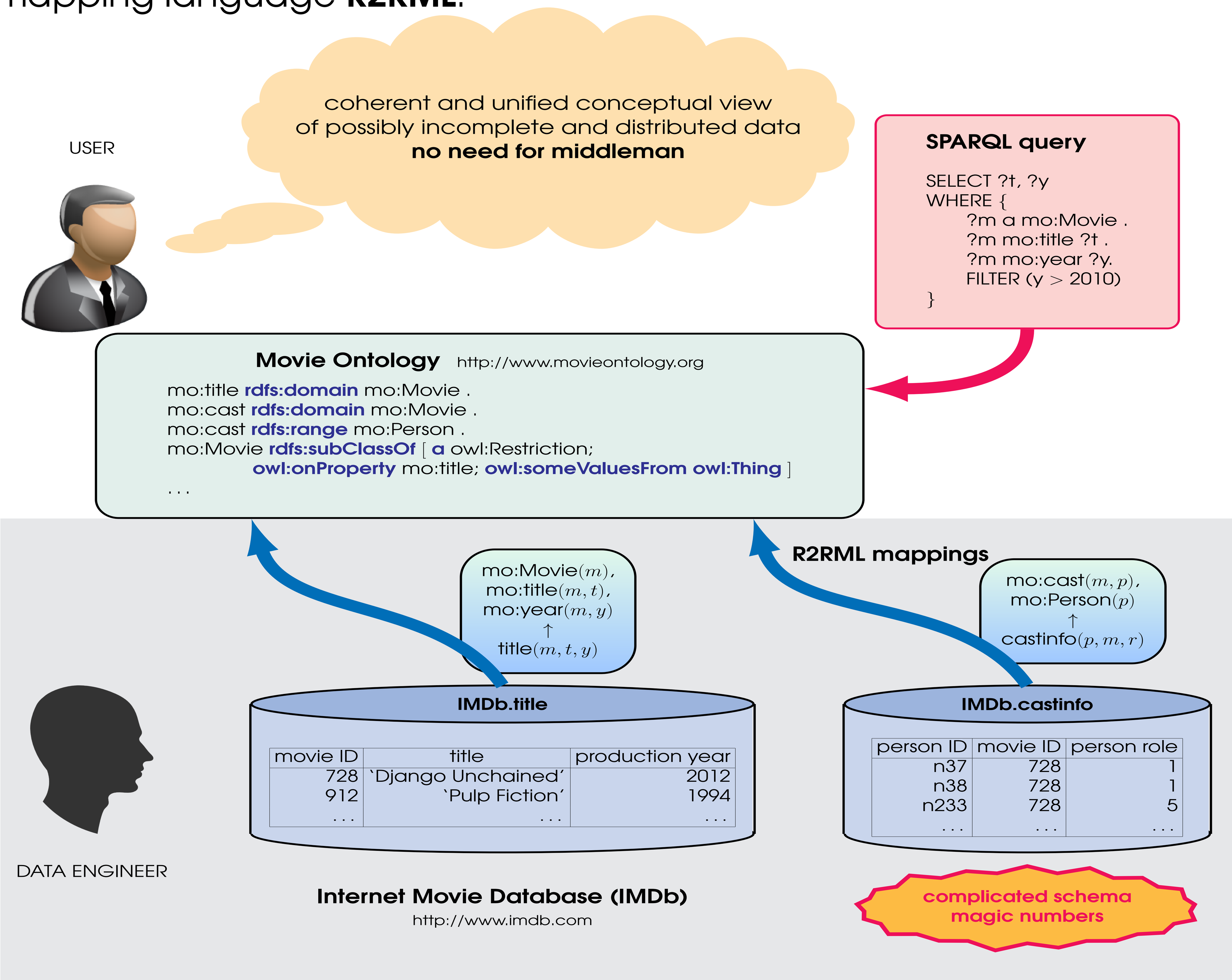
Supervisors: Michael Zakharyaschev, Roman Kontchakov

Ontology-Based Data Access (OBDA)

Data is stored in a **relational database** D. However, the schema of D is often complex, unfamiliar to the user, and does not reflect their information needs.

The user is provided with an **ontology** $\mathcal{T}$ created by a domain expert and defining a vocabulary that is convenient for the user. The vocabulary of $\mathcal{T}$ is used to formulate **SPARQL queries**.

The ontology is related to the data source D via a mapping $\mathcal{M}$, which is formulated in the RDB-to-RDF mapping language **R2RML**.



Query-rewriting approach to OBDA:

1. **Rewrite** Q into a first-order (FO) query $q'(\vec{x})$ such that

$$\mathcal{T}, \mathcal{M}(D) \models q(\vec{a}) \quad \text{iff} \quad \mathcal{M}(D) \models q'(\vec{a}),$$

for any data instance D and any tuple $\vec{a}$ of constants from D, where $\mathcal{M}(D)$ is the result of applying the mapping $\mathcal{M}$ to D.

2. Use the mapping $\mathcal{M}$ to **unfold** $q'(\vec{x})$ into an SQL query $q^*(\vec{x})$ such that

$$\mathcal{M}(D) \models q'(\vec{a}) \quad \text{iff} \quad D \models q^*(\vec{a}).$$

FO-rewritings of conjunctive queries are known to exist for ontologies $\mathcal{T}$ formulated in the **OWL 2 QL** profile of the Web Ontology Language **OWL 2**.

Ontop is a state-of-the-art **OBDA** system developed in collaboration between the Free University of Bozen-Bolzano and Birkbeck, University of London

Key Features

✓ $\mathcal{T}$**-mappings** ensure shorter SQL rewritings

✓ **Semantic Query Optimisation (SQO)**: database integrity constraints are used extensively to optimise rewritings

Query Transformation

$\mathcal{T}$**-mapping** $\mathcal{M}^{\mathcal{T}}$ is obtained by combining the ontology $\mathcal{T}$ and the mapping $\mathcal{M}$ at the offline stage

Virtual ABox $\mathcal{A}'$ can be thought of as the result of applying the $\mathcal{T}$-mapping $\mathcal{M}^{\mathcal{T}}$ to data D (the virtual ABox, however, is not materialised)

The process of transforming a SPARQL query into SQL is as follows:

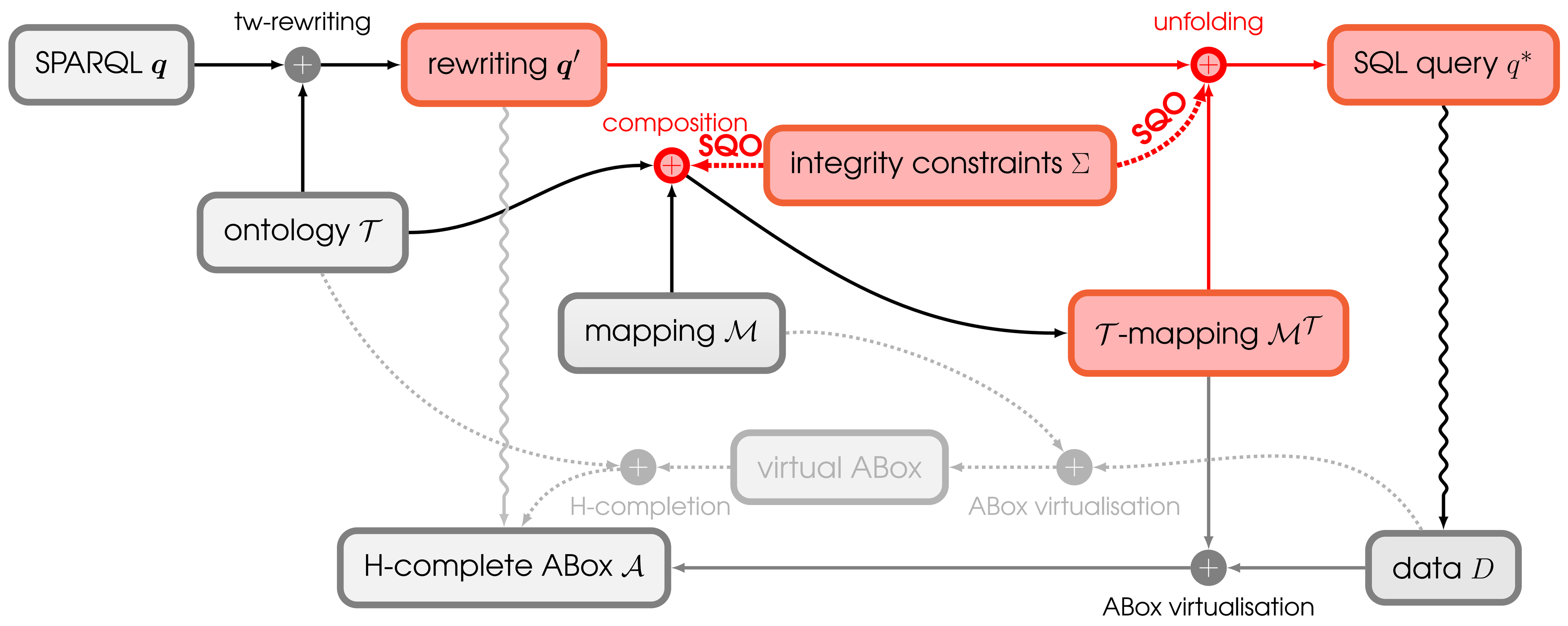
1 user query $q(\vec{x})$ is rewritten into a first-order query q' over the virtual ABox $\mathcal{A}'$

2 the rewritten query q' is unfolded using the $\mathcal{T}$-mapping into an SQL query q^*

3 the optimised SQL query q^* is evaluated over the data instance D

> Semantic Query Optimisation is applied during both the $\mathcal{T}$-mapping construction and query unfolding, and is crucial for avoiding exponential blow-up in practice.

We aim to investigate possible ways of improving the SQO techniques that Ontop implements at the moment. The modifications will affect the components of the system highlighted in the diagram in red.



SQO features and improvements for ontop

– Implement new versions of the **SQL parser** and the **mapping analysis module** for Ontop.

– Improve the current implementation of **SQO** in Ontop: Foreign Keys (**FKs**), Primary Keys (**PKs**), Unique Constraints (**UCs**) and **OR**. Support for **LEFT JOIN**, **NULL** values and **arithmetic expressions**.

– Further advanced SQO techniques, in particular, for queries with aggregation.

[1] R. Kontchakov, M. Rezk, M. Rodríguez-Muro, G. Xiao & M. Zakharyaschev. Answering SPARQL Queries under the OWL 2 QL Entailment Regime with Databases. In Proc. of ISWC 2014, Part II, vol. 8796 of LNCS, pp. 552–567. Springer, 2014

[2] M. Rodríguez-Muro, R. Kontchakov & M. Zakharyaschev. Ontop at Work. In Proc. of OWLED 2013. CEUR, 2013.