Report

MSc. Data Science & AI

Web of Data

Dr. Catherine FARON

Student: Aglind Reka

Student code: 22207839

Course: Web of Data

Spring term Year

10-04-2023 of submission

Abstract

This report focuses on the models and technologies of the Semantic Web, specifically in creating the foundational elements for a Semantic Web application tailored for a higher education institution like UCA. To illustrate this, I analyzed three masters programs: ”Data Science and Artificial Intelligence,” ”MSC Modelling Of Neuronal And Cognitive Systems,” and ”MSc Biobanks Complex Data Management.”

In particular, I explored the hierarchical structure of these programs, starting with the director, coordinator, Accountant, Administration, Human Resources of each master, followed by the professors and researchers who contribute to the program through their teachings, and ending with the students who are the primary beneficiaries of the program.

The files ”UCA data” and ”UCA schema” comprehensively capture the data, classes, and properties utilized. While ”UCA data” includes the instances of classes along with their associated properties, ”UCA schema” lists the classes and properties employed in the data.

Contents

1 Model Presentation **1**

1.1 University Data 1

1.2 University Schema 1

2 Queries **3**

2.1 SELECT query to find all the properties of Aglind. 3

2.2 Gives to me the description of Aglind, also with the graph 3

2.3 This query indicates for each person if her age is even (true or false) . . 3

2.4 ”contruct and insert” Construct returns a new RDF graph 3

2.5 Creates a relationship between student and professor 5

2.6 classes of the ontology and subClassOf in the ontology 5

3 Conclusion **5**

1. Model Presentation

The goal of this project is to model a University hierarchy (UCA) using the RDF(S) language. We use a mix of Class, Property, label, comment, range, domain, subClassOf, and subPropertyOf.

Prefixes are used as shorthand abbreviations for URIs (Uniform Resource Identifiers) in order to make RDF graphs easier to read and write. Here are the prefixes I used:

```
@prefix : http://ns.uca.fr/humans/schema# .
@prefix d: http://ns.uca.fr/humans/data# .
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# .
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema# .
@prefix xml: http://www.w3.org/XML/1998/namespace .
@prefix xsd: http://www.w3.org/2001/XMLSchema# .
```

This code defines the necessary prefixes for the given RDF(S) graph. Note that the `@prefix` keyword is used to define the prefix and the colon is used to denote the namespace abbreviation. The dot at the end of each line is used to terminate the prefix definition.

1.1 University Data

The data in a Turtle file consists of a series of statements, each separated by a period (”.”). Each statement is composed of a subject, a predicate, and an object, with each component separated by a whitespace and the predicate followed by a semicolon. In my example I will explain the structure of my data.

To begin, I defined the Director as the head of the Master’s program. Then, for each of the different Master’s programs, I created one unique instance of the Director. Every master has an unique director. Next, I proceeded to model the administrative offices of the Master’s program, creating roles for Coordinator, Accountant, Administration, and Human Resources, to simulate a real university environment.
Moving on to the next step, I designed the infrastructure for the Master’s program classes. This involved creating structures such as libraries and campus facilities. Additionally, I added activities that the Master’s program offers, including a sample activity that I created for the students like Language Exchange.
After creating Researchers and Professors, the final group I will mention is the students.

1.2 University Schema

In RDF (Resource Description Framework), a schema is a set of rules or vocabulary that defines the structure of the data model. It is used to describe the classes, properties, and relationships that are used to represent information in RDF.

The RDF schema (also known as RDFS) defines a set of constructs for describing the structure of RDF data.

Within the schema, I have defined all the classes and properties utilized in the data, including
Director, *Person*, *Student*, *Professor*, *Man*, *Woman*, *Researcher*, *Coordinator*, *Accountant*, *Administration*, *humanResources*, *Library*, *Campus*, *TeamResearchGroup*, and *languageExchange*.

And for the properties: *paperPublication*, *nationality*, *decideCourseTeachers*, *acceptedStudents*, *yearsOld*, *carrierYears*, *courseTeaches*, *MasterDirector*, *name*, *hasDirector*, *teacherCoordinator*, *coordinatorMaster3IA*, *masterCountryGratuate*, *hasColleague*, *hasStudent*, *teaches*, *Lab*, *hasFriend*, *hasTeacher*, etc.
To ensure clear and organized ontology development, it's important to include labels, comments, domains, ranges, subClassOf, and subPropertyOf for every class and property. In my ontology, many classes are subclasses of Person because roles like director, professor, researcher, and man all fall under the umbrella of "person." However, it's crucial to note that not all classes should be subtypes of Person - for instance, Campus should not be classified as a subtype of Person since it's a physical location and not a human being.

Within the properties, I will take an example of property:

```
:nationality a rdf:Property ;
    rdfs:label "nationality"@en, "nationalité"@fr ;
    rdfs:comment "The nationality of a person."@en,
                 "La nationalité d'une personne."@fr ;
    rdfs:domain :Person ;
    rdfs:range  rdf:Literals
```

the properties "rdfs:domain" and "rdfs:range" are used to specify the domain and range of a property in an ontology.

In the example I provided, "rdfs:domain :Person" indicates that the property to which this applies can only be used with instances of the class "Person". This constraint implies that the property is only meaningful when used to describe individuals who belong to the "Person" class. In creating an ontology, it's important to be mindful of which classes can possess certain properties. For example, when defining the property "nationality," we must ensure that it is only used to describe instances of the class "Person." It wouldn't make sense to assign "nationality" to a campus or a library since those are physical locations and not individuals. By limiting the use of "nationality" to instances of the "Person" class, we can maintain the integrity of our ontology and avoid creating inconsistencies or confusion.
Similarly, "rdfs:range rdf:Literals" indicates that the property can only have values that are RDF literals, such as strings.

I will take another example,

```
:decideCourseTeachers a rdf:Property ;
    rdfs:label "Decision course teachers"@en, "Enseignants du cours de décision"@fr
    rdfs:comment "To decise which teacher will teach a particular class."@en,
```

```
                "Pour décider quel enseignant enseignera une classe particulière."@
    rdfs:domain :Director ;
    rdfs:range :Professor .
```

(With this property I wanted to show that the director decide which teacher will give lesson for one particular class.)

The property can only be used to describe instances of the class "Director" and that its values can only be instances of the class "Professor". In other words, the property is restricted to a specific subset of individuals within the ontology.

2. Queries

In this section, we present a number of queries to study our RDF and its schema. Queries in this section should be performed with the following prefices defined:

```
@prefix : http://ns.uca.fr/humans/schema# .
@prefix d: http://ns.uca.fr/humans/data# .
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# .
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema# .
@prefix xml: http://www.w3.org/XML/1998/namespace .
@prefix xsd: http://www.w3.org/2001/XMLSchema# .
```

I created the file .txt with all the queries, but here I will show some of the most important.

2.1 SELECT query to find all the properties of Aglind.

```
select * where {<http://ns.uca.fr/humans/data#Aglind> ?p ?t.}
```

2.2 Gives to me the description of Aglind, also with the graph

```
describe <http://ns.uca.fr/humans/data#Aglind>
```

It is shown in the figure 1.

2.3 This query indicates for each person if her age is even (true or false)

```
select * where {?person a :Person;
:yearsOld ?age
BIND(IF((STRENDS(STR(xsd:decimal(?age)/2),".5")), "odd", "even") AS ?even)
}
```

2.4 "contruct and insert" Construct returns a new RDF graph

```
construct { ?student :hasTeacher ?professor }
where { ?student :hasFriend ?student2 . ?student2  :hasTeacher ?professor }
```

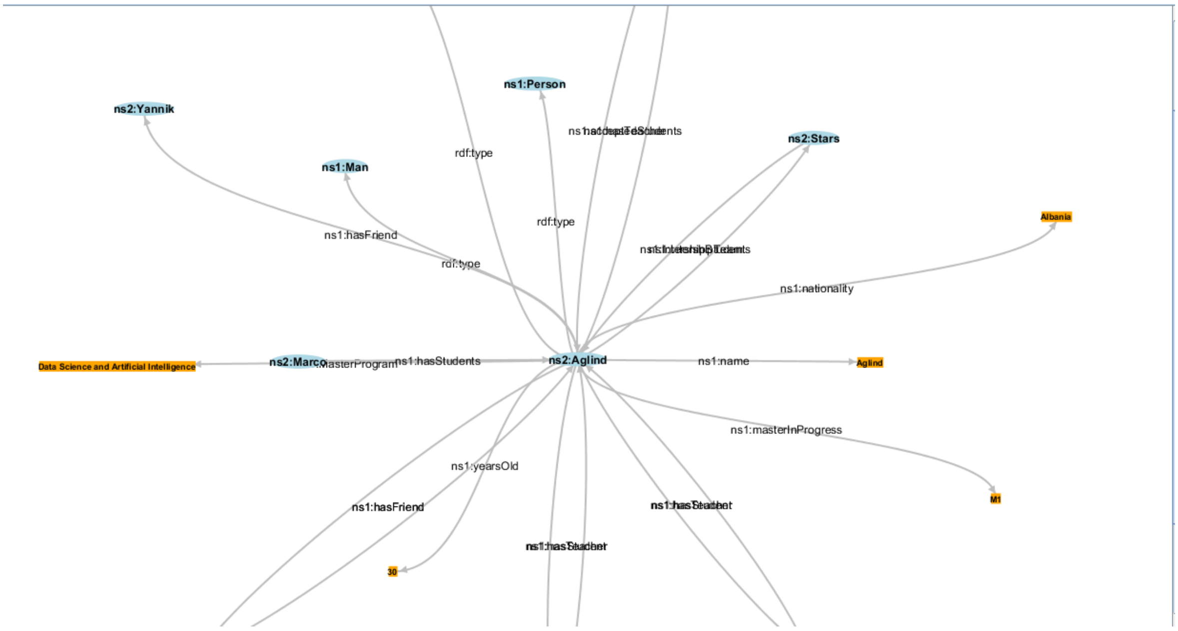

Figure 1: Aglind description

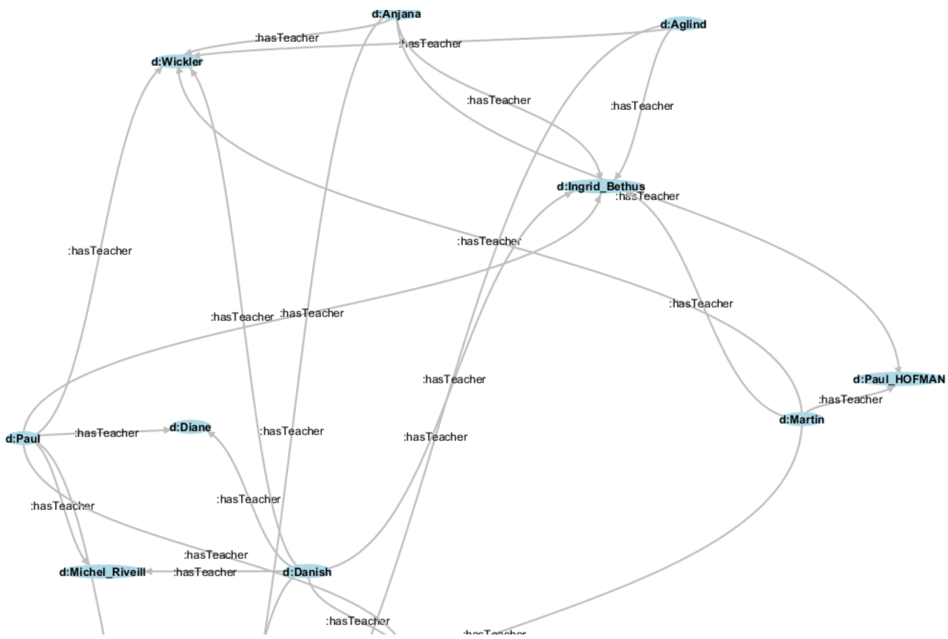

Figure 2: Construct RDF graph

It is shown in the figure 2.

```
insert { ?student :hasTeacher ?professor }
where { ?student :hasFriend ?student2 . ?student2
:hasTeacher ?professor }
```

2.5 Creates a relationship between student and professor

```
construct {
?student rdfs:seeAlso ?professor.
}where{
   ?student   :hasTeacher ?professor.
   ?professor :hasStudent ?student .
filter(?student != ?professor)
}
```

2.6 classes of the ontology and subClassOf in the ontology

```
select * where{?class rdf:type  rdfs:Class}
select * where{?subClass rdfs:subClassOf  ?class}
```

3. Conclusion

To conclude my work, I have made a concerted effort to provide a comprehensive description of the UCA by incorporating new classes and properties while establishing relationships between them.

Furthermore, in the query section, I have endeavored to include the most significant queries covered in our lessons and clearly explain the data I am seeking.
I am thrilled to have taken this class, as I have learned new and fascinating concepts about web of data. The course has broadened my knowledge and perspective on the topic, leaving me feeling enriched and gratified.