

Web datamining and semantics project

Syrine BOUCHELLEGHEM

Céline BEN MANSOUR

Mohamed Nagy ATALLA

Maikel GALI

TABLE OF CONTENTS

MODELING THE ONTOLOGY.....	i
POPULATING THE ONTOLOGY	ii
QUERYING THE ONTOLOGY	iii
OUR API (How to install, How to use)	iv

Git repository : https://github.com/syrineb/wdatamining_project

MODELING THE ONTOLOGY

To design our ontology using protege editor, we have chosen four sets of data. The data represent some facilities in the city of Argenteuil, as parking, parking for two wheelers, sports fields and parks. It gives us the possibility to finds four types of points of interest (POI).

The data can be found in the following pages :

Parking :

<https://www.data.gouv.fr/fr/datasets/les-parkings-a-argenteuil/>

Parking for two wheelers :

<https://www.data.gouv.fr/fr/datasets/le-stationnement-2-roues-a-argenteuil/>

Sport fields :

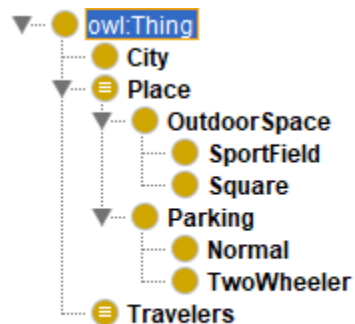
<https://www.data.gouv.fr/fr/datasets/les-terrains-sportifs-de-proximite-a-argenteuil/>

Squares :

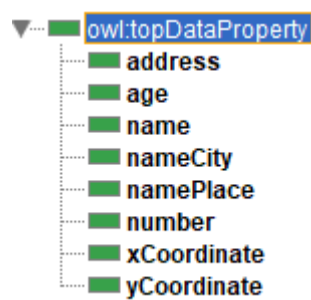
<https://www.data.gouv.fr/fr/datasets/les-squares-a-argenteuil/>

Our ontology was designed as follows :

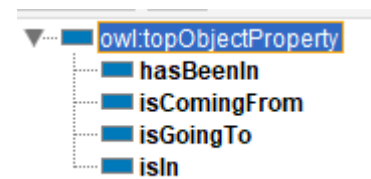
Classes :



Data Properties :

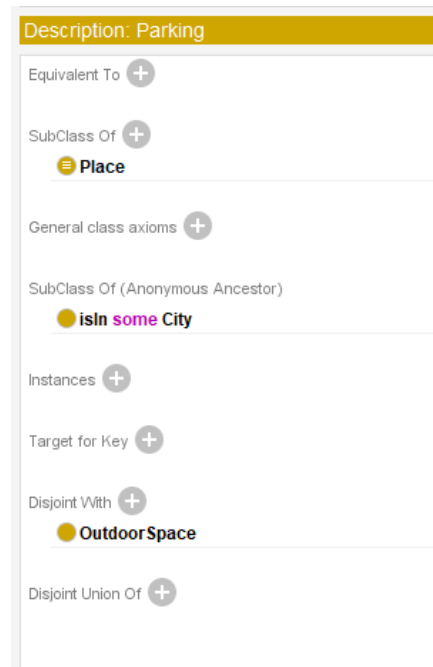


Object properties :



We created some individuals that we added into the ontology, for each class.

Some restrictions were added to the classes we made.
Here is an example.



To use our datasets, we converted the given geojson files into json-ld files.

The vocabulary we defined is based on our ontology. We used the following context to create the json-ld files :

Fields and squares :

```
"@context": {
  "@vocab": "http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#",
  "@base": "http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11/",
  "namePlace": "namePlace",
  "address": "address",
  "xCoordinate": "xCoordinate",
  "yCoordinate": "yCoordinate"
}
```

(terrains.jsonld and squares.jsonld)

Parking and parking for two wheelers :

```
"@context": {
  "@vocab": "http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#",
  "@base" : "http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11/",
  "namePlace": "namePlace",
  "number": { "@id" : "number", "@type" : "http://www.w3.org/2001/XMLSchema#int"},

  "address" : "address" ,
  "xCoordinate": "xCoordinate",
  "yCoordinate": "yCoordinate"
}
```

(parking.jsonld and deuxroues.jsonld)

POPULATING THE ONTOLOGY

To populate our ontology, we set up a triple store using Apache Jena Fuseki containing our triples :

Filter datasets		Clear
name	Actions	
/parking_normal	query add data edit info	
/parkings_deuxroues	query add data edit info	
/squares	query add data edit info	
/terrains	query add data edit info	

```
1 SELECT ?subject ?predicate ?object
2 WHERE {
3   ?subject ?predicate ?object
4 }
5 LIMIT 25
```

subject	predicate	object
1 b0	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#address>	MAIL STENDHAL
2 b0	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#namePlace>	Provence
3 b0	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#xCoordinate>	"2.22382715673505e0"<http://www.w3.org/2001/XMLSchema#double>
4 b0	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#yCoordinate>	"48.9536915017711e0"<http://www.w3.org/2001/XMLSchema#double>
5 b1	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#address>	ALLEE HENRI WALLON
6 b1	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#namePlace>	TSP HENRI WALLON 1
7 b1	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#xCoordinate>	"2.22494596799725e0"<http://www.w3.org/2001/XMLSchema#double>
8 b1	<http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#yCoordinate>	"48.9512678030724e0"<http://www.w3.org/2001/XMLSchema#double>

To make our data available we used our json-ld files to convert it into RDF/XML. (see files in the repository)

QUERYING THE ONTOLOGY

All queries can be found in the file query.txt.

Here is a few queries tested on our ontology :

- List of all travelers that are going to Bapaume parking :

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ns: <http://www.semanticweb.org/syrin/ontologies/2022/2/untitled-ontology-11#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT ?x
WHERE {
    ?x ns:isGoingTo ns:Bapaume.
}
```

x
Yohann

- A query with two alternatives :

```
SELECT ?name ?hasBeenIn ?isGoingTo
WHERE {
    { ?name ns:name "Thomas" . ?name ns:hasBeenIn ?hasBeenIn . }
    UNION
    { ?name ns:name "Marie" . ?name ns:isGoingTo ?isGoingTo . }
}
```

name	hasBeenIn	isGoingTo
Thomas	TSP_ERNEST_BRAY	
Thomas	Parc_Mirabeau	
Marie		TSP_ERNEST_BRAY

- ASK query to see if traveler Sarah is 51 years old :

```
ASK{ ?x ns:name "Sarah";
      ns:age 51
}
```

Result
False

Etc.

We also defined the following SWRL rules for the ontology :

Travelers(?t)^hasBeenIn(?t,?p)^Place(?p) -> isComingFrom(?t,?p)

This rule means that a traveler who has been to a place is necessarily coming from this same place.

OUR API : How to install, How to use

We made our api using Flask with Python. Before trying it, make sure you have Flask installed on your computer.

How to install :

- Download API.zip and unzip
- Open new terminal
- Go to your working directory
- Run **python FlaskApp.py**
- Go to <http://127.0.0.1:5002/>

You are now on the homepage of our app. You can see a map of the city of Argenteuil and the weather forecast in this city. Weather forecast is updated every 24h.

From the homepage you are able to visit other pages via top left buttons to get information about the city. You can search for parking lots for cars, parking lots for two wheelers, sports field and squares. When loading these pages, you can have direct access to the place information such as name, address and capacity for parking, by simply clicking on a marker.