Available online at www.elixirpublishers.com (Elixir International Journal)

Information Technology

Elixir Inform. Tech. 140 (2020) 54219-54225

ISSN: 2229-712X

Semantic Web Modeling of a High School's Information System along with Sparql Queries

Konstantinos Kotrotsios, Konstantinos Karamitsios and Foteini Emmanouilidou
MyCompany Projects O.E.

ARTICLE INFO

Article history:
Received: 7 February 2020;
Received in revised form:
14 March 2020;
Accepted: 25 March 2020;

Keywords
WebProtege,
Sparql,
Ontology,
School system,
Semantic web.

ABSTRACT

In the first part of this work we will present the modelling of a high school information system with the use of WebProtege. System ontologies and class properties will be presented. In the second part we will present an introduction for SPARQL and examples of queries that were made, with the results returned to us.

Part I

Modeling of High School's Information System

Introduction

For the elaboration of the work was used WebProtege, a web application of collaborative writing and development of OWL ontologies.

OWL is an ontologies' language that is based on the well-known RDF and RDFS, and its initials mean Web Ontology Language, and was created, just like RDF, in order to be interpreted by computers [4]. The difference with RDF is that it is a much richer language with much greater vocabulary and much better capability of interpretation by computers.

There are three versions of OWL: OWL Lite, OWL DL (contains Lite) and OWL Full (contains DL) [1].

The composition OWL is based on RDF and RDFS languages which are based on the "triple" of RDF/XML composition [2]. The key features of an OWL ontology's composition are:

- The header
- The class
- The instances of the classes
- The properties

Header

The header is essentially the root of the ontology and is defined as a rdf:RDF element that specifies a number of namespaces. The namespaces used in the rdf:RDF tags exist to identify the vocabulary of other tags in onrder to be used later in the ontology. To understand this, we will use an example of rdf:RDF header

<rdf : RDF
xmlns= "http:// example.org /Example#"
xmlns:exd="http://example.org/Example#"
xmlns:owl= "http://www.w3.org/2002/07owl#"
xmlns:rdfs="http://www.w3.org/2000/01rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd= "http://www.w3.org/2001/XMLSchema#" >

In our example, the *xmlns = "http:// example.org /Example#"* and *xmlns:exd ="http://example.org/Example#"* are our default namespaces.

Class

An OWL class is expressed in RDF/XML through an owl:Class element. The owl:Class tag contains the statement of an rdf:ID element that locally identifies the class name in this ontology file.

<owl:Class rdf:ID="Object">
</owl:Class>

Various additional elements are particularly important for determining the class.

An rdfs:subClassof element allows an abstract clustering to be subdivided into smaller groups. This shows that all members of the class declared are also members of the superclass defined by rdf:resource.

As shown in the example, 'Person' is defined as a subclass of 'Object', which means that all 'Persons' are 'Object' at the same time.

<owl:Class rdf:ID="Person">
<rdfs:subClassOf rdf:resource="#Object"/>
</owl:Class>

An owl:oneOf element combined with rdfs:subClassOf can be used to define a class by deregulating instances that belong to this class.

In the example shown below the owl:Class "Season" may be just one of the 'Spring', 'Summer', 'Fall', and 'Winter' instances.

Tele:
E-mail address: kotrotsios@mycompany.com.gr

```
<owl:Class rdf:ID="Season">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <Season rdf:ID="Spring"/>
        <Season rdf:ID="Summer"/>
        <Season rdf:ID="Fall"/>
        <Season rdf:ID="Winter"/>
      </owl:oneOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

Let us note that classes can also be anonymous. In the example above, the 'Season' owl:Class element is anonymous. Also the owl:Restriction element also creates anonymous classes.

Also the owl:equivalentClass declares that this class is the same as another. Example: ('PoliticalDivision' is the same as 'AdministrativeBoundary')

```
<owl:Class rdf:ID="PoliticalDivision">
      <owl:equivalentClass
rdf:resource="#AdministrativeBoundary"/>
</owl:Class>
```

Instances

Class instances are defined by specifying the class of which they are instances. For example, the following statement sets an instance with ID 'George' of the Person class.

```
<Person rdf:ID="George"/>
```

Properties

Of course we could not create a meaningful ontology by simply associating classes. So using the ontology properties we display elements and specific facts about class members. It is a binary relationship and is expressed by two types of properties:

• datatype properties
• object properties

Owl Datatype Properties

An owl:DatatypeProperty element expresses the relationship between an instance and a given value. As the example shows, the 'hasAge' property is declared with a value for the 'Person' instance.

```
<owl:DatatypeProperty rdf:ID="hasAge">
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
<Person rdf:ID="Joe">
  <hasAge>32</hasAge>
</Person>
```

Owl Object Properties

An owl:ObjectProperty element expresses the relationship between two instances. As shown in the example, the definition of ObjectProperty as 'hasWife' is used to declare a 'Male' value.

```
<owl:ObjectProperty rdf:ID="hasWife">
  <rdfs:domain rdf:resource="#Man"/>
  <rdfs:range rdf:resource="#Woman"/>
</owl:ObjectProperty>
<Man rdf:ID="Joe">
  <hasWife rdf:resource="#Susan"/>
</Man>
```

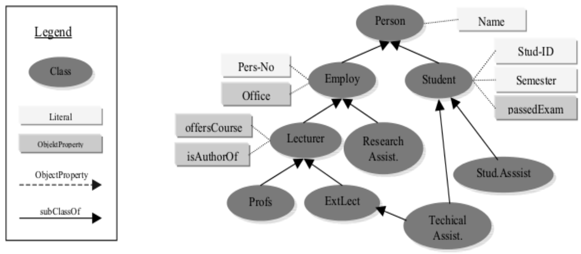


Figure 1. Ontology Example.

In the figure above we see an example of a simple ontology showing all of its elements [5]. Embedded XML Schema datatypes including the known integer, string, boolean, time and date types can be used for property types.

WebProtege

The process of writing ontologies in OWL language is greatly simplified by WebProtege [3]. Having created an interface through an open source web application enables you through collaborative writing to create the classes and properties with just a few clicks.

In this way you can collaboratively write an ontology with your team, leaving comments or notes to other members of the group as well as leaving your project free for use by others.

WebProtege offers you a multitude of options when it comes to creating ontologies. Some of these support OWL 2, a default working environment that provides access to frequently used OWL structures, full change and history tracking for all team members, customizable interface and support of multiple formats for downloading ontology or uploading another in the web environment.

Ontologies

Introduction

In the present work we attempted to implement an ontology for modelling a high school/secondary school information system. We used webProtégé as an ontology development tool. Then we introduced the ontology in Protégé 4.3.0 to get a better graphic representation.

Ontology Description

For the modelling of the high school/secondary school information system we considered that:

- The classrooms belong to the school.
- A class corresponds to each classroom.
- Students belong to a class.
- Teachers have a specialty.
- Teachers teach lessons.
- The lessons correspond to a class.
- Students attend-belong to classes.

Class Hierarchy

At the top of our hierarchy are nine classes: Course, Punishment, Person, TeacherFaculty, SchoolTrip, ClassRoom, School, GlassGrade and Accolade which were implemented by Protégé as Thing class subclasses.

Next, we have developed the Parent, Staff and Student classes as subclasses of the Person class. A person can be either a student or parent, or belong to the school staff. From the Staff class we created two new classes that inherit its attributes, the OtherStaff and Teacher classes. Of which school staff can be either an educator or have another capacity.

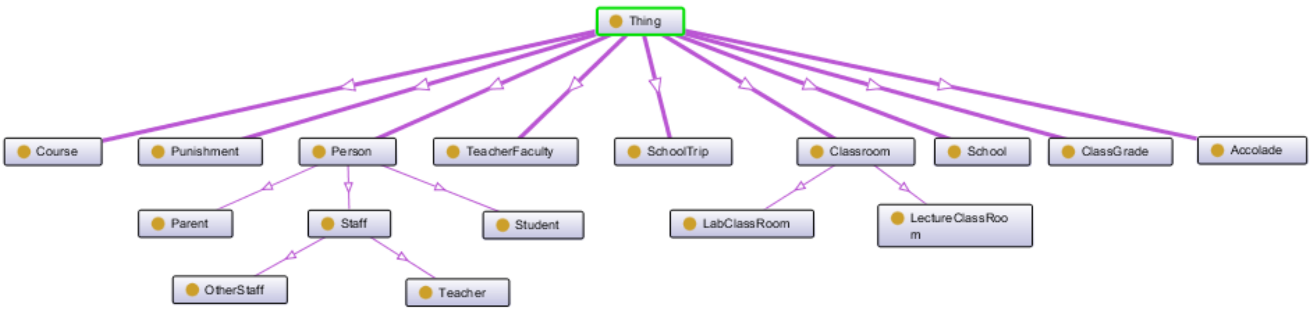


Figure 2 . Representation of ontology's hierarchy.

The Classroom class inherits the features of the other two classes, LabClassRoom and LectureClassRoom. In which a room can be either a lab or a lecture room.

Figure 2 shows the hierarchy of classes.

Object Properties

Object properties in Protégé refer to the characteristics of the classes that receive instances of other classes as values, through which the classes are linked. In the ClassRoom class we have two object properties, classInSchool and classroomHasClass (Figure 3).

Figure 3 . Object Properties Class Classroom.

With the classInSchool property we declare that the Class Room class must have a School class object, which will be the school that the classroom belongs to. The classInSchool property has the Functional attribute because the classroom can only belong to one school. The classroomHasClass property accepts an object of the ClassGrade class that describes which class is hosted in that classroom.

In the School class we have an object property, SchoolHasClasses (Figure 4).

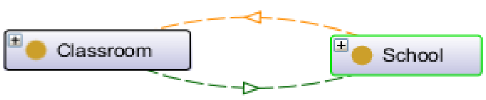


Figure 4 . Object Properties Class School.

Which accepts Classroom items that show which classrooms the school has Class Grade class declares school class (1st grade Secondary School, 1st grade High School, etc.). In Class Grade we have three object properties: classSetInClassroom, going School Trip, has Courses (Figure 5).

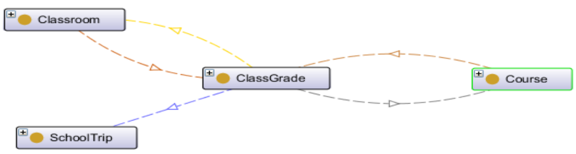


Figure 5 . Object Properties Class ClassGrade

The classSetInClassroom property accepts an object of the Classroom class that indicates which classroom the class is housed in. The going School Trip property is associated with the School Trip class and indicates that a class participated in an excursion. Finally, with the has Courses property, Course class objects are declared that show the lessons a class has.

The Parent class refers to the guardians of the students. It has a guardian Of property (Figure 6) that is associated with the Student class and indicates of which students they are guardians.

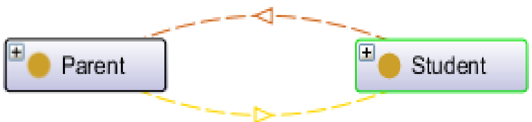


Figure 6 . Object Properties Class Parent

Student class refers to students and has four properties: has Accolade, has Punishment, study, guardian By (Figure 7). The has Accolade and has Punishment properties refer to objects in the Accolade and Punishment classes respectively that refer to whether the student has been praised or punished. The study property is associated with the ClassGrade class and indicates which class the student belongs to. Finally, the guardian By property connects the Student class to the Parent class and refers to the student's guardian.

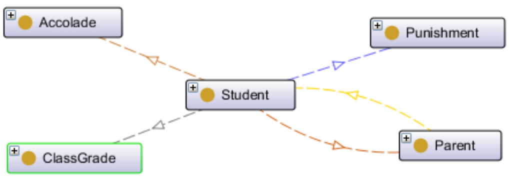


Figure 7 - Object Properties Class Student

The Teacher class has two properties: isSpecialize and teaches (Figure 8). The isSpecialize property points to objects of TeacherFaculty class from which the teacher receives his specialty. The teaches property points to courses of Course class that are the lessons he teaches.

Figure 8 . Object Properties Class Teacher.

Finally we have the Cource class with teached By and teachedInGrade properties (Figure 9). The teached By property accepts Teacher class objects that show which teacher teaches the lesson. The teachedInGrade property accepts ClassGrade objects that show which class each lesson belongs to.

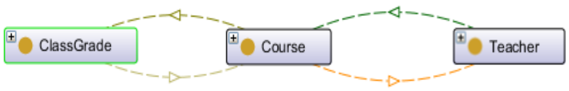


Figure 9 . Object Properties Class Course.

Data Properties

Data properties are the properties of the classes that associate objects with data values (string, integer, date, etc.) and not attributes of other objects such as object properties.

In this section we will describe some of the data properties of the classes we created for our ontology. The data properties we have developed are shown in Figure 10:

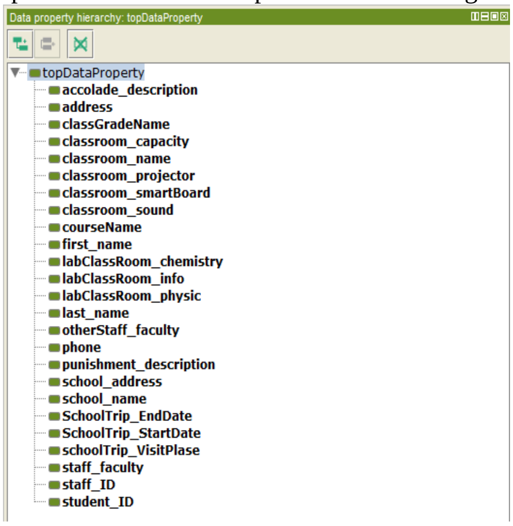


Figure 10 . Data properties.

The accolade_description and punishment_description attributes are alphanumerics (string) belonging to the Accolade and Punishment classes respectively and contain a description of the type of praise and punishment respectively. In Figure 11 we see that the attribute belongs to the Accolade class and is of string type.

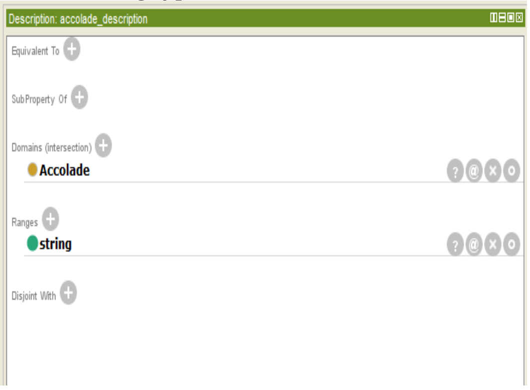


Figure 11 . accolade_description

The address, first_name, and last_name attributes are alphanumeric (string) belonging to the Person class and contain values for the address, name and surname of the persons. Parent, Student, Staff, OtherStuff, and Teacher classes inherit these attributes from the Person class.

Figure 12 shows that address attribute belongs to the Person class and is a string type.

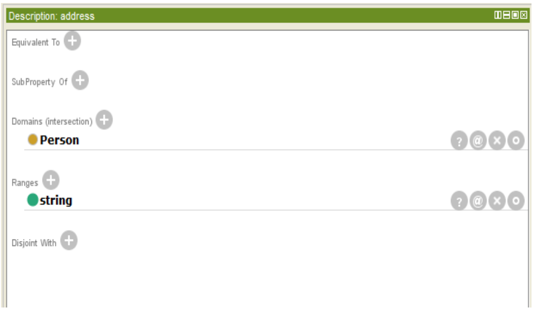


Figure 12 . address

The classroom_projector, classroom_smartBoard, classroom_sound, lab Class Room_chemistry, lab Class Room_info and labClassRoom_physic attributes are boolean, so they can take true or false values and describe how a class can be used. In Figure 13 we see the Classroom_projector attribute that belongs to the LectureClassRoom class and is a boolean type.

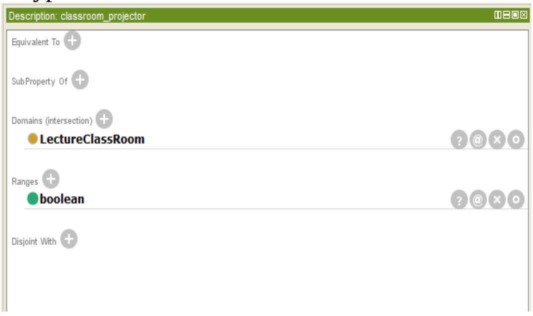


Figure 13. Classroom_projector.

Individuals

Individuals are instances of classes. For our own ontology, we will present some indicative instances to describe how the modelling of a high school/secondary school information system works. Figure 14 shows a part of the instances we have created.

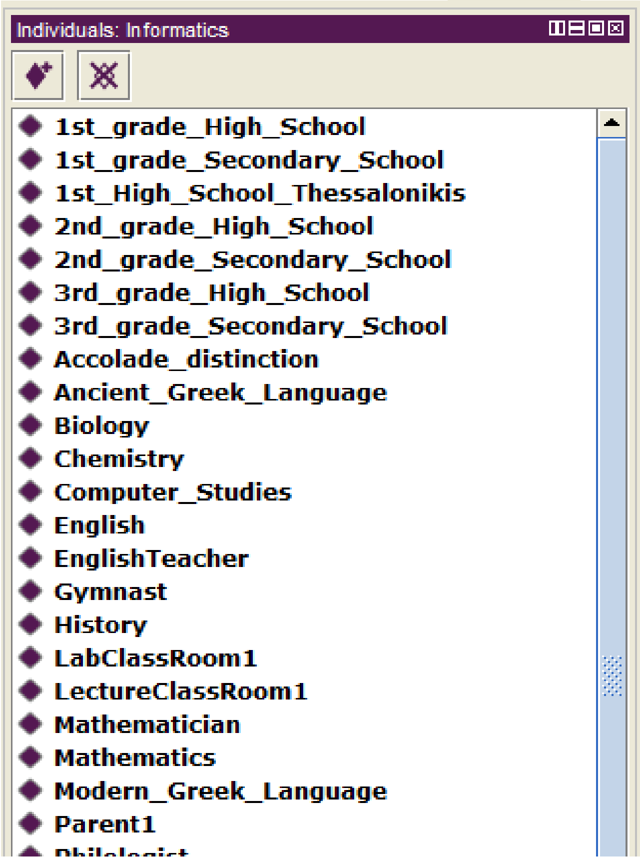


Figure 14 . Individuals.

We originally created an instance of the school class called 1st_High_school_Thessalonikis. The characteristics of the instance are shown in Figure 15.

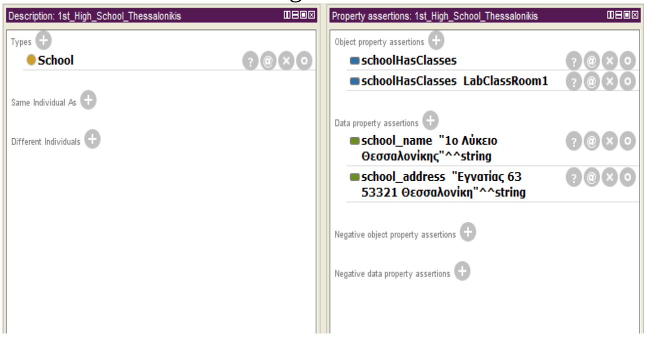


Figure 15. 1st_High_School_Thessaloniki.

We see that the instance we created belongs to the School (Types) class. It has schoolHasClasses as object properties, in which it accepts two instances: LabClassRoom and LectureClassRoom. Also, as data properties, it is given two string values for school_name and for school_address.

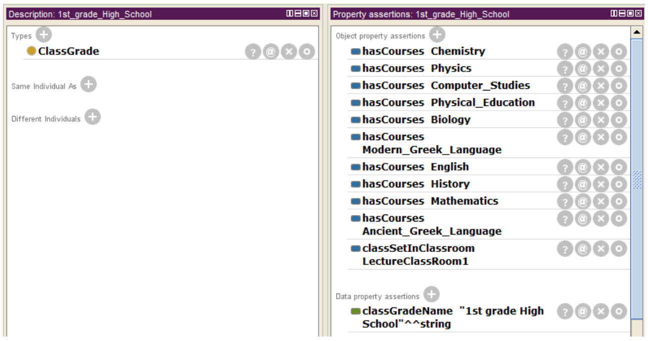


Figure 16 .1st_grade_High_School.

The instances we have created from the ClassGrade class are1st_grade_High_School,2nd_grade_High_School,3rd_grade_High_School,1st_grade_Secondary_School,2nd_grade_Secondary_School, and 3rd_grade_Secondary _School. The features of the 1st_grade_High_School instance are shown in Figure 16.

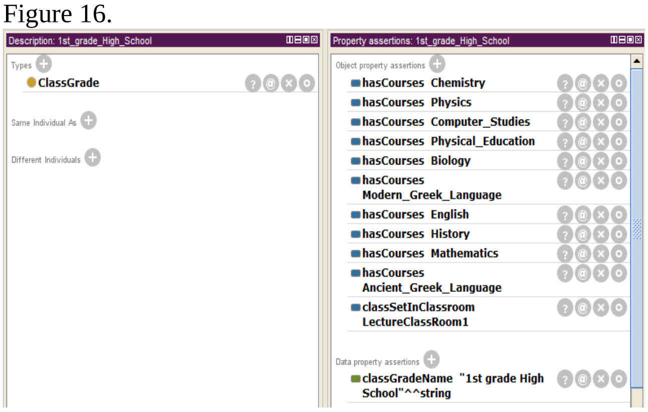


Figure 16 . 1st_grade_High_School.

We see in the 1st_grade_High_School instance features that it has as object properties the hasCourses in which it receives instances of the Courses class, and the classSetinClassRoom in which it receives instances of the ClassRoom class. To classGradeName data properties which is alphanumeric (string), has been given the value "1st grade high school". One of the highlights of the Course class we have created is Biology (image 17).

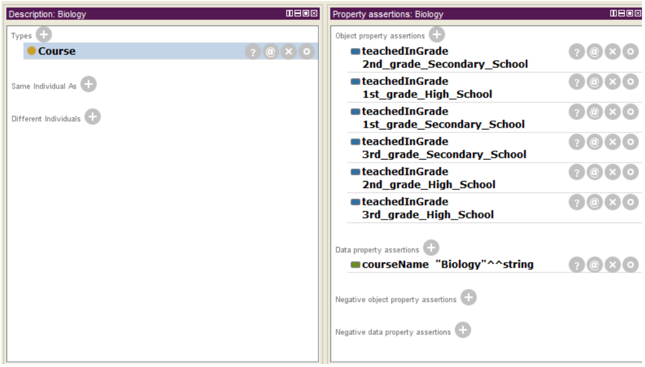


Figure 17. Biology

This instance contains attributes of the teachedInGrade type in which we have instances of GlassGrade class and in addition we have the courseName attribute which is alphanumeric and contains the name of the course.

From the Student class we have created multiple instances. Specifically the instance Student1 (image 18).

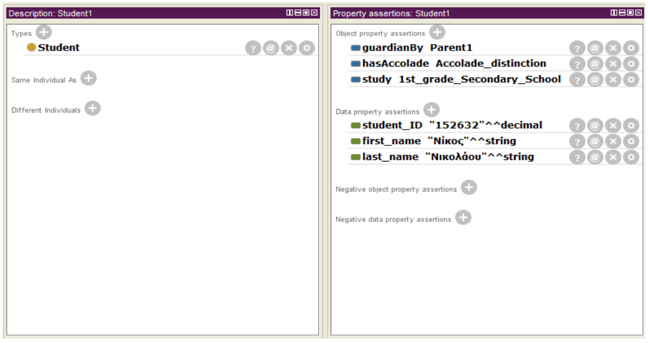


Figure 18. Student.

We see that it has GuardianBy features that contain a Parent class instance with parent1 value which is the pupil's guardian, hasAccolade instance of Accolade class, that is, the student has been awarded, a study instance of classGrade class showing which class the pupil attends. We also have values for the student's name, surname and code.

Finally, from the Teacher class instances in Figure 19 we see the instance teacher1.

Where it has teaches as attributes that are instances of the Courses class and show us what lessons the particular teacher teaches, the isSpecialized attribute is of the TeacherFaculty type and contains the teacher's specialty.

Reasoner

We have used Pellet for the control of ontology. An automated OWL-DL reasoning program, written in java. It collaborates with the Jena framework and is the primary solution for applications written in java. It has the ability to classify ontologies together with their relationships. It also has a built-in engine for SPARQL.

We first executed the pellet with the classify parameter image which shows the class hierarchy as a result.

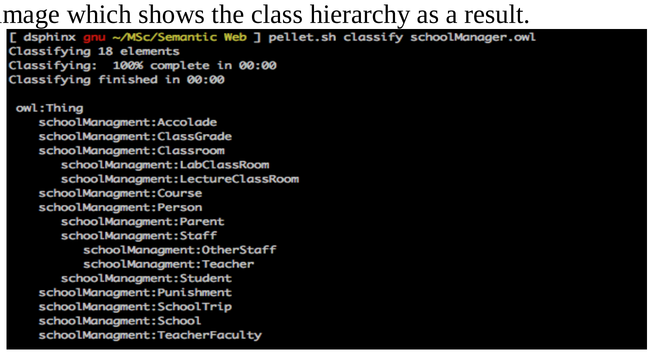


Figure 20 . Pellet classify.

With the consistency parameter we checked the consistency in our ontology image.

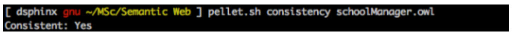


Figure 21 . Pellet consistency.

With the info parameter we see information for ontology (Figure 23).

Figure 22. Pellet info.

Finally, with the lint parameter we check if there are problematic classes (Figure 23).

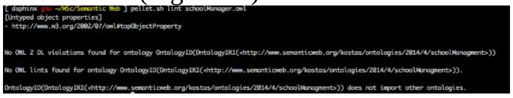


Figure 23 . Pellet lint.

Part II SPARQL

SPARQL (Simple Protocol and RDF Query Language)

SPARQL (Simple Protocol and RDF Query Language) is the official query language for RDF. It was standardized by the W3C RDF Data Access Working Group and in 2008 the W3C was formally established for SPARQL 1.0. SPARQL is for RDF precisely what SQL is for relational databases, or what XQuery is for XML. And because the Semantic Web is based on RDF knowledge representation, SPARQL and its related protocols are paramount in it.

Trying to use Semantic Web without SPARQL is like trying to use a relational database without SQL [6].

SPARQL queries are based on triple patterns. A triple pattern is the same as a RDF triple, except that one or more of its resources-components are variable. A variable is denoted by ?name or $name, where name is its name. For variables the prefix ? or $ is the same [8].

The SPARQL engine that executes a query searches from all resources, those that verify the query triple patterns, in accordance with the RDF suggestions in the knowledge base with which the SPARQL engine is connected.

SPARQL has many similarities to SQL. A query in SPARQL can contain conjugations, disjunctions, optional limitations, limitations on the number of results, and generally almost all of the basic fundamental elements that SQL has (Figure 24).

There are four SPARQL query formats: SELECT, CONSTRUCT, ASK and DESCRIBE.

- SELECT: These are queries that return the objects in a table format

(resources and verbal) that verify the desired triple patterns

- CONSTRUCT: These are queries that return an RDF graph, according to a graph template included in the query.
- ASK: These are queries that answer whether or not there is a solution for some triple patterns, without stating what that is, if it exists, eg.

ASK
WHERE {
<http://dbpedia.org/resource/School> a?type.
<http://dbpedia.org/resource/School>
<http://dbpedia.org/property/schooltype> "1ο λύκειο"
}

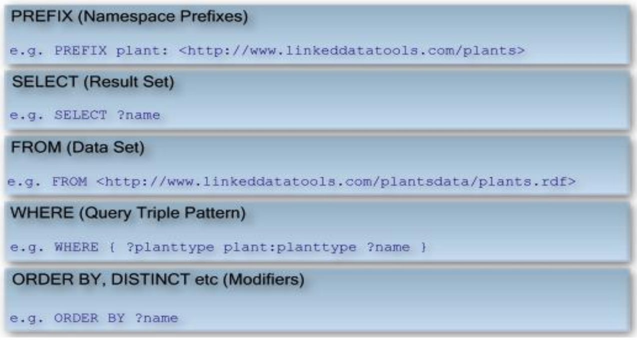
- DESCRIBE: These are queries that return an RDF graph containing data about some of the resources declared in the query, e.g.

DESCRIBE <http://dbpedia.org/resource/Teacher> .

There are SPARQL implementations for many programming languages and tools such as:

- the connection to a SPARQL interface and the semiautomatic construction of a query, and
- the translation of SPARQL queries into other query languages (eg. SQL, XQuery, etc.).

SPARQL 1.1 was designated as W3C Recommendation on March 21, 2013 [7]. [9].

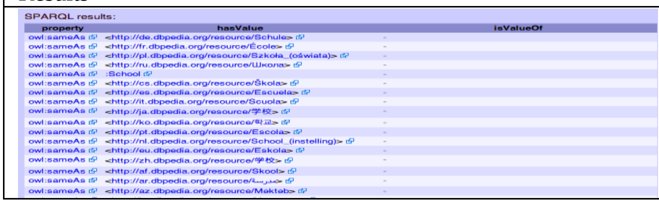


Εικόνα 24 . Τυπική σύνταξη ερωτήματος SPARQL.

ANNEX A - SPARQL QUESTIONS

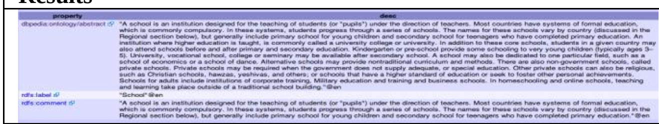
On all queries, the Ontology School was used because the Greek version of dbpedia (el.dbpedia.org) is out of order.

In the exercise it is mentioned that the examples are executed at a DBpedia endpoint, unfortunately there are some restrictions to these, for example: SERVICES outputs *"Virtuoso 42000 Error SQ200: Must have privileges on view DB.DBA.SPARQL_SINV_2"*. We therefore used also different endpoints for the exercise needs.

SPARQL query # 1
Endpoint : http://dbpedia.org/snorql
SPARQL Query
SELECT DISTINCT ?property ?hasValue ?isValueOf **WHERE {** **{ <http://el.dbpedia.org/resource/Σχολείο> ?property ?hasValue }** **UNION** **{ ?isValueOf ?property <http://el.dbpedia.org/resource/Σχολείο> }** **} LIMIT 300**
Description
Displays the ontology names in languages other than Greek (Ontology School).
Results


Ερώτημα SPARQL # 2
Endpoint : http://dbpedia.org/snorql
SPARQL Query

SELECT distinct ?property ?desc
WHERE {
:School ?property ?desc .
FILTER (lang(?desc) = "en")
}

Description
Displays the related information - properties such as abstract, commentary for English.
Results


SPARQL query # 3

Endpoint : http://dbpedia.org/snorql

SPARQL Query

```
PREFIX : <http://dbpedia.org/resource/>
SELECT distinct ?Category ?Concept
WHERE {
  ?Category ?type ?Concept .
   FILTER (?Category IN (:School, :Gymnasium, :Lyceum)) .
   FILTER ( lang(?Concept) = "en" ) .
} LIMIT 10
```

Description

Displays the related information - properties such as abstract, commentary for English. The difference with the previous query is that it looks for Gymnasium and Lyceum beyond the School ontology.

Results

SPARQL results:

Category	Concept
:School	"A school is an institution designed for the teaching of students (or "pupils") under the direction of teachers. Most countries have systems of formal education, which is commonly compulsory. In these systems, students progress through a series of schools. The names for these schools vary by country (discussed in the Regional section below), but generally include primary school for young children and secondary school for teenagers who have completed primary education. An institution where higher education is taught, is commonly called a university college or university. In addition to these core schools, students in a given country may also attend schools before and after primary and secondary education. Kindergarten or pre-school provide some schooling to very young children (typically ages 3–5). University, vocational school, college or seminary may be available after secondary school. A school may also be dedicated to one particular field, such as a school of economics or a school of dance. Alternative schools may provide nontraditional curriculum and methods. There are also non-government schools, called private schools. Private schools may be required when the government does not supply adequate, or special education. Other private schools can also be religious, such as Christian schools, hawzas, yeshivas, and others; or schools that have a higher standard of education or seek to foster other personal achievements. Schools for adults include institutions of corporate training, Military education and training and business schools. In homeschooling and online schools, teaching and learning take place outside of a traditional school building."@en
:Lyceum	"The lyceum is a category of educational institution defined within the education system of many countries, mainly in Europe. The definition varies between countries; usually it is a type of secondary school."@en
:Gymnasium	"Gymnasium"@en
:School	"School"@en
:Lyceum	"Lyceum"@en
:School	"A school is an institution designed for the teaching of students (or "pupils") under the direction of teachers. Most countries have systems of formal education, which is commonly compulsory. In these systems, students progress through a series of schools. The names for these schools vary by country (discussed in the Regional section below), but generally include primary school for young children and secondary school for teenagers who have completed primary education."@en

SPARQL query # 4

Endpoint : http://dbpedia.org/snorql

SPARQL Query

```
SELECT distinct ?Category ?Concept
WHERE {
     ?Category ?type ?Concept .
      FILTER ((?Category IN (:Teacher, :Paraprofessional ))
&& ( lang(?Concept) = "en" )) .
} order by ?Concept
```

Description

Displays related information - properties such as label, comment for English for Teacher and Paraprofessional ontologies sorted by ?Concept variable

Results

Category	Concept
:Teacher	"$43,009 2006-2007 school year"@en
:Teacher	"A teacher or schoolteacher is a person who provides education for pupils (children) and students (adults). The role of teacher is often formal and ongoing, carried out at a school or other place of formal education. In many countries, a person who wishes to become a teacher must first obtain specified professional qualifications or credentials from a university or college. These professional qualifications may include the study of pedagogy, the science of teaching."@en
:Teacher	"A teacher or schoolteacher is a person who provides education for pupils (children) and students (adults). The role of teacher is often formal and ongoing, carried out at a school or other place of formal education. In many countries, a person who wishes to become a teacher must first obtain specified professional qualifications or credentials from a university or college. These professional qualifications may include the study of pedagogy, the science of teaching. Teachers, like other professionals, may have to continue their education after they qualify, a process known as continuing professional development. Teachers may use a lesson plan to facilitate student learning, providing a course of study which is called the curriculum. A teacher's role may vary among cultures. Teachers may provide instruction in literacy and numeracy, craftsmanship or vocational training, the arts, religion, civics, community roles, or life skills. A teacher who facilitates education for an individual may also be described as a personal tutor, or, largely historically, a governess. In some countries, formal education can take place through home schooling. Informal learning may be assisted by a teacher occupying a transient or ongoing role, such as a family member, or by anyone with knowledge or skills in the wider community setting. Religious and spiritual teachers, such as gurus, mullahs, rabbis, pastors/youth pastors and lamas, may teach religious texts such as the Quran, Torah or Bible."@en
:Teacher	"Classroom at a secondary school in Pendembu, Sierra Leone."@en
:Paraprofessional	"Paraprofessional"@en
:Paraprofessional	"Paraprofessional is a job title given to persons in various occupational fields, such as education, healthcare, engineering and law, who are trained to assist professionals but do not themselves have professional licensure. The Greek prefix "para" as used here indicates beside or side by side (as in "parallel"); hence, a paraprofessional is one who works alongside a professional."@en
:Paraprofessional	"Paraprofessional is a job title given to persons in various occupational fields, such as education, healthcare, engineering and law, who are trained to assist professionals but do not themselves have professional licensure. The Greek prefix "para" as used here indicates beside or side by side (as in "parallel"); hence, a paraprofessional is one who works alongside a professional. The paraprofessional is able to perform tasks requiring significant knowledge in the field, and may even function independently of direct professional supervision, but lacks the official authority of the professional. For example, a person who hires a paralegal in lieu of a lawyer to complete bankruptcy forms is still considered to be filing pro se. Some paraprofessional occupations require special testing or certification in the field, while others require only a certain level of education. In some occupations, such as that of teaching assistant, requirements differ geographically although duties are essentially the same. This concept of a paraprofession exists predominantly in the United States; it is not commonly found in the sociology of professions in the UK or Europe, for example."@en
:Teacher	"Schools"@en
:Teacher	"Teacher"@en
:Teacher	"Teacher, schoolteacher"@en
:Teacher	"Teaching abilities, pleasant disposition, patience"@en

SPARQL query # 5

Endpoint : http://dbpedia.org/snorql

SPARQL Query

```
SELECT distinct ?Category ?Concept ?type
WHERE {
?Category ?type ?Concept .
FILTER ((?Category IN (:Lesson, :Teacher, :School,
:Paraprofessional , :Classroom , :Grade,  :Course , :Person,
:Student, :Lyceum, :Gymnasium, :Punishment  )) && (
lang(?Concept) = "en" )) .
} order by ?Concept
```

Description

Displays the related information - properties such as label, comment, abstract for English for the ontologies that would be used in our model. The results are sorted based on the description.

Results

SPARQL results:

Category	Concept	type
:Teacher	"$43,009 2006-2007 school year"@en	dbpedia2:averageSalary
:Classroom	"A classroom is a room in which teaching or learning activities can take place. Classrooms are found in educational institutions of all kinds, including public and private schools, home schools, corporations, and religious and humanitarian organizations. The classroom attempts to provide a safe space where learning can take place uninterrupted by other distractions."@en	dbpedia:ontology/abstract
:Classroom	"A classroom is a room in which teaching or learning activities can take place. Classrooms are found in educational institutions of all kinds, including public and private schools, home schools, corporations, and religious and humanitarian organizations. The classroom attempts to provide a safe space where learning can take place uninterrupted by other distractions."@en	rdfs:comment
:Lesson	"A lesson is a structured period of time where learning is intended to occur. It involves one or more students (also called pupils or learners in some circumstances) being taught by a teacher or instructor. A lesson may be either one section of a textbook (which, apart from the printed page, can also include multimedia) or, more frequently, a short period of time during which learners are taught about a particular subject or taught how to perform a particular activity."@en	rdfs:comment
:Lesson	"A lesson is a structured period of time where learning is intended to occur. It involves one or more students (also called pupils or learners in some circumstances) being taught by a teacher or instructor. A lesson may be either one section of a textbook (which, apart from the printed page, can also include multimedia) or, more frequently, a short period of time during which learners are taught about a particular subject or taught how to perform a particular activity. Lessons are generally taught in a classroom but may instead take place in a situated learning environment. In a wider sense, a lesson is an insight gained by a learner into previously unfamiliar subject-matter. Such a lesson can be either planned or accidental, enjoyable or painful. The colloquial phrase "to teach someone a lesson", means to punish or scold a person for a mistake they have made in order to ensure that they do not make the same mistake again. Lessons can also be made entertaining. When the term education is combined with entertainment, the term edutainment is coined."@en	dbpedia:ontology/abstract
:Person	"A person is a being, such as a human, that has certain capacities or attributes constituting personhood, which in turn is defined differently by different authors in different disciplines, and by different cultures in different times and places. In ancient Rome, the word "persona" (Latin) or "prosopon" originally referred to the masks worn by actors on stage. The various masks represented the various "personae" in the stage play. The concept of a "person" was further developed during the Trinitarian and Christological debates of the first through sixth centuries. Since then, a number of important changes to the word's meaning and use have taken place, and attempts have been made to redefine the word with varying degrees of adoption and influence. In addition to the question of personhood, of what makes a being count as a person to begin with, there are further questions about personal identity: both about what makes any particular person that particular person instead of another, and about what makes a person at one time the same person as he or she was or will be at another time despite any intervening changes. The common plural of "person", "people", is often used to refer to an entire nation or ethnic group (as in "a people"), so the plural "persons" is often used in contexts which require precision such as philosophical and legal writing."@en	dbpedia:ontology/abstract
:Person	"A person is a being, such as a human, that has certain capacities or attributes constituting personhood, which in turn is defined differently by different authors in different disciplines, and by different cultures in different times and places. In ancient Rome, the word "persona" (Latin) or "prosopon" originally referred to the masks worn by actors on stage. The various masks represented the various "personae" in the stage play."@en	rdfs:comment

References

[1] http://www.w3.org/TR/2004/REC-owl-guide-20040210/#StructureOfOntologies (Μαιος 2014)

[2] http://acuity.sourceforge.net/acuitycontrollerwebpages/OwlHowToBasics.html (Μαιος 2014)

[3] http://protegewiki.stanford.edu/wiki/WebProtege (Μαιος 2014)

[4] Antoniou, G., & Van Harmelen, F. (2004). A semantic web primer. MIT press.

[5] Dunkel, J., Bruns, R., & Ossowski, S. (2006). Semantic e-Learning agents. In Enterprise Information Systems VI (pp. 237-244). Springer Netherlands.

[6] http://www.w3c.gr/press/pressreleases/2008/01/sparql-pressrelease.el.html (Μαίος 2014)

[7] http://www.w3.org/TR/sparql11-query/

[8] Toby Segaran, Colin Evans, Jamie Taylor, Programming the Semantic Web, O'Reilly Media, 2009.

[9] Bob DuCharme, Learning SPARQL, "O'Reilly Media, Inc.", 2011