UNIVERSITEIT VAN AMSTERDAM

TIMES400ALUMNI

Semantic Web application

By group 50; Chris Tolmeijer (10219749) and Filipp Peresadilo (6183719)

Introduction

After high school many students will have to make a, sometimes rather hard, decision; to which university shall I go? There are a lot of factors someone may take into account, for example; how does the university score? Or where is the university located? But what you may not hear so much is that students take the alumni into account. Almost everyone has someone they look up to; it could be someone from their own area of expertise, or just someone in their area of interest. If a student is able to see if their idol went to a certain university, or if they are able to see what noticeable people went to the university they currently have in mind, it might make the choice feel more comfortable or even influence their decision.

There is however, currently no easy way to browse to universities, find the location with points of interests and find the noticeable people that went to that university, or search for a person and find the corresponding university and information all at the same time. Therefore for this assignment, our goal is to create a semantic web application that does provide these functionalities; a semantic web application that is built for users to discover where all great scientists, politicians, and artists have studied at one of the top universities in the Times Higher Education ranking. Additionally, we have added companies and organizations to see where these alumni have worked after their graduation. We kindly invite you to have a look at our application via the following link:

`http://crizit.nl/school/sw/site/`

In short, to bring this project to a realisation three data sources are used: 1) Times Higher database, contains the top 200 of universities and some basic scores. 2) DBPedia, an ontology containing a lot of information from Wikipedia entries. 3) LinkedGeoData, an ontology containing lots of points of interests with their geographical location. The Times Higher data source is originally in MySQL format; R2DQ is used to generate a mapping to the ontology that will be created for this project and to create an RDF dump of the MySQL data. The data is then imported into Joseki, and SPARQL is subsequently used to query through Joseki to the external semantic endpoints to retrieve the data we require for this project. ARC2 is installed on a web server to create our own online end point, which our semantic web application uses.

In this report we will describe the steps we have taken to realize this project. First, we have built an ontology that supports categorisation of universities, people that went to the universities, their area of expertise and the companies they have been working for or are working for now. This is described in the following chapter.

Ontology

In the ontology development process we have used the 7-step framework we have discussed during the lectures. The steps are described below. A visualization of the complete ontology can be seen in **figure 1**.

Step 1. Scope

To determine the scope of our ontology we have determined what is relevant to show in the semantic web application. As can be read in the introduction, our goal with our application is to provide students with an overview on which noticeable people went to their university, the companies those people have been working for or are working for now, the location of the university, and university ranking information. Additionally, the university's location will be shown on a map along with some points of interests. Only noticeable people that went to the universities listed in the Times Higher database will be used within this project. This means that our ontology contains information about alumni of the top 200 universities from the Times Higher Education list, the companies they work for, the locations of the universities, and of course the top universities themselves with their accompanying information.

Considering the quality of our ontology we have used the following competency questions, and successfully tested queries to answer these questions:

- What people have studied at X top university?
- At what top university has studied X alumni?
- In what city has X alumni studied?
- At what universities have alumni, employed by X company, studied?
- Where have alumni, who have studied at X city, worked?

Step 2. Reusing existing ontologies

Luckily, we did not have to build our ontology from scratch. A lot of data in this project is retrieved from external ontologies. For that reason we are able to reuse a lot of classes and properties available from the different ontologies. We have reused the following external ontologies.

- **DBpedia**. We have used DBpedia for this ontology contains a lot of information about people, and information about them like their birthdate, birthplace, names, etc. Furthermore, it contains information about the companies they work(ed) for. For the ontology we have reused properties like `birthplace`, `ethnicity`, `occupation`, `residence` and `religion`, and classes like `Company`, `Person`, `Industry`, `Religion`, and `EthnicGroup`.
- **LinkedGeoData**. We have used LinkedGeoData because this ontology contains a lot of information about geographical entities and additional information about them. For example, its class `GeographicalObject` contains subclasses like `Café`, `Library` and `Restaurant` about additional places within a particular city.

The class that is not going to be reused is the Universities class. The universities class from our ontology contains different properties and we think it is best practice to keep it separated for that reason.

Step 3. Enumerating terms

Prior to the declaration of classes, we have enumerated important concepts in our ontology, which formed the basis for our classes and properties:

University	Person	Location
Point of Interest	Name	Longitude
Latitude	Rank	Score
Citation level	Industry income	Research level
International mix level	Teaching level	Birthplace
Company	ID	Order
Net worth	Industry	Employees
City	Country	

Has religion	Founded by	works at
Is located in	studied at	is named
Has ethnicity	Founded in	active in

Lives in Born in

Step 4. Define the classes and the class hierarchy

In our ontology, we have constructed the following classes:

- **`Company`**, indicating the company where the alumni have worked or are still working.
- **`EthnicGroup`**, indicating the ethnicity of a particular alumnus.
- **`GeographicEntity`**, indicating one of the following subclasses:
  - **`City`**
  - **`Country`**
- **`GeographicalObject`**, indicating a certain place in a city, one of the following subclasses:
  - **`Café`**
  - **`FastFood`**
  - **`Library`**
  - **`Restaurant`**
  - **`Shop`**
  - **`SportCentre`**
  - **`Supermarket`**
- **`Industry`**, indicating where a particular company is specialized in.
- **`Person`**, indicating the alumni.
- **`University`**, indicating one of the top 200 universities, divided in the following two subclasses:
  - **`PublicUniversity`**, indicating public or state universities.
  - **`PrivateUniversity`**, indicating private universities.
- Religion, indicating the religious views of an alumnus.

Step 5 and 6. Defining properties and their properties

In our ontology we have constructed the following object properties:

- **`activeIn`**, indicating the industry where a company is active in. The domain of this property is **`Company`**, and the range is **`Industry`**.
- **`almaMater`**, indicating the university where an alumnus has studied. The domain of this property is **`Person`**, and the range is **`University`**.

- **birthplace**, indicating where an alumnus was born. The domain of this property is **Person**, and the range is **City**.
- **ethnicity**, indicating the ethnicity of an alumnus. The domain of this property is **Person**, and the range is **EthnicGroup**.
- **foundedBy**, indicating which person has founded a certain company. The domain of this property is **Company**, and the range is **Person**.
- **location**, indicating where a university or company is located. The domain of this property is **Company** and **University**, and the range is **GeographicEntity**.
- **Occupation**, indicating in which company an alumnus is or was active. The domain of this property is **Person**, and the range is **Company**.
- **religion**, indicating the religious view of an alumnus. The domain of this property is **Person**, and the range is **Religion**.
- **residence**, indicating the location where an alumnus lives. The domain of this property is **Person**, and the range is **GeographicEntity**.

The visualization of our complete ontology can be seen in **figure 1**.

Step 7. Defining instances

The instances for our ontology are first derived from the Times Higher Education Top 200 universities. The techniques we have used to map this data onto our ontology are described in the next chapter. For the instances regarding geographic entities (cities and countries), persons, and companies with their additional information, we have used the data from LinkedGeoData and DBpedia respectively.

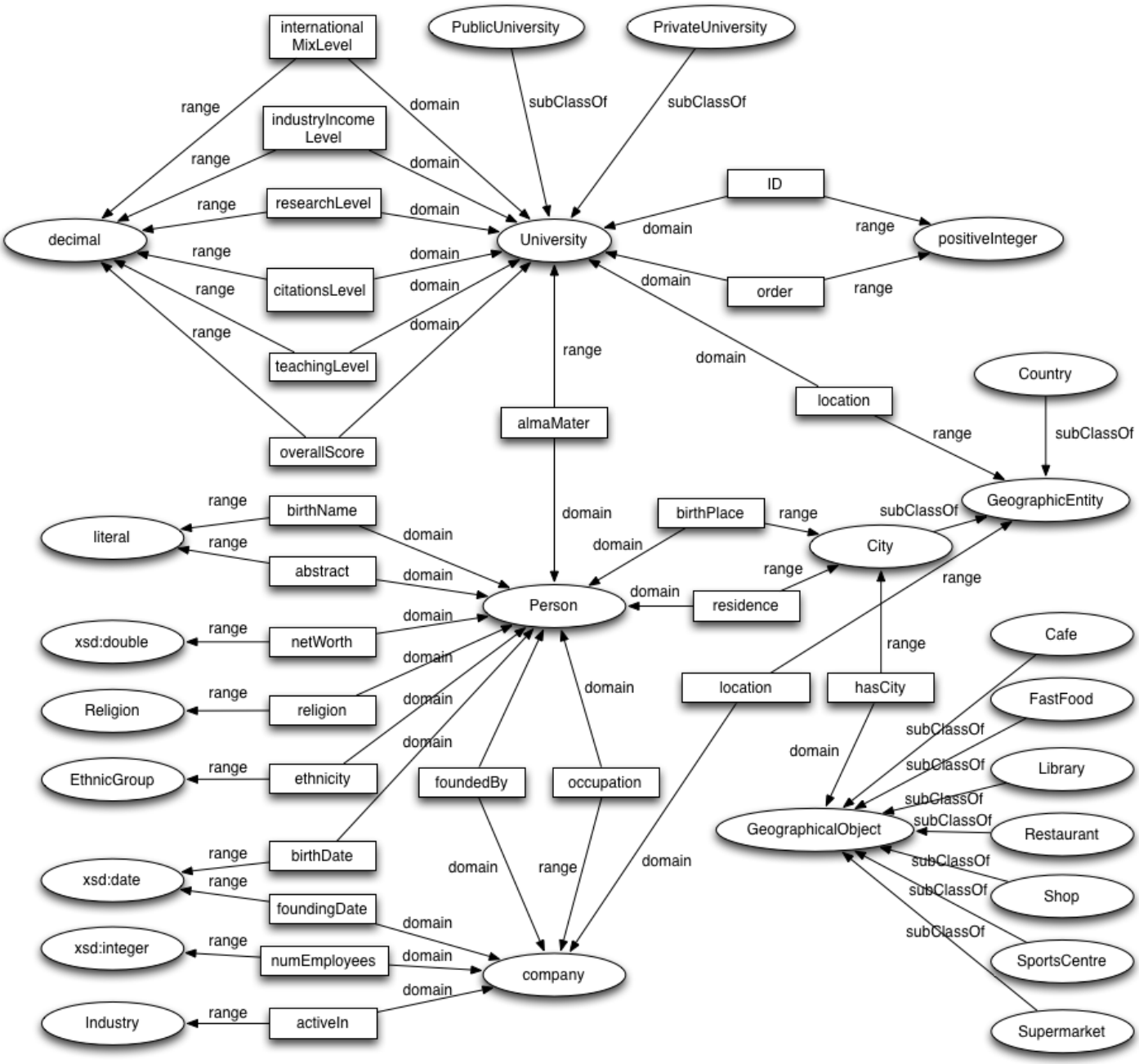

Figure 1 Visualization of the ontology

Information integration

For our semantic web application we need a lot of different information. Some of that information is present in the Times Higher database, other information we need to collect from external resources. Every piece of information we collect from external resources is stored locally into our own ontology. This makes the information accessible for us, even if the external resource is offline, and provides the best user experience for our users. In this chapter we will describe what information we need, what technologies we used and how we used them to collect that information.

Data description

The core of our application is based on the Times Higher database. The Times Higher database contains information about the top 200 universities of the world. The information represents the rank in the list and other ranking information about each university. This is information we would like to show in the web application. To extract that data we generate a mapping with D2RQ and dump the database's data into RDF, this will be explained further in the next sub chapter.

We also would like to show the location of each university, and the people that attended the university. This information is not present in the Times Higher database. This information is retrieved from DBPpedia. DBpedia is an ontology containing a lot of information collected from Wikipedia. We query to DBpedia using SPARQL to retrieve the information we require and put it in our own ontology for offline use.

Next to all the information we require about universities, we also require information about the surroundings of the universities and show this on a map. This information cannot be found in DBpedia and is therefore retrieved from LinkedGeoData. LinkedGeoData is an ontology containing a lot of objects with their geographical location collected from the OpenStreetMap project. We also query to LinkedGeoData using SPARQL to retrieve the information we require and put it in our own ontology for offline use.

For almost every resource we have in our ontology, we would like to show an English readable name; therefore the most minimal information we collect from a resource is the RDFS label. Almost all the other information we collect about a resource is optional, we do that to make sure every instance is present in our ontology. It is not required for our web application to have all the information about each instance to function correctly. For every

geographical instance we require the longitude and latitude of the resource, because the only location such instances will be shown in our application is on a map.

Mapping the Times Higher database

D2RQ is used to generate a mapping from the Times Higher MySQL. It generates a turtle file, which contains information about the table, and the format it will export the data to when the mapping file is used in the *dump-rdf* command. By changing or removing the instances in the mapping file we were able to change the output. The following list shows which fields we exported to which data property from our own ontology.

- *rankings__label* to *rdfs:label "university name"@en*
- *rankings_order* to *UniversitiesTop200:order*
- *rankings_overall_score* to *UniversitiesTop200:overallScore*
- *rankings_teaching* to *UniversitiesTop200:teachingLevel*
- *rankings_international_mix* to *UniversitiesTop200:internationalMixLevel*
- *rankings_industry_income* to *UniversitiesTop200:industryIncomeLevel*
- *rankings_research* to *UniversitiesTop200:researchLevel*
- *rankings_citations* to *UniversitiesTop200:citationsLevel*

The rankings_university field was removed because it represents the university's name which we are already using in the rdfs:label.

The decision to add the language attribute to the rankings__label field was made so we were not required to specify a language filter in our SPARQL query that is being used to match the universities from DBpedia with the universities from the Times Higher database.

When our mapping process was completed we used the *dump-rdf* command combined with the mapping file to dump the MySQL data to RDF instances. With this RDF file we are able to query to all the external ontologies to collect information.

Collecting external information

To collect all the external information, a Joseki server is set up. Our ontology along with the Times Higher RDF dump generated by D2RQ is imported into Joseki. Joseki enables us to execute SPARQL queries onto our ontology or onto an external ontology by using the SPARQL Service keyword.

To make sure we wouldn't run into any performance issues, we split up the queries into different parts. This allowed us the collect just a small set information at a time and made us able to respond relatively fast on any mistakes being made and correct them. The first query is to match the universities from the Times Higher database, which has been dumped to RDF, with the private- and public universities from the DBpedia ontology. After we have collected the information about the universities location on DBpedia we were able to directly use those resources URI's to retrieve more information from DBpedia like cities, countries, persons and companies.

To collect all the geographical objects from the LinkedGeoData ontology, we first collected all the cities from DBpedia. To collect the cities we used three different queries, one to collect each city where the persons were born in, one to collect each city where the universities are located at and one to collect each city where the companies are located at. We manually combined the output from the three different queries into a single graph that contains all the cities possible in our data. This single graph is used in Joseki to collect cafes, fast food restaurants, libraries, restaurants, shops, sport centres, super markets and universities for each city.

All the different queries to collect information from external resources use the construct keyword to generate a graph in the form of an OWL file that matches our ontology. All the OWL files combined represent all the instances in our ontology. The semantic web application built in this project uses AJAX to send SPARQL queries to those instances using an ARC2 endpoint. All the queries used in this project can be found in the appendix.

Application interface

During the creation of the user interface design of our application we had three main goals: first (1) our application should give a clear representation of the relevant data for the user, second, (2) a user should be able to easily discover new universities, alumni, and companies, and third, (3) our application should incorporate geographic as well as other information in one screen. The result can be accessed via the following link:

http://crizit.nl/school/sw/site/

An overview of the user interface can also be seen in **figure 2**. The application contains three main building blocks:

1. **The search bar**. Here the user can search for a person, university or company. The data comes from the database of the top 200 universities, alumni and companies. This

search bar is integrated with a automatic drop-down menu, which shows the relevant concepts during a search.

2. **The information bar**. Here all the relevant information about a person, university or company is shown. Besides the static information, this section contains links to alumni, links to companies, and links to universities. In this way, the user can easily click through to a page of an alumnus if the user is on a university or company page.
3. **The map** shows the location of the selected university or company, with additional landmarks (geographic objects) like shops and libraries. We have used the Google Maps API for the map.

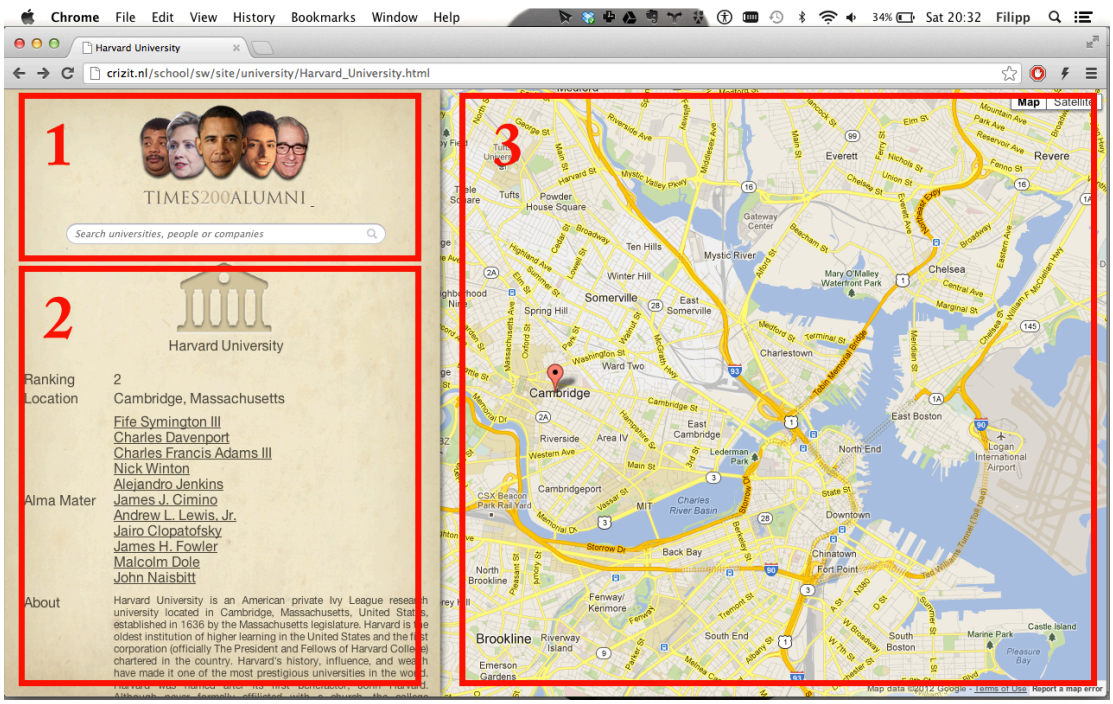

Figure 2 User interface of the application

The data is viewed with the use of the AJAX getJSON function from jQuery framework to query the ARC2 endpoint installed on the same server to acquire the data. A SPARQL query is present in the JavaScript code and occasionally filled in with user input, the SPARQL query is being send in the 'query' GET variable to the ARC2 endpoint. The ARC2 endpoint processes the query and returns the results in JSON format, which is being parsed by the getJSON function into a JavaScript object. The information in this object is written to the HTML of the view.

External applications can access our data using the same endpoint as our own application does. The endpoint is set up with ARC2, which is a PHP based SPARQL server.

To make sure our server doesn't get flooded we limited the maximal number of results to 250 and disabled certain query keywords like INSERT, UPDATE and SERVICE. Because our website is mostly HTML5 and the for the user visible information is generated on run time. Because it is not possible for an application to read this we did not put any RDFa into the HTML tags. The location of our endpoint is: http://crizit.nl/school/sw/endpoint.php

Conclusion

In this report our goal was to create a semantic web application, which let's the user discover where all famous scientists, artists, politicians and business people have studied. Additionally, the application should show who the alumni of the top universities are, and where they have worked or are still working. We began by describing the creation of the ontology according to the 7-step approach. Subsequently, we elaborated how we integrated the data, mapped the Times database, and collected the external data. Finally, we showed the ideas and practicalities behind the user interface design, and the way the data is represented. With this application, we are convinced that we provide current students and upcoming students a new and interesting way to discover universities.

In this course, and during the creation of this application in particular, we have seen the great potential of the Semantic Web. Although we could not integrate everything we would like (for example, the *TechCrunch* linked data was temporarily offline during the course), we hope we have made a significant addition to the Semantic Web with the creation of this application.