Knowledge & Data Engineering Group Project

Technical Report - Group D

Haoxian Liu, Juan Carlos Montenegro, Kaize Shen, Mengqi Wang

Code: Click here to view project.

I. Introduction

In this project we aimed to combine different datasets to model our ontology model and produce a set of queries that make use of our datasets when outputting their results.

The datasets used in this assignment are Regional ethnic diversity in England and Wales which consist on: population by ethnicity region and urban rural location, ethnic population of local authorities in England and Wales, population by ethnicity and region, population of England and Wales, Unemployment by region, Average hourly pay and Household income.

II. Approach to Ontology Modelling

A. Description of Competency Questions

Our questions focus on the employment of different ethnic groups in different parts of England. We have designed five problems for retrieving information using relation between classes and five problems for searching information directly from two different classes.

The questions included a wide range of topics, exploring the distribution of ethnicity by region, the distribution of income by ethnicity, the distribution of income by region, and so on. We also proposed two imaginative questions, which had little practical significance but gave us a more tangible understanding of regional unemployment employment.

B. Description of datasets selected for application

Datasets:

1. **Household Income**: This dataset describes how much each household earns for each ethnic group, it also defines the date, the type of ethnic group, the geography, value and denomination.
2. **Average Hourly Pay**: This dataset describes how much you earn on average annually by ethnicity, value, date, and type of currency
3. **Ethnic Population of Local Authorities in England and Wales**: This dataset measures population by local authority and ethnicity, value, value type, population source, date, and geography.
4. **Population by Ethnicity and Region**: This dataset measures the population by ethnicity and region, as well as all usual residents, and the population, value, percentage of value, population source, geography, and census date.
5. **Population by Ethnicity Region and Urban Rural Location**: This dataset measures the population by ethnicity and region,urban or rural, value, percentage of value, population source, geography, and census date.
6. **Population of England and Wales**: This dataset measures the population by ethnicity and region, value, value_note, population source, geography, and census date.
7. **Unemployment by Local Authority**: This dataset measures the unemployment rate for ethnicities sorted by annual date, local authority, age, confidence interval, numerator, denominator
8. **Unemployment by Region**: This dataset measures the unemployment rate for ethnicities sorted by annual date, local authority, age, sex, confidence interval, numerator, denominator

C. Assumptions Made

- The definitions of ethnic information in different datasets are uniform and accurate
- The unemployment rate is basically the same over time
- Primary key IDs of different classes are unique and can be used as unique identifiers

D. References to sources used/reused for people

- Jena documentation https://jena.apache.org
- R2RML documentation https://www.w3.org/TR/r2rml/
- The evolution of Protégé: an environment for knowledge based systems development, International Journal of Human Computer Studies, https://doi.org/10.1016/S1071-5819(02)00127-1.

E. Discussion of data mapping process

- **Data Pre-processing**
  In this step, each data in the dataset is given its own ID, for example, the ID of each item of data is the number_household in the dataset “Household Income 2021.csv”; There is some missing data in the raw data, so the items with missing data are removed from the dataset; Some data types in the raw datasets are defined as literal, but these types cannot be used in subsequent queries, so pre-processing is done to change the data under the same label in different datasets to the correct data type. For example, “population” and “time” are changed to integer type, “value” is changed to double type, etc.
- **Mapping File (final.ttl)**
  The data is imported by classes, the name of the imported property and data type should be defined when importing, The following code shows an example of how to import the value column of class ”Population”.

```
rr:predicateObjectMap [
rr:predicate groupD:hasValue;
rr:objectMap [
rr:column "VALUE";
rr:datatype xsd:double;
];
];
```

From the code we can know that this property is defined as “hasValue”, and the data type is “double”.
The next step is to define the relationship, which is necessary between every two classes that are related. The following code shows how we define the relation between different classes. This code defined a relation between class ”Population” and class ”Household”. We can know that all population have a household income. Thus we defined the relation between them is ”Have_a”. Also from the code we can know that these two classes have a same column, which is ”Ethnicity”.

```
rr:predicateObjectMap [
rr:predicate groupD:Have_a ;
rr:objectMap [
rr:parentTriplesMap <#Household> ;
rr:joinCondition [ rr:child "ETHNICITY" ; rr:
    parent "ETHNICITY" ]
]
];
```

- **Import Data**
  Based on the rules described above, we created ’final.ttl’ as the final mapping file. Next we just need to generate another property file responsible for letting the computer know where to import the data from. Once we have these two files, we can uplifting them with the provided jar package. Then the generated file ’final_new.ttl’ file is our output and can be used to do the query.

F. Explanation of Use of Inverse, Symmetric and Transitive Properties

- **Transitive Proprities:**
  We used transitive proprities in our ontology design. We can know that all population contains the unemployment population. All these people have to live in one region and each region contains one address. Thus we can see there is a transitive property in class ”Population”, class ”Region” and class ”Address”.

III. Overview of Design

A. Description of Ontology Design

Our main idea was to do a study of the different ethnicities in the regions of England and Wales, which we started looking for the best datasets that would help us model these use cases on the platform: https://data.gov.ie/.

Once we chose the topic to develop and select the datasets to use for our project, we take the initiative to create the ontology model which we base on the design of a description of all the objects and their relationships, we take the main actors, actions of user, tasks, properties, instances, etc.

We identify 6 main classes: Person, Address, Region, Population, HourlyAvgPay, Ehnicity and HouseHold. AsianEthnicity, BlackEthnicity and WhiteEthnicity were created as subclasses of Ethnicity since they were the main ethnicities we wanted to investigate. In region we have two subclasses: Rural and Urban, since we wanted to investigate the percentage and range of people within these ethnicities who lived in these areas.

Once this is done we relate the classes between giving them a logical order. We use the main columns of the datasets as properties within our ontology model to subsequently make queries consistent with our preliminary questions. The following fig.1, fig.2 and fig.3 show the ontology design and the class diagram of the ontology and the VOWL diagram of the ontology design.

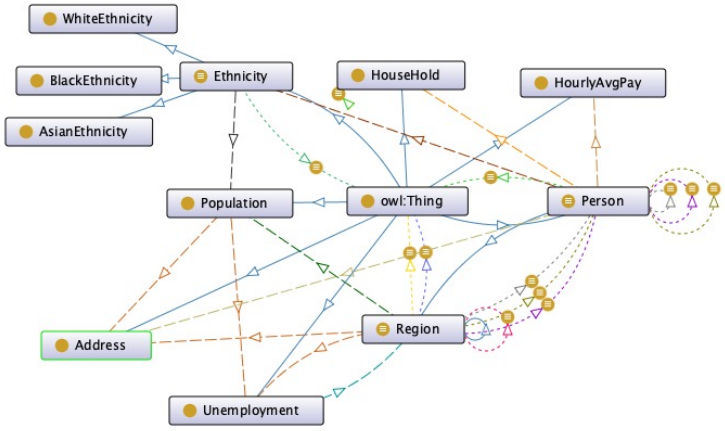

Fig. 1. Class Diagram

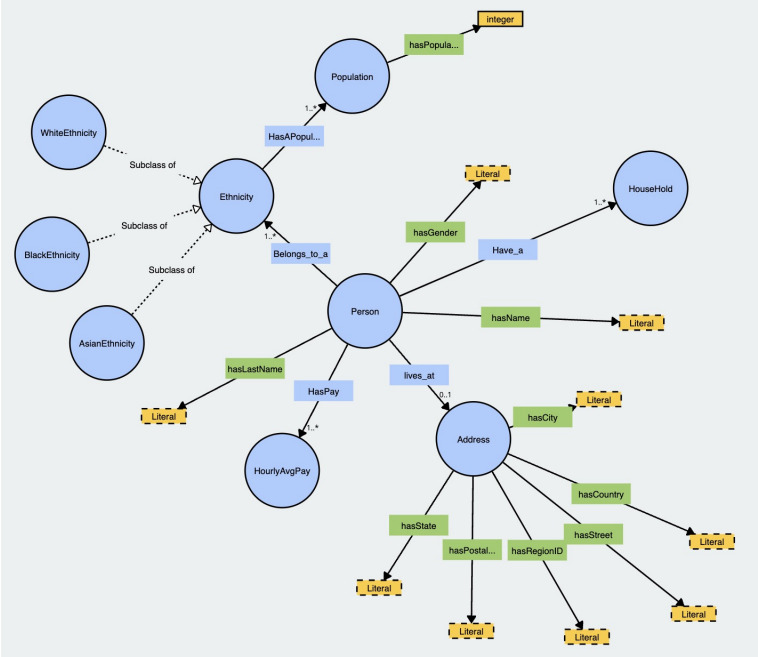

Fig. 2. VOWL Diagram

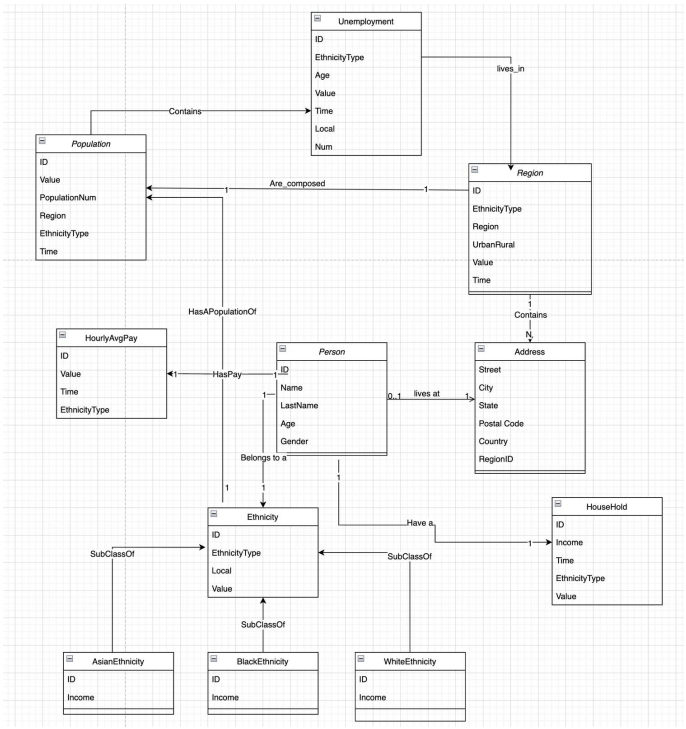

Fig. 3. Ontology Design

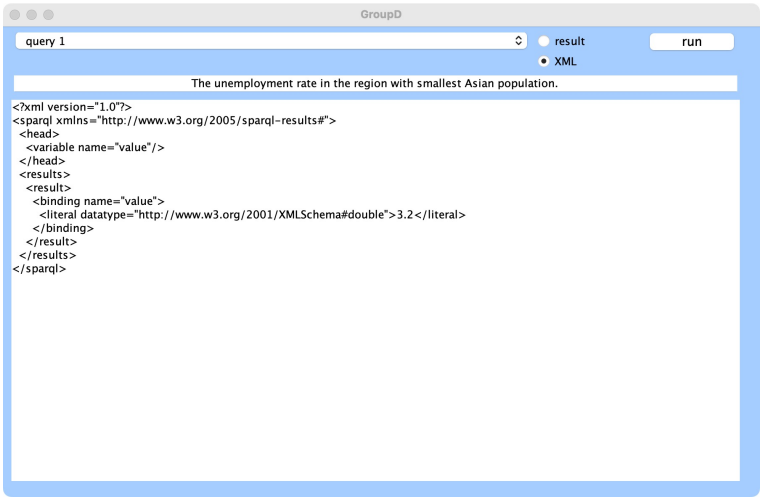

Fig. 4. output XML result

B. Description of Application Query Interface

Once the ontology and corresponding owl file was created, the next task was to query for results. The queries are again executed.

We have designed a Graphical User Interface(GUI) for selecting and running the query. The design and implementation of GUI is achieved using Java technologies-JFrame. We achieved executing queries by using Jena- a free and open source Java framework for building Semantic Web and Linked Data applications.

We showed our output with two different formats-XML and Text in fig.4 and fig.5. We can choose the format that we want and output it.

```
ResultSet results = qexec.execSelect();
ResultSetFormatter.asXMLString(results);
```

```
ResultSetFormatter.toList(results);
```

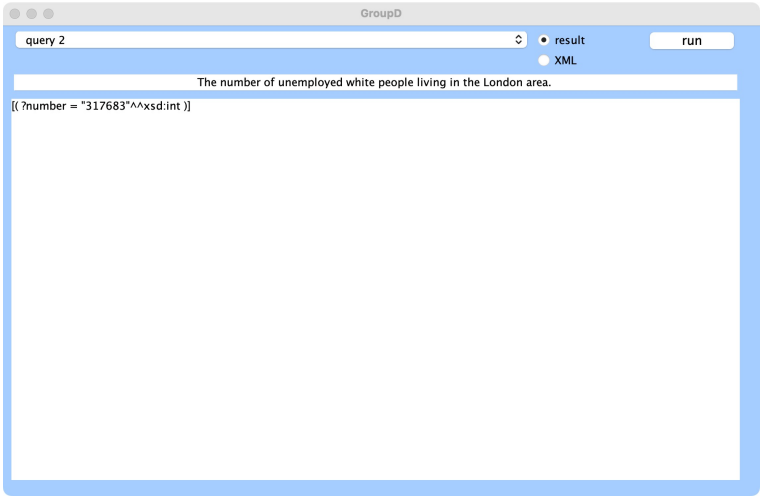

Fig. 5. output Text result

C. Description of Queries

- **Common Prefix:**

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX groupD:  <http://www.semanticweb.org/liaosiyu/ontologies/2022/10/untitled-ontology-6/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

1. **The unemployment rate in the region with smallest Asian population.**
   This question is mainly used to explore how the unemployment rate is when there are fewer Asians. This problem requires the following two datasets, "Population by Ethnicity Region and Urban Rural Location" and "Unemployment by Region". This question need the following classes, "Unemployment" and "Region". These two classes have one common property, "Region". Thus we need to first get the region information which has the smallest Asian population and then use this information to search the unemployment rate.

```
SELECT ?value
WHERE {
?a a groupD:UnemploymentRegion.
?a groupD:hasEthnicityType ?ethnicity;
   groupD:hasTime ?time;
   groupD:hasValue ?value;
   groupD:hasRegion ?region.
   {
        Select ?region{
            ?x groupD:hasRegion ?region;
        groupD:hasEthnicityType ?ethnicity;
        groupD:hasPopulationNum ?population;
        groupD:hasValue ?value.
        FILTER(?ethnicity="Asian")
    }
    ORDER BY ?population
    LIMIT 1
  }
  FILTER(?ethnicity="Asian"&&?time=2019)
}

```

2. **The number of unemployed white people living in the London area.**
   The question aims to find out how many unemployed people live in specific region and ethnicity. This problem requires the following two datasets,"Population by Ethnicity and Region" and "Unemployment by Region". In the query process, ethnic information and regional information are the common property of the two classes, in which there is an "Contain" relationship. The whole population must contains the unemployed population. Thus I use these relations to query the result.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:UnemploymentRegion.
    ?x a groupD:Population.
    ?x groupD:Contains1 ?region.
    ?x groupD:Contains2 ?ethnicity.
    ?x groupD:hasPopulationNum ?pop_num.
    ?a groupD:hasValue ?value.
    ?a groupD:hasRegion ?region1.
    ?a groupD:hasTime ?time.
    ?a groupD:hasEthnicityType ?ethnicity2.
        FILTER(?region1 = "London" && ?ethnicity2 = "White" && ?time = 2010)
        FILTER(?region = ?a && ?ethnicity = ?a)
    bind(xsd:int(?pop_num*?value*0.01) As ?number)
}
```

3. **Based on the unemployment data of Newham in 2004, predict the number of unemployed Chinese in Newham in 2011.**
   This question is mainly intended to obtain the exact number of unemployed people in different regions in 2011 by ethnicity. This problem requires the following two datasets,"Unemployment by Local Authority" and "Ethnic Population of Local Authorities in England and Wales". There is one common property between class "Unemployment" and "Ethnicity", which is "Local" information. Between these two classes, we can find that all ethnicity have a certain population of unemployed people. Thus we used this relation to query the result.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:Unemployment.
    ?x a groupD:Ethnicity.
```

```
    ?x groupD:HasAPopulationOf ?local1.
    ?x groupD:hasValue ?value.
    ?x groupD:hasEthnicityType ?ethnicity1.
    ?a groupD:hasEthnicityType ?ethnicity2.
    ?a groupD:hasLocal ?local2.
    ?a groupD:hasTime ?time.
    ?a groupD:hasNum ?num
        FILTER(?local2 = "Newham" && ?ethnicity2 = "All" && ?time = 2004 && ?ethnicity1 = "Asian/Asian British: Chinese")
        FILTER(?local1 = ?a)
    bind(xsd:int(?num*?value*0.01) As ?number)
}
```

4. **How many black people in North East of England live in rural areas in 2011?**
   This question was designed to explore whether people of different ethnicities and regions prefer to live in rural or urban. This problem requires the following two datasets, ”Population by Ethnicity and Region” and ”Population by Ethnicity Region and Urban Rural Location”. In this query, we used class ”Population” and class ”Region”. These two classes have two commom property, ”Region” and ”EthnicityType”. We can also know that population are composed by people in different region. Thus they have a relation ”Are_composed” and we used this relation to do the query.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:Population.
    ?x a groupD:Region.
    ?x groupD:Are_composed ?region1.
    ?a groupD:hasEthnicityType ?ethnicity1.
    ?a groupD:hasPopulationNum ?num.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasRegion ?region2.
    ?x groupD:hasValue ?value.
    ?x groupD:hasUrbanRural ?loc.
        FILTER(?region2 = "North East" && ?ethnicity2 = "Black" && ?loc = "Rural" && ?ethnicity1 = "Black")
        FILTER(?region1 = ?a)
    bind(xsd:int(?num*?value*0.01) As ?number)
}
```

5. **How many black people in London have a average weekly household income over GBP 2000?**
   This question is intended to explore the number of rich people of different ethnicities in the London area. This problem requires the following two datasets, ”Population by Ethnicity and Region” and ”Household Income 2021”. In this query, we need to use class ”Population” and class ”Household”. These two classes have one common property, which is ”EthnicityType”. Also we know that all the people have a household income. Thus they have a ”Have_a” relation. We used this relation to do the query.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:Population.
    ?x a groupD:Household.
    ?a groupD:Have_a ?ethnicity1.
    ?a groupD:hasRegion ?region.
    ?a groupD:hasPopulationNum ?num.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasIncome ?income.
    ?x groupD:hasValue ?value.
    ?x groupD:hasTime ?time.
        FILTER(?income = "GBP 2,000 or more" && ?ethnicity2 = "Black" && ?time = 2011 && ?region = "London")
        FILTER(?ethnicity1 = ?x)
    bind(xsd:int(?num*?value*0.01) As ?number)
}
```

6. **Based on the unemployment rate in 2004, predict the number of unemployed asian people who live in rural areas of London in 2011.**
   We created this question primarily to explore the urban-rural distribution of the unemployed population. This problem requires the following two datasets, ”Unemployment by Region” and ”Population by Ethnicity Region and Urban Rural Location”. In order to get the query result, we need to query class ”Region” and class ”Unemployment”. We can know that all unemployed person should live in one region. Thus we define ”live_in” as the relation between these two classes.These two classes have the same property ”Ethnicity”. We used this relation to query the result.

```
SELECT distinct ?num
WHERE {
    ?a a groupD:Region.
```

```
    ?x a groupD:UnemploymentRegion.
    ?a groupD:lives_in ?ethnicity1.
    ?a groupD:hasUrbanRural ?loc.
    ?a groupD:hasValue ?value.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasNumerator ?num.
    ?x groupD:hasRegion ?region.
    ?x groupD:hasTime ?time.
        FILTER(?region = "London" && ?ethnicity2 = "Asian" && ?time = 2004 && ?loc = "Rural")
        FILTER(?ethnicity1 = ?x)
    bind(xsd:int(?num*?value*0.01) As ?number)
}
```

7. **How much does the proportion of white Irish people in Manchester differ from the proportion of white Irish people in England and Wales as a whole?**
   This question focuses on the difference in the proportion of designated ethnicity in the city and in the country as a whole. This problem requires the following two datasets, ”Ethnic Population of Local Authorities in England and Wales” and ”Population England and Wales”. This query need to get information from two classes, ”Ethnicity” and ”PopWhole”. These two classes have the common property, which is ”EthnicityType”. We used this to do the query.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:Ethnicity.
    ?x a groupD:PopWhole.
    ?a groupD:hasEthnicityType ?ethnicity1.
    ?a groupD:hasLocal ?region.
    ?a groupD:hasValue ?value1.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasValueType ?valueType.
    ?x groupD:hasValue ?value2.
    ?x groupD:hasTime ?time.
        FILTER(?region = "Manchester" && ?ethnicity1 = "White: Irish" && ?time = 2011 && ?ethnicity2 =
    "White - Irish" && ?valueType = "% of pop")
    bind(xsd:double(?value1 - ?value2) As ?number)
}
```

8. **If we took all the white British people in England and Wales and put them together work an hour, how much would the boss have to pay?**
   This question is an imaginative one without much practical significance, stemming mainly from our curiosity. We wanted to find out how much it would cost to pay everyone for an hour’s work. This problem requires the following two datasets, ”Population of England and Wales” and ”Average Hourly Pay”. We need to get information from class ”HourlyAvgPay” and ”PopWhole”. These two classes have a same property named ”EthnicityType”. We used this to do the query.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:HourlyAvgPay.
    ?x a groupD:PopWhole.
    ?a groupD:hasEthnicityType ?ethnicity1.
    ?a groupD:hasTime ?time.
    ?a groupD:hasValue ?value1.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasValueType ?valueType.
    ?x groupD:hasValue ?value2.
        FILTER(?ethnicity1 = "White British" && ?time = 2013 && ?ethnicity2 = "White - British" && ?
    valueType = "count")
    bind(xsd:int(?value1 * ?value2) As ?number)
}
```

9. **How much does the proportion of the unemployment rate of white people in Liverpool which is a city in the North East of England differ from the unemployment rate of white people in the whole North East of England?**
   This question explores the difference between the unemployment rate in cities and the unemployment rate in the local area. This problem requires the following two datasets, ”Unemployment by Region” and ”Unemployment by Local Authority”. In this query, there are two classes need to be used, which are ”Unemployment” and ”UnemploymentRegion”. They have a same property ”EthnicityType”. We used this to do the query.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:Unemployment.
    ?x a groupD:UnemploymentRegion.
    ?a groupD:hasEthnicityType ?ethnicity1.
    ?a groupD:hasTime ?time1.
    ?a groupD:hasLocal ?loc.
```

```
    ?a groupD:hasValue ?value1.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasTime ?time2.
    ?x groupD:hasRegion ?region.
    ?x groupD:hasValue ?value2.
        FILTER(?ethnicity1 = "White" && ?time1 = 2004 && ?ethnicity2 = "White" && ?time2 = 2004 && ?region = "North East" && ?loc = "Liverpool")
    bind(xsd:int(?value1 - ?value2) As ?number)
}
```

10. **If we took all the all the unemployment white people in Adur and put them together working an hour, how much would the boss pay?**
    This question, like the eighth, is also an imaginative one. We are curious about how many jobs an hour and how much we need to pay to solve all the unemployment problems. This problem requires the following two datasets, ”Unemployment by Local Authority” and ”Average Hourly Pay”. We need to search information from class ”HourlyAvgPay” and class ”Unemployment”. These two classes have two common properties which are ”EthnicityType” and ”Time”. Thus we combined these properties together to do the query.

```
SELECT distinct ?number
WHERE {
    ?a a groupD:HourlyAvgPay.
    ?x a groupD:Unemployment.
    ?a groupD:hasEthnicityType ?ethnicity1.
    ?a groupD:hasTime ?time1.
    ?a groupD:hasValue ?value1.
    ?x groupD:hasEthnicityType ?ethnicity2.
    ?x groupD:hasTime ?time2.
    ?x groupD:hasLocal ?region.
    ?x groupD:hasNum ?num.
        FILTER(?ethnicity1 = "White" && ?time1 = 2013 && ?ethnicity2 = "White" && ?time2 = 2013 && ?region = "Adur")
    bind(xsd:int(?value1 * ?num) As ?number)
}
```

IV. Challenges

Throughout the assignment, we were unfamiliar with many of the techniques at the beginning and did not have a deep enough understanding of the project. Thus we encounter many challenges in the following areas.

- **Competency Questions and Queries**
  In the section on designing competency questions, it is easy to design the first few questions, but as the number of problems increases and the number of classes already used increases, it becomes harder to come up with some novel problems next. As I was in charge of both this part and the query part, I encountered some more challenges, which is the questions were not designed with the possibility of being implemented in the query in mind. And because of this, we also changed one question later on, as it could not be implemented.
- **Ontology Design**
  The main challenge I found in the design of our ontology was to analyze the datasets and then model the first classes to then create their instances, there was also some confusion about how to make the connections between classes and attributes, understand the relation of cardinalities, and how use the properties symmetric, inverse, transitive, however after reviewing the readings and consulting the tutors it was possible to delve deeper into this topic. In the use of Widoco it was complicated since there is no official documentation or tutorials that teach how to use the tool, everything was trial and error, but once the tool was installed it was quite easy to implement our documentation of our model.
- **Mapping and Uplifting**
  The biggest challenge we encountered in this section was the difficulty of ensuring uniformity in every part. At the beginning of ontology design, there are always small, unforeseen problems, such as incorrect relationships between classes, or missing properties of the class needed for a search, etc. These problems are difficult to detect before the query is performed, and once there is a problem it requires the members responsible for each of these sections to make changes, which is time-consuming and inefficient. In order to solve this problem, we first analysed all the queries and standardised the design of the ontology, and there were no major problems later.
- **UI Design**
  The biggest challenge I encountered was how to connect data to UI by using Java. I learnt Jena - a free and open source Java framework for building semantic web and Linked Data applications. The basic function was not hard, I realised the query function and output them in two different formats. But how to improve the aesthetics of my interface is really hard, and I try my best to do that.

V. Group Collaboration

We have weekly group meetings to discuss existing work and assign new tasks to each person. We have also created a shared folder to make it easier for everyone to share information on progress and work. We have also created a shared GitHub repository where people can upload their code at any time, so that others can follow up and suggest changes.

In the process of selecting the dataset and topics, we each took part in choosing a possible topic and eventually selected the most detailed and relevant topic and dataset. We then divided the work into four parts: design the problem, design the ontology, upload the data and design the UI. It was difficult to achieve complete synchronisation between the different parts and we encountered some problems during the work. We encountered some problems during our work, but they were all solved with the help of our team members.

In the end, the various modules worked well and we were able to bring the various parts together to form the final product. Overall it was a great collaboration and we all learnt a lot about different areas during the project.

Conclusions

Throughout the project we learnt a lot and gained a better understanding of knowledge engineering and also have a better understanding about various technical tools such as protege, SPARQL statements and the R2RML language. All of us knew very little about these related technologies before this course. Thus the principles and use of these related technologies were learned in an ad hoc manner during this assignment. Due to this, we also encountered many problems. At the beginning we were not confident enough in our ontology design. We asked for teacher's opinion several times and made many changes to the overall structure before we got the final version.

We think that our final result has our strengths and at the same time there are many small issues that still need to be improved. We think our strengths were firstly our choice of topic, which combined ethnicity, population and employment together, which gave us a better understanding of the distribution of different ethnicity and employment information. We have a total of eight datasets and the datasets information are very comprehensive. Secondly, all of our queries utilised different classes, with no two questions utilising the exact same dataset or the exact same attribute information, and the complexity of the questions was high.

We think our weakness is that although the structure of our proposed questions is not similar and utilises all the defined classes, the structure of the query statements implementing the search is similar as the question outputs are all specified with a particular result.