Ontology-Based Data Access via Ontop

Asad Ali and MelikeSah

Department of Computer Engineering,

Near East University, North Cyprus via Mersin 10 Turkey

Abstract:Ontology Based Data Access (OBDA) is an emerging research area to query relational databases via ontology, to which these databases are mapped. The ontology depicts the conceptual representation of the domain of interest which along with data sources and corresponding mapping allows us to translate the SPARQL queries into the data level queries and executed by the underlying database management systems. This paper uses an open source OBDA system, called Ontop which is developed by Free University of Bozen-Bolzano, for querying virtual RDF graphs with SPARQL.

Keywords: Semantic Web, OBDA, Ontop, RDF, Database, Ontologies, Mapping, SPARQL, Inference

1. Introduction.

Over the past 20 years, companies interact with very large data sources for analysing and fetching statistics from them. Thus the main challenge for IT experts is to reach the “right” information rather than just to get information. The problem is especially more challenging when the knowledge is stored in some heterogeneous data sources. In order to extract and analyse data from these diverse sources, companies invest millions to hire experts.

To cope with this problem, Ontology-Based Data Access (OBDA) was proposed in 2000s, which provides a conceptual layer in the form of an ontologyfor user queries. The ontology defines a shared vocabulary,models the domain and provides a way of accessing data sources using complex user queries. Data sources can be either a relational database or a structured database that uses Semantic Web technologies to define semantics of the data (aka called triple store). Users’ requests are translated into SPARQL queries by the OBDA system and executed over potentially very large (usually relational and federated) data sources [1].

In our work, we present a case study of ontology-based data access using Ontop in education domain. This paper is organised as follows: Section 2 explains Ontop framework and its components. Section 3 describes the mapping components such as data source manager and mapping manager. Section 4 explains the proposed work, results and our inference model. Finally section 5 concludes the paper.

2. **OntopPro Framework**:

Developed at Free university of Bozen-Bolzano, Italy, OntopPro is an open source OBDA system for querying virtual RDF graphs with SPARQL [2]. To summarize Ontop reformulates a SPARQL query into a SQL query that we can send to a legacy relational database**.**The main and widely used standards supported by Ontop framework are shortly discussed below.
2.1: Mapping: Ontop supports both native Ontop mapping language and R2RML. The R2RML is W3C RDB2RDF Mapping Language while the advantage of native Ontop mapping language is that it is easy to learn and use, howeverOntop allows us to convert one type of language to another.

2.2: Ontology: Ontop supports OWL 2 QL ontology language, which is DL-Lite(Description Logic) family of description logics. OWL 2 QL ensures that the queries can be translated into corresponding equivalent queries over the relational data sources.

Ontop is recently extended to support Semantic Web Rule Language(SWRL).

2.3: Data Source: Ontop, today support all major relational data sources which implement SQL 99 like PostgreSQL, H2,MySQL, DB2, ORACLE, and MS SQL Server.

3. OntopPro Plugin:

OntopPro is a plugin for Protégé 4 and Protégé 5 which provides a mapping editor and enables us to use Quest reasoner directly from Protégé. It provides tabs which allow connecting to a relational database,perform corresponding mapping and formulate SPARQL and SQL queries. The following tabs are used inside OntopPro plugin of Protégé.

- **3.1: Quest:** It is a query engine/reasoner which is an important component of the Ontop. When we talk about Ontop, infact it is the Quest that queries the RDBMS via OWL/RDFS ontologies. It is extremely faster than other OWL reasoner which answers queries on the fly.

- **3.2Ontop Mappings tab**: This is the initial tabwhere the user have to deal with while using Ontop and OBDA. Mapping tab contains the following two sub tabs:

3.2.1: Datasource Manager: It allows us to connect to the database using the following credentials.

- Connection URL: URL of the database
- Database User Name: sa
- Database Password: usually keep empty
- Driver Class: The JDBC driver class of the selected database.

3.2.2:Mapping Manager: This tab is utilized to create and edit mappings. It has three sections:

- **Mapping ID**: It is id of mapping, where any namecan be given to the mapping.
- **Source(SQL Query**): In this tab, we use the SQL query to retrieve data from database.
- **Target (Triples Template):** The query here is written as RDF “Subject-Predicate-Object” followed by a period, represented in Turtle syntax. It is explained with example later in section 4.

The Mapping has the following form:

Triple Template ← SQL Query
When we save our ontology, there are overall three files saved on our local drive:
Somename.owl for our ontology
Somename.obda for our mapping, and
Somename.qfor our queries.

3.2.3:Ontop SPARQL : It is the query editor which allow us to write SPARQL queries based on the created mappings in the Mapping manager and retrieve the required answer from database. Quest reasoner must be started before any query is executed.

4. **Proposed work**: This paper presents a small case study of ontology-based data access using Ontop in education domain. We useda small ontology, Student. Owl and H2 in-memory database management system as our data source.

4.1: **Student Ontology**: Class diagram of the student ontology is shown in Protégé 5 (Figure 1). The ontologyhas Teacher and Student and Assistant classes. Assistant is sub class of Teacher. All these classes are subclasses of Person class, which in turn is a subclass of the Thing class as shown in Figure 1. In addition, there are consults, teaches, Istaughtby object properties. Consults is sub-property of Istaughtby.

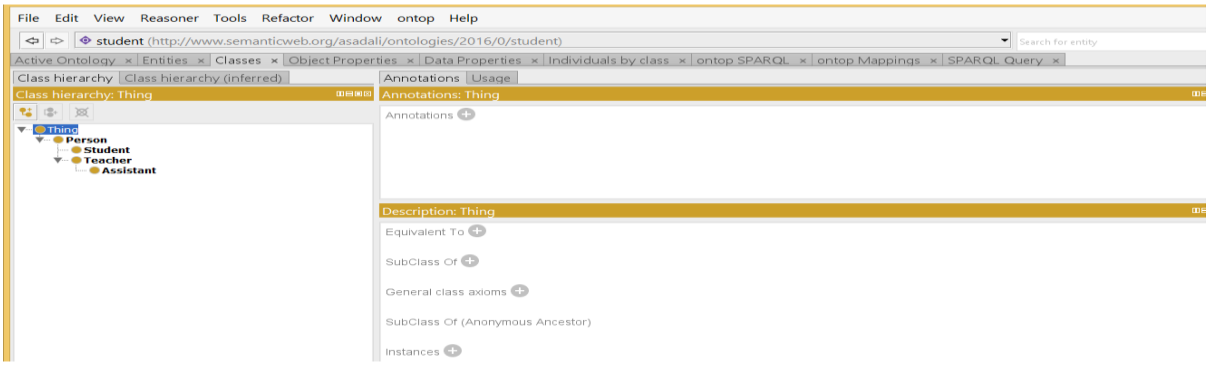


Figure 1. OWL Ontology

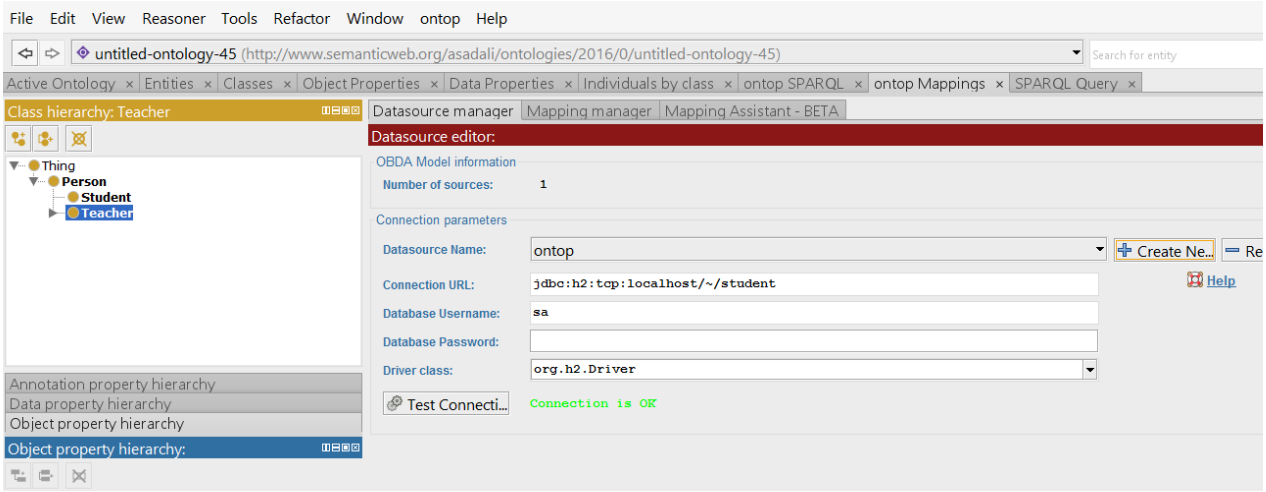
4.2 **H2 Database**: H2 is open-source and fast database system which can be used both as an embedded or server-based. The database has Student, Teacher, Assistant, teaches, and consult tables. Assistant is also a teacher but rather than teaching, he/she assists students in assignments and lab activities and act as a bridge between students and teacher.

The object properties Teaches and consults are taken as database tables here and called Relation tables. The Id and name are the corresponding data properties of the ontology.
The instances will be empty as we will achieve it via database values.
The first step is to build connection with H2 data base which is shown in Figure 2. After making the connection, we have to perform the mapping step, which is important and can be complicated to some extent.
To get Id and name of teachers from Teacher class, the following mapping was created, also shown in Figure 3. The syntax that is recognized by Quest reasoner is Turtle.

:{id} a :Teacher ; :name {name} .

Similarly to extract id and name from Student class, the following Triple template was generated:
:{id} a :Student ; :name {name} .

Also if we want to know who teaches to whom, we will use the mapping of Teaches object property:
{tid} :teaches {sid} .
Where {tid} and {sid} are the column values of Teaches table in the database.


File Edit View Reasoner Tools Refactor Window ontop Help
untitled-ontology-45 (http://www.semanticweb.org/asadali/ontologies/2016/0/untitled-ontology-45)
Search for entity
Active Ontology × Entities × Classes × Object Properties × Data Properties × Individuals by class × ontop SPARQL × ontop Mappings × SPARQL Query ×
Class hierarchy: Teacher
Thing
Person
Student
Teacher
Annotation property hierarchy
Data property hierarchy
Object property hierarchy
Object property hierarchy:
Datasource manager
Mapping manager
Mapping Assistant - BETA
Datasource editor:
OBDA Model information
Connection parameters
Create Ne...
Re
Help
Test Connecti...

Number of sources: 1

Datasource Name: ontop

Connection URL: jdbc:h2:tcp:localhost/~/student

Database Username: sa

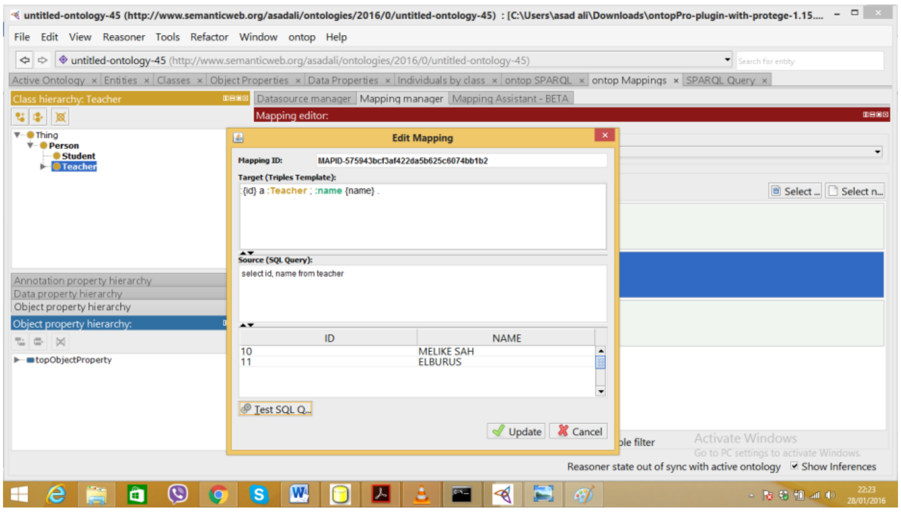
Database Password:

Driver class: org.h2.Driver

Connection is OK

Figure 2. Data source Credentials

:


untitled-ontology-45 (http://www.semanticweb.org/asadali/ontologies/2016/0/untitled-ontology-45) : [C:\Users\asad ali\Downloads\ontopPro-plugin-with-protege-1.15....
File Edit View Reasoner Tools Refactor Window ontop Help
Active Ontology × Entities × Classes × Object Properties × Data Properties × Individuals by class × ontop SPARQL × ontop Mappings × SPARQL Query ×
Class hierarchy: Teacher
Thing
Person
Student
Teacher
Annotation property hierarchy
Data property hierarchy
Object property hierarchy
Object property hierarchy:
topObjectProperty
Datasource manager
Mapping manager
Mapping Assistant - BETA
Mapping editor:
Edit Mapping
Select ...
Select n...
Test SQL Q...
Update
Cancel
Activate Windows
Go to PC settings to activate Windows.
Reasoner state out of sync with active ontology
Show Inferences
22:23
28/01/2016

Mapping ID: MAPID-575943bcf3af422da5b625c6074bb1b2

Target (Triples Template):

```
:{id} a :Teacher ; :name {name} .
```

Source (SQL Query):

```
select id, name from teacher
```

ID	NAME
10	MELIKE SAH
11	ELBURUS

Figure 3. Source/Target Mapping tabs

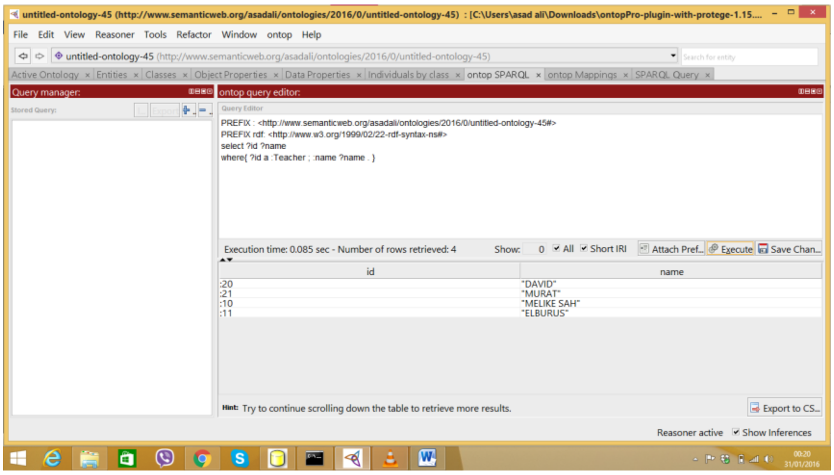
Finally to get the required result we need to create appropriate SPARQL queries in the Ontop query editor. For example the following SPARQL query can be used to retrieve values from Teacher table as shown in Figure 4:

```
Select?id  ?name
Where {?id a :Teacher ; :name ?name . }
```

The Ontop query editor allow us to include the necessary Prefixes like owl,rdf,rdfs etc. However, to get the answer, Quest reasoner must be started before the query execution.

4.3: Relation Table: The object property "Teaches" is called Relation table as describes relation between the Teacher and Student classes. This table contains Tid for Teacher Id and Sid for Student Id which shows who teach whom?

For example, Figure 5 depicts the answer from Teaches table as opposed to Figure 4:



```
PREFIX : <http://www.semanticweb.org/asadali/ontologies/2016/0/untitled-ontology-45#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select ?id ?name
where{ ?id a :Teacher ; :name ?name . }
```

Execution time: 0.085 sec - Number of rows retrieved: 4

id	name
:20	"DAVID"
:21	"MURAT"
:10	"MELIKE SAH"
:11	"ELBURUS"

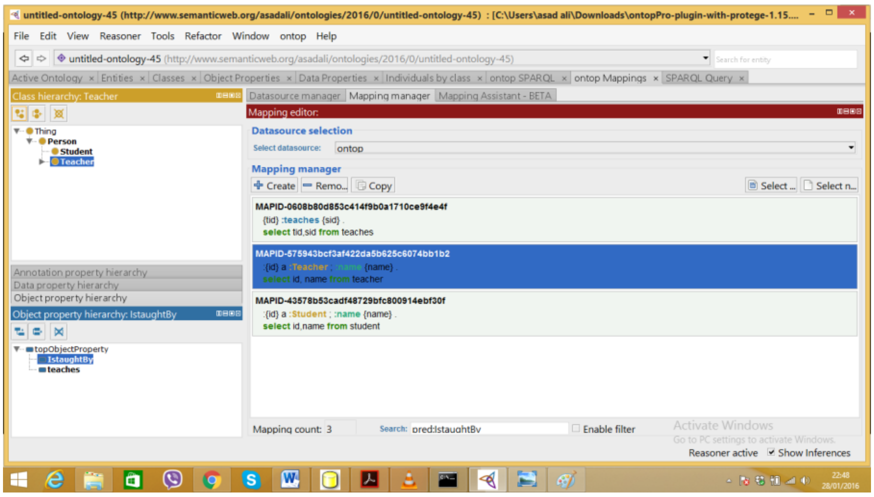
Hint: Try to continue scrolling down the table to retrieve more results.

Figure 4. SPARQL query answer

4.4: Inference: We mapped the object property Teaches and it performs well. However, what about if we have another Object property “IsTaughtBy”, an inverse property of Teaches? Can we get the result of Student taught by a Teacher without mapping for it? Thanks to the Quest, the answer is yes.

As shown in Figure 5, we did not have mapping for the IsTaughtBy property but still we can get the answer as shown in Figure 6.

Also there are only two values in Teacher database i.e. Melike and Elburus but when we fetch values using SPARQL, it gives us four values i.e. David and Murat which are values of the Assistant table. This is because Assistant is the subclass of Teacher in our ontology and the reasoner fetches its values also when we try to get from Teacher class.



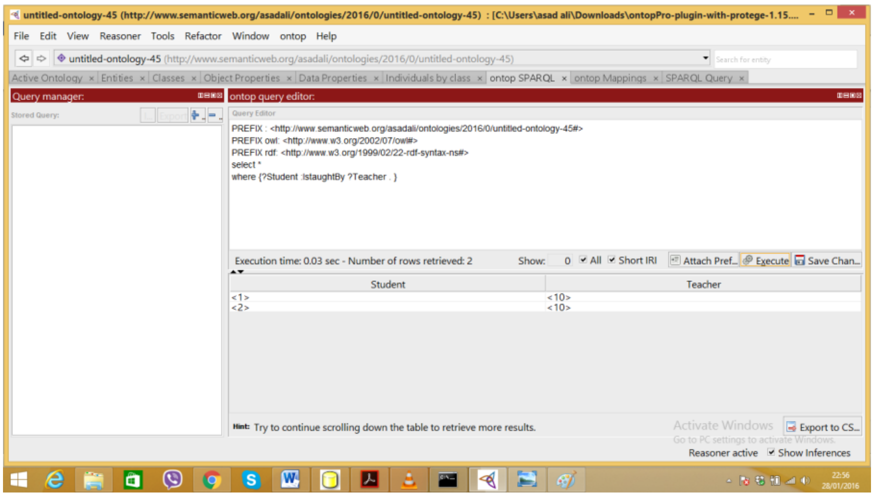
Select datasource: ontop

```
MAPID-0608b80d853c414f9b0a1710ce9f4e4f
  {tid} :teaches {sid} .
  select tid,sid from teaches

MAPID-575943bcf3af422da5b625c6074bb1b2
  :{id} a :Teacher ; :name {name} .
  select id, name from teacher

MAPID-43578b53cadf48729bfc800914ebf30f
  :{id} a :Student ; :name {name} .
  select id,name from student
```

Mapping count: 3 Search: pred:IstaughtBy

Figure 5. Mapping of different Tables

ISSN: 2278 – 1323



untitled-ontology-45 (http://www.semanticweb.org/asadali/ontologies/2016/0/untitled-ontology-45) : [C:\Users\asad ali\Downloads\ontopPro-plugin-with-protege-1.15....
File Edit View Reasoner Tools Refactor Window ontop Help
Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals by class | ontop SPARQL | ontop Mappings | SPARQL Query
Query manager:
ontop query editor:
Query Editor

```
PREFIX : <http://www.semanticweb.org/asadali/ontologies/2016/0/untitled-ontology-45#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select *
where {?Student :IstaughtBy ?Teacher . }
```

Execution time: 0.03 sec - Number of rows retrieved: 2

Show: 0 ☑ All ☑ Short IRI
Attach Pref... Execute Save Chan...

Student	Teacher
<1>	<10>
<2>	<10>

Hint: Try to continue scrolling down the table to retrieve more results.

Export to CS...
Reasoner active ☑ Show Inferences

Figure 6. Inferred SPARQL Result

5. Conclusions: Ontop and Quest allow us to retrieve“what we need” rather than just to get information. Its purpose is to query virtual RDF graphs with SPARQL and allow us to reformulates a SPARQL query into a SQL query that can be sent to a legacy relational database.

6. Future Work: This paper used only OntopPro plugin to query over relational databases. However, in future, we can use the Ontop OWL API and implement as Java application. Similarly, Ontop support only one data source at a time, we can extract data from multiple and heterogeneous data sources using JBoss tool, called Teiid.

7. References:

[1] Diego Calvanese , Benjamin Cogrel, et al, “Ontop: Answering SPARQL Queries overRelational Databases”

[2] Timea Bagosi1, Diego Calvanese, et al, “The Ontop Framework for Ontology Based Data Access” Faculty of Computer Science, Free University of Bozen-Bolzano, Italy Obidea Technology, Indonesia IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

[3] Diego Calvanese et al, “The Mastro System for Ontology-based Data Access” Free University of Bozen-Bolzano, Italy, Sapienza Universita di Roma,Via Ariosto 25, I-00185, Roma, Italy.

[4] Mariano Rodr´ıguez-Muro et al, “Ontology-Based Data Access: Ontop of Databases” Free University of Bozen-Bolzano, Italy, Birkbeck, University of London, U.K..