**F21BD Big Data Management**

**Coursework1: Watt Radio Music Ontology**

In this report, the work done for implementing the Heriot Watt Music ontology is presented.

Our report consists of three sub-reports, each represents group-member’s report / contribution

**Design Approach / Merits:**

* Linked Open Data Principles: Used IRI for our ontology to identify things or concepts in our ontology on the web and we also linked other resources through their IRIs.
* Open World Assumption: “In a formal system of logic used for knowledge representation, the open-world assumption is the assumption that the truth value of a statement may be true irrespective of whether or not it is known to be true. It is the opposite of the closed-world assumption, which holds that any statement that is true is also known to be true.”. “Heuristically, the open-world assumption applies when we represent knowledge within a system as we discover it, and where we cannot guarantee that we have discovered or will discover complete information.” (Wikipedia, 2019).
* Agile methodology was used for consolidating our ontologies. That is, requirements and solutions were evolving through iterations of inspection and adaptation. The ontology is built to analyze data and answers questions related to music tracks played on Heriot Watt Radio. Some of the main Ontological and reasoning concepts are explained with examples and snippets.

**Ontology Alignments**

The final part of your report should detail the mapping alignment that you have performed to an existing ontology. You should include a summary of the target ontology and why it was selected. You should then provide the 5 alignments that you would declare, and for each one provide the design choice for the alignment. You should include and populate the following table in your report.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Source Entity** | **Mapping Property (Turtle syntaxt)** | **Target Entity** |
| M1 | watt:Person | owl:equivalentClass | foaf:Person |
| M2 | watt:MusicCompany | owl:equivalentClass | foaf:Organization |
| M3 | watt:MusicTrack | owl:equivalentClass | mo:Track |
| M4 | watt:Genre | owl:equivalentClass | mo:Genre |
| M5 | watt:Lyrics | owl:equivalentClass | mo:Lyrics |
| M6 | watt:hasGenre | owl:equivalentProperty | mo:genre |
| M7 | watt:hasGender | owl:equivalentClass (Restriction) | foaf:gender |

**Classes, Object Properties and Data Properties Mapping**

Classes from Music Ontology were imported and reused. This is to support the Open Linked Data movement and to link our data on the web with data in the music domain / ontologies.

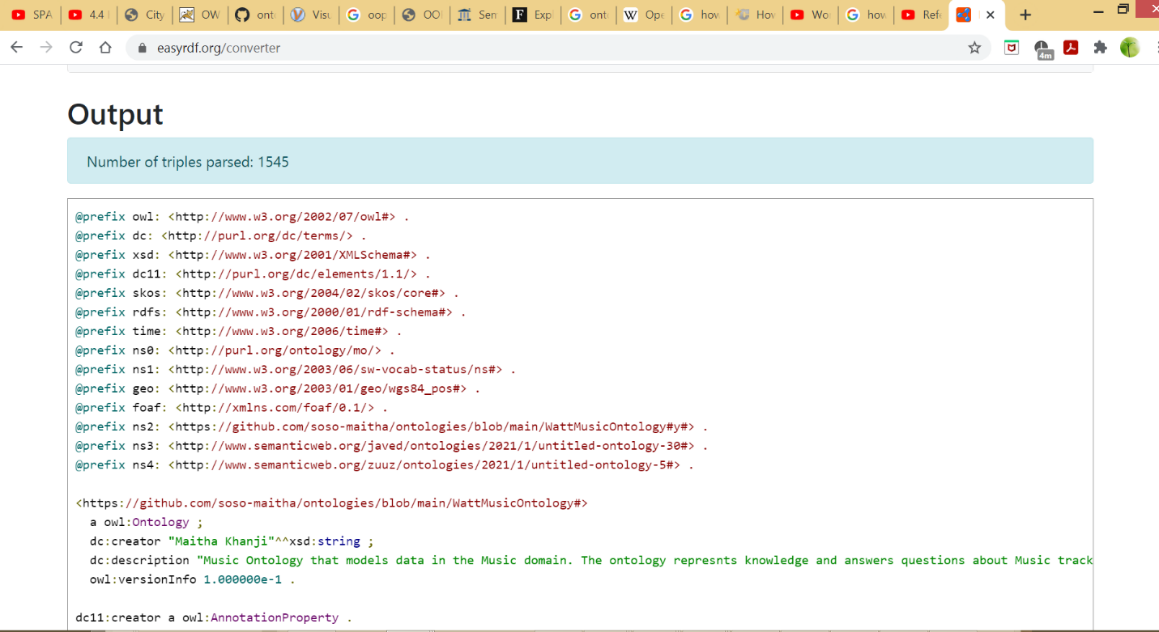
foaf is a very common and widely used ontology, therefore we used the foaf:Person class to link data with those persons on the web. For example, link Artists. The same applies to foaf:Organization class. foaf:gender data property was reused to make analysis and support research on gender balance related initiatives, for example ratio of female artist versus male artists..etc.

mo:genre object property is an important class in the Music ontology, and it is mapped with our watt:hasGenre object property. This will help classify some genres that do not exist on the current ontologies in the web. For example, the Arabic genres Maqam Hijaz, Maqam Saba and Nahawand. The same applies to mo:Gender and watt:Gender classes.

Finally mo:Lyrics class was mapped with our watt:Lyric class to support improving the Arabic digital content on the web. We have included Lyrics in Arabic language.

**Ontology Validation**

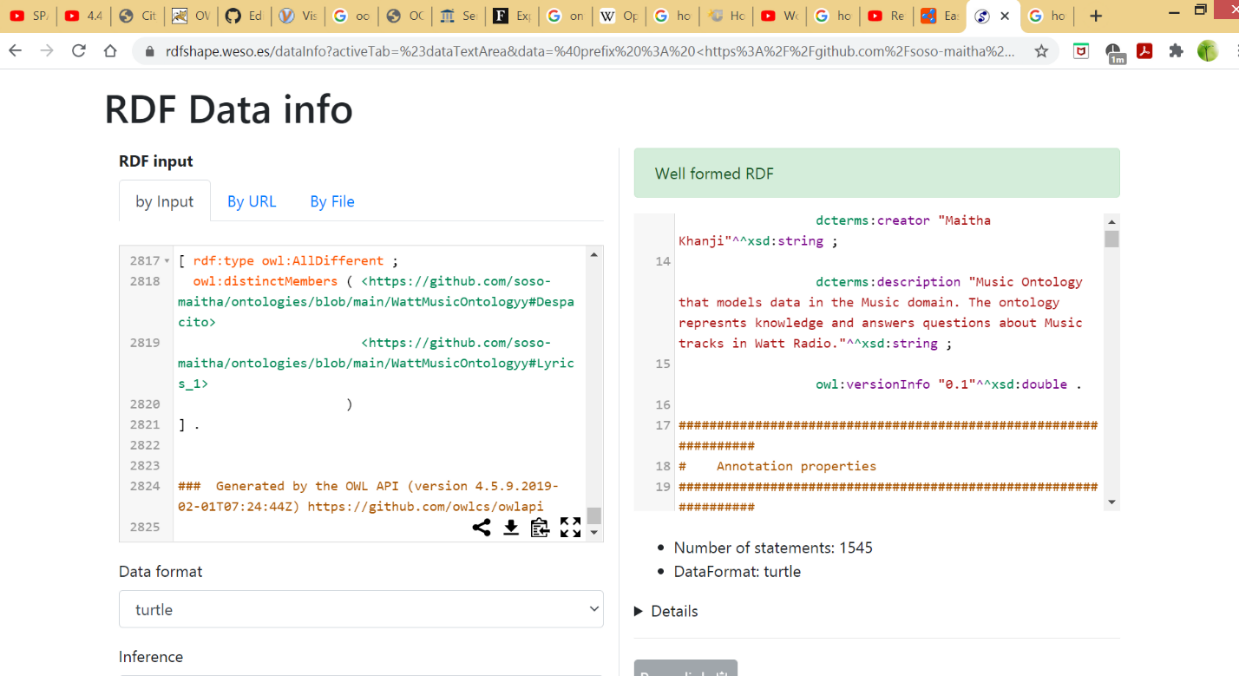
1. **Logical Validation**

****

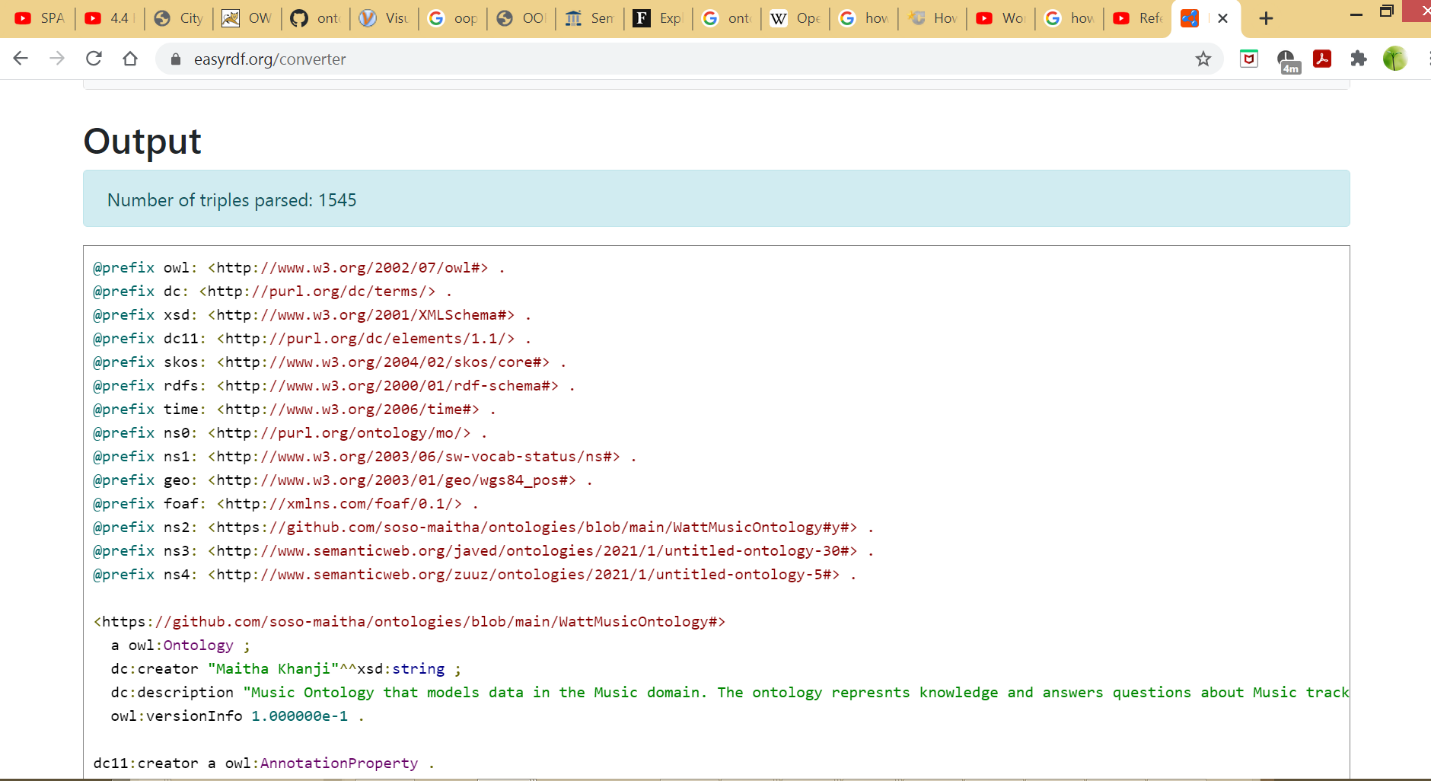
**Syntax validity**

Ontology was validated using two online validators, following are screen shots

1. <https://rdfshape.weso.es/>



1. <https://www.easyrdf.org/converter>



**Content validity, Logical validity and Application validity**

**Inferencing**

Note: there are other basic inferencing in the ontology, here the deepest ones are mentioned only

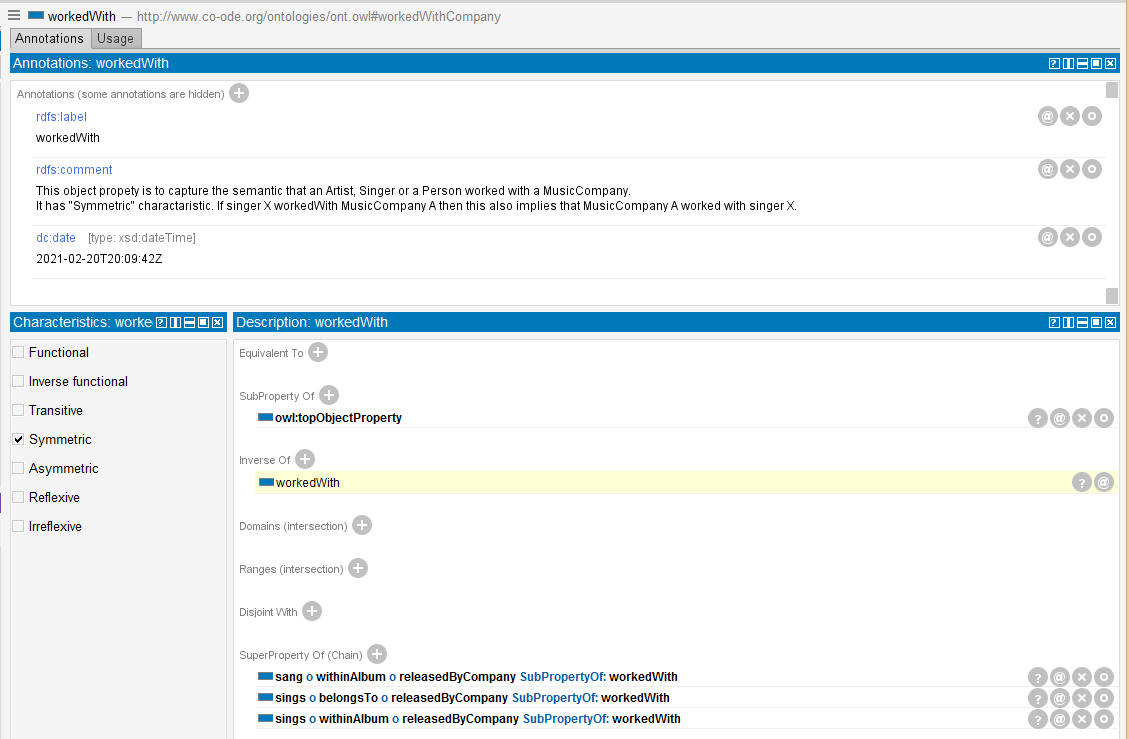
* **Inferencing Artists and Music Companies who worked with each other through Property Chain**

Object property “**workedWith**” is defined is to capture the semantic meaning that an Artist worked with a MusicCompany and vice versa. It has "Symmetric" characteristic. i.e. If singer X workedWith MusicCompany A then this also implies that MusicCompany A worked with singer X.

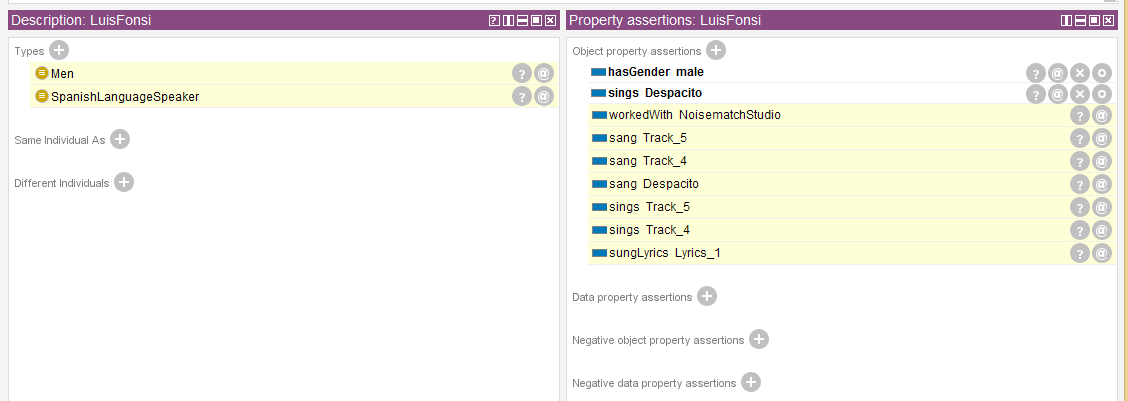
This object property is a chain of properties as the following:

sings o withinAlbum o releasedByCompany.

This means the result of all these relationships: singer sings a song X 🡪 song X withinAlbum A 🡪 Album A releasedByCompany C

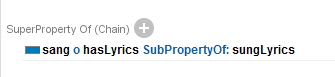


And when running the Reasoner, it infers that LuisFonsi worked with NoisematchStudio.

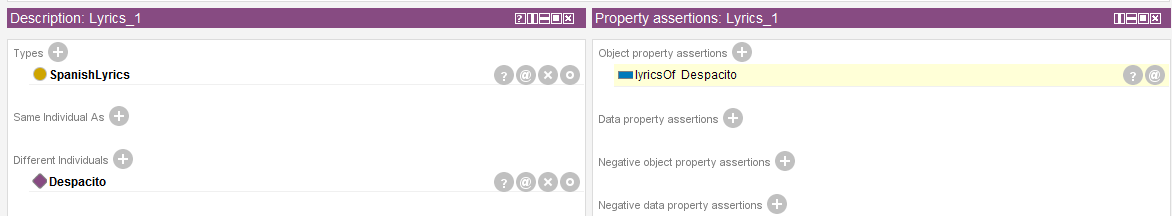


* **Inferencing that Lyrics were sung by an artist**

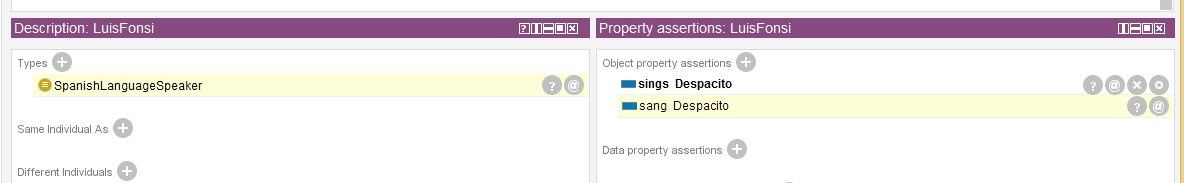
“**sungLyrics**” is a **Property Chain** to connect lyrics sung to a Singer who sung a song that contains these Lyrics. For example, UmKalthoom sung Alf\_Leila\_wa\_Leila\_Lyrics, and Alf\_Leila\_wa\_Leila\_Lyrics has Lyrics, so these Lyrics were sung by UmKulthoom



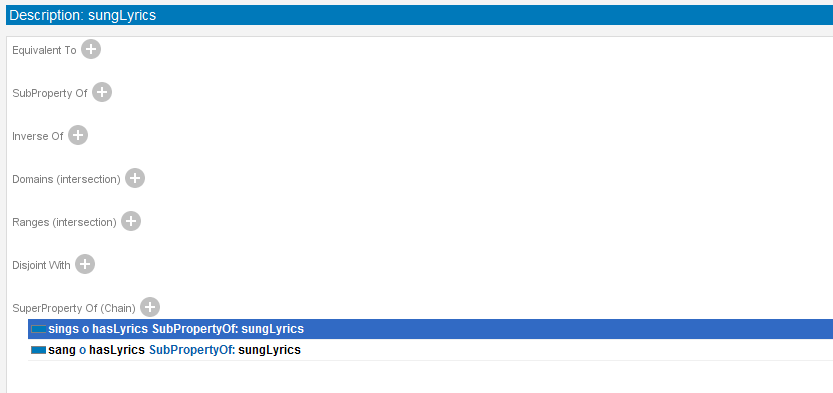
Before defining “sungLyrics” property chain, it is showing that Lyrics\_1 are just belonging to Despacito song



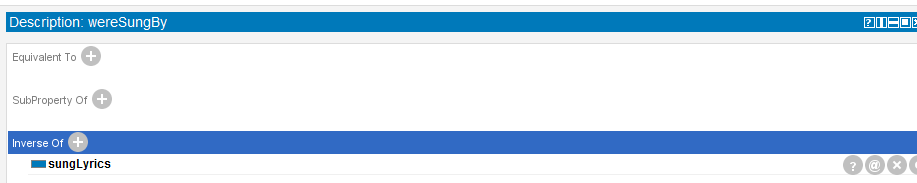
LuisFonsi (singer of Dispacito song)



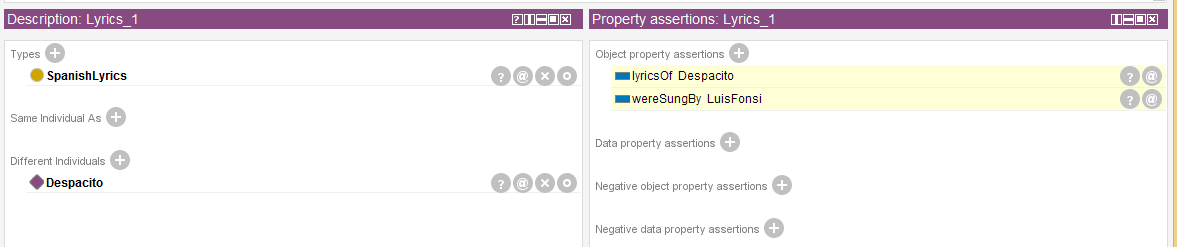
After defining the property chain “sungLyrics” and synchronizing reasoner



And defining object property “wereSungBy” that is inverse of it



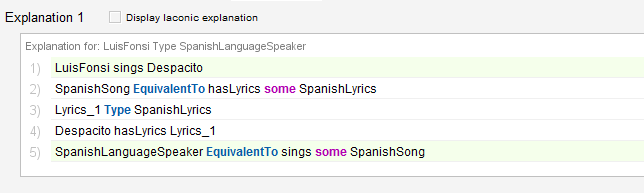
As shown below, the reasoner inferred that Lyrics\_1 were sung by LuisFonsi



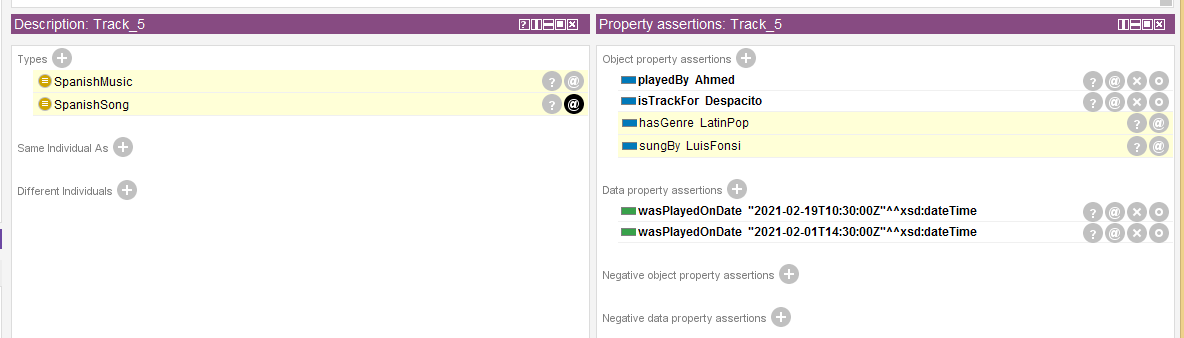
* **Inferencing speakers of languages based on song’s / lyrics’ language**

Here is an example simplified as the following:

1. Since LuisFonsi sings Dispacito song
2. Dispacito has Lyrics\_1 which is of type SpanishLyrics
3. SpanishSong is equivalent to any song that has Spanish lyrics
4. Spanish Language Speaker is an artist or a person who sings some SpanishSon



Similar chain property applies to Tracks. For example, Track\_5 isTrackFor Despacito song, and Despacito is sungBy LuisFonsi and hasGenre LatinPop. This entails that Track\_5 is sung by Luis.

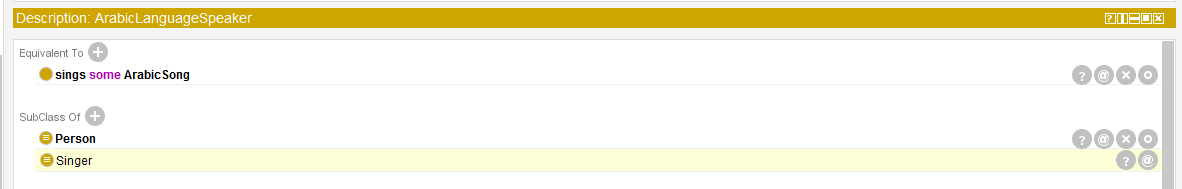


In the same example above, it is inferred that Despacito is a Spanish Song and Spanish Music.

* **Inferencing the Language spoken in a song or spoken by an Artist**

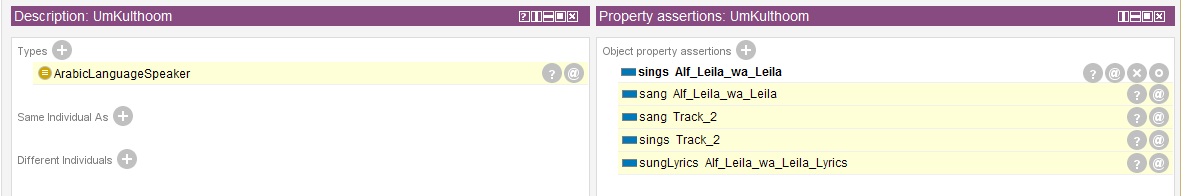
**ArabicLanguageSpeaker** class can be inferred the following Identical Property that uses existential quantification:

Equivalent To: sings some ArabicSong

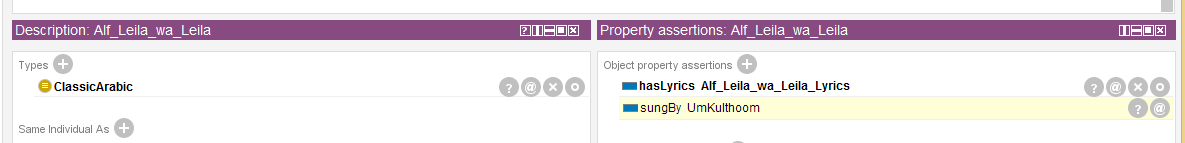


UmKalthoom inferred as ArabicLanguageSpeaker as the following:

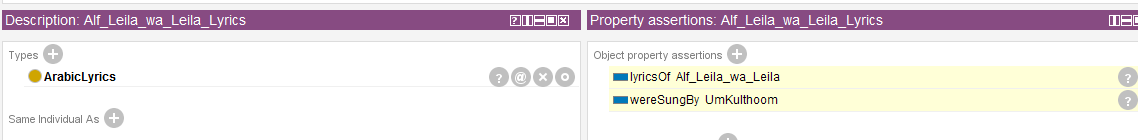
UmKalthoom is individual that has only one assertion define explicitly which sings Alf\_Leila\_wa\_Leila



Alf\_Leila\_wa\_Leila has Lyrics



These Lyrics are of type “ArabicLyrics”

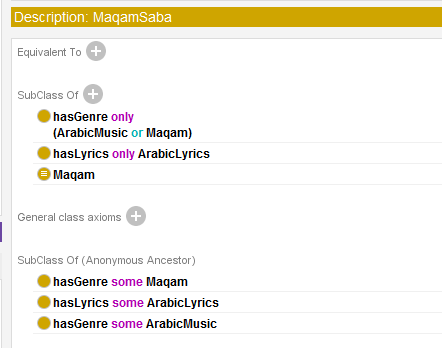


This entails that UmKulthoom is an ArabicLanguageSpeaker

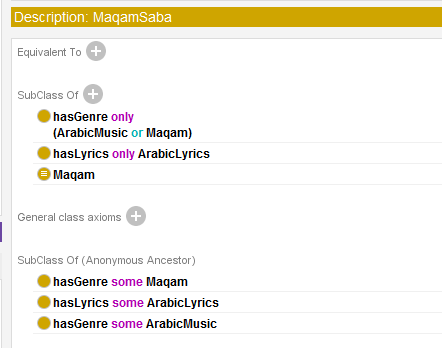
* **Closure Axiom**

Closure Axiom is a combination of universal and existential quantification. “A closure axiom on a property consists of a universal restriction that acts along the property to say that

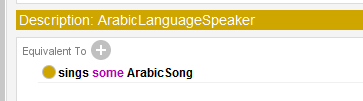
it can only be filled by the specified fillers. The restriction has a filler that is the union of the fillers that occur in the existential restrictions for the property”. (Horridge)



* Universal Quantification

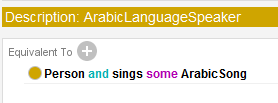


* Existential Quantification



* **Necessary and Sufficient Conditions**

Necessary and Sufficient Conditions can be expressed with the “Equivalent To” restriction which means the semantic is implicit in two ways (Horridge). In this example, an Arabic speaking artist implies that he “sings some Arabic song”, and other way holds true. Anyone who sings some Arabic song he is an Arabic speaking artist.

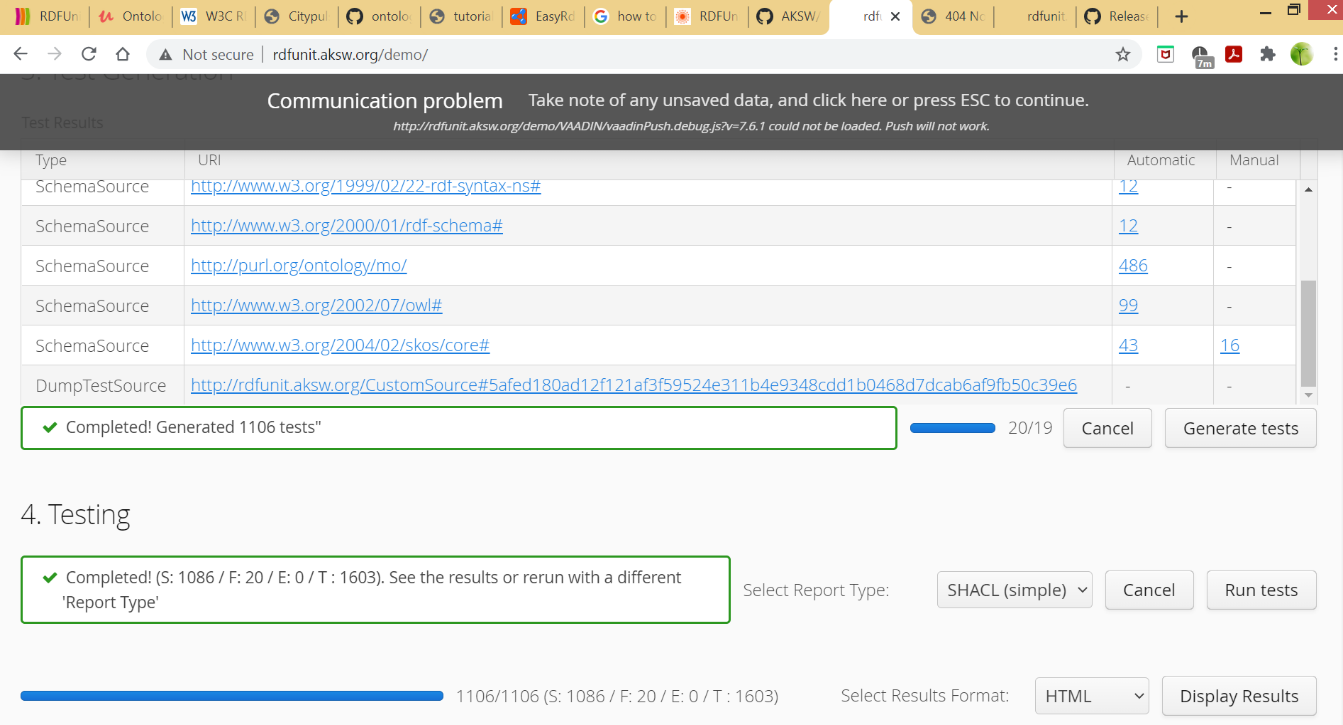


In addition to many other types of axioms, assertions and Properties characteristics

* Union / Intersection
* Negation / Disjoints
* Symmetric, Functional, Reflexive, Irreflexive, Transitive relationships

**Ontology Testing**

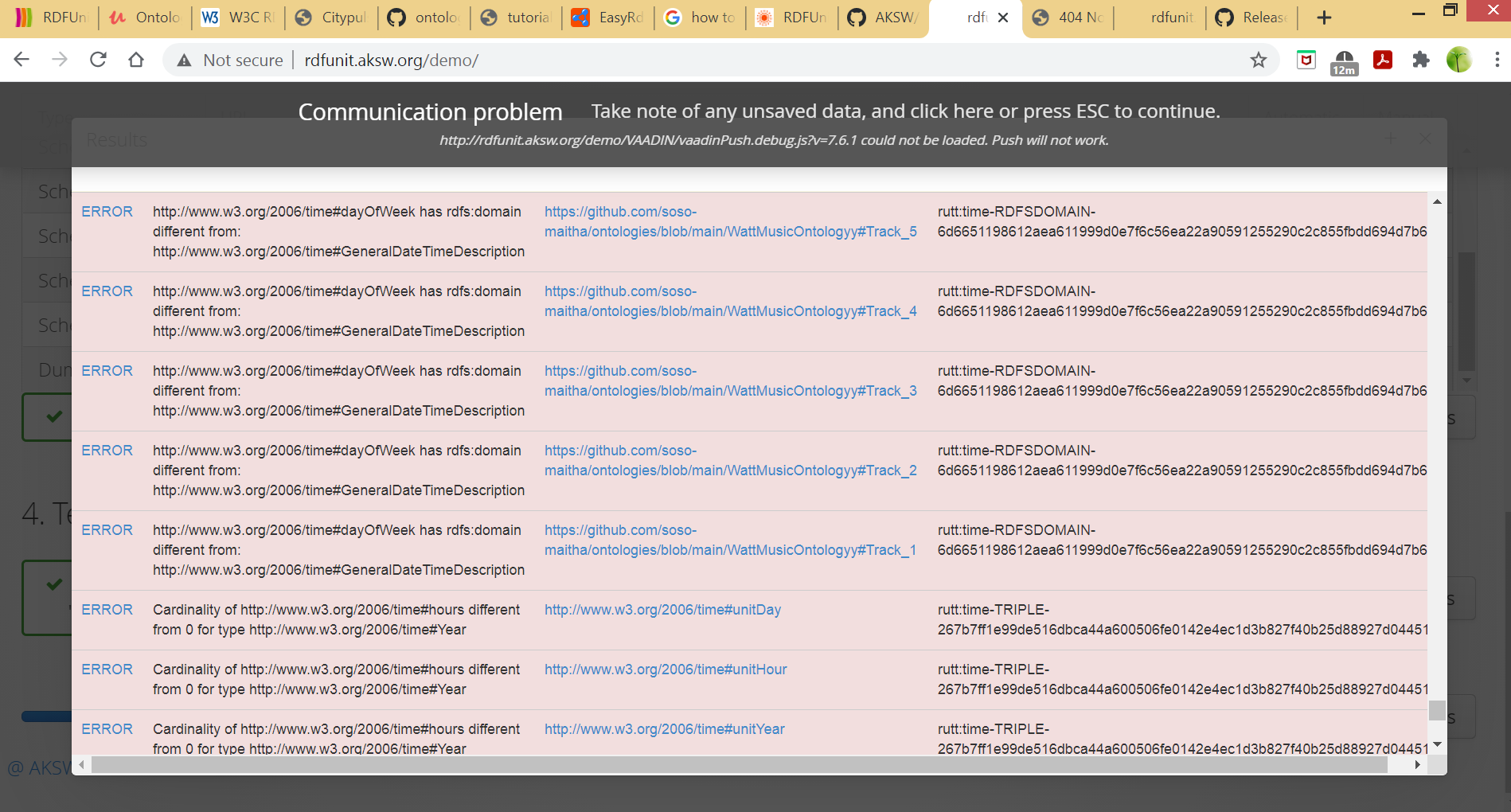
Jena Fesuki server and web-based tool were used to populate the knowledge graph and run the queries. In addition some other testing tools were used as part of our ontology validation and consistency checking. Test Suit cases were generated and executed using the online open source tool RDFUnit. Mainly most of the failed test were due to improper import of entities from the time ontology <https://www.w3.org/TR/owl-time/> . This is because temporal entity domain value was modified to our own class, for example domain “Music Track” for the imported dayOfWeek object property from time domain. Despite that this didn’t stop the reasoner, however, in unit testing this resulted in failed test.



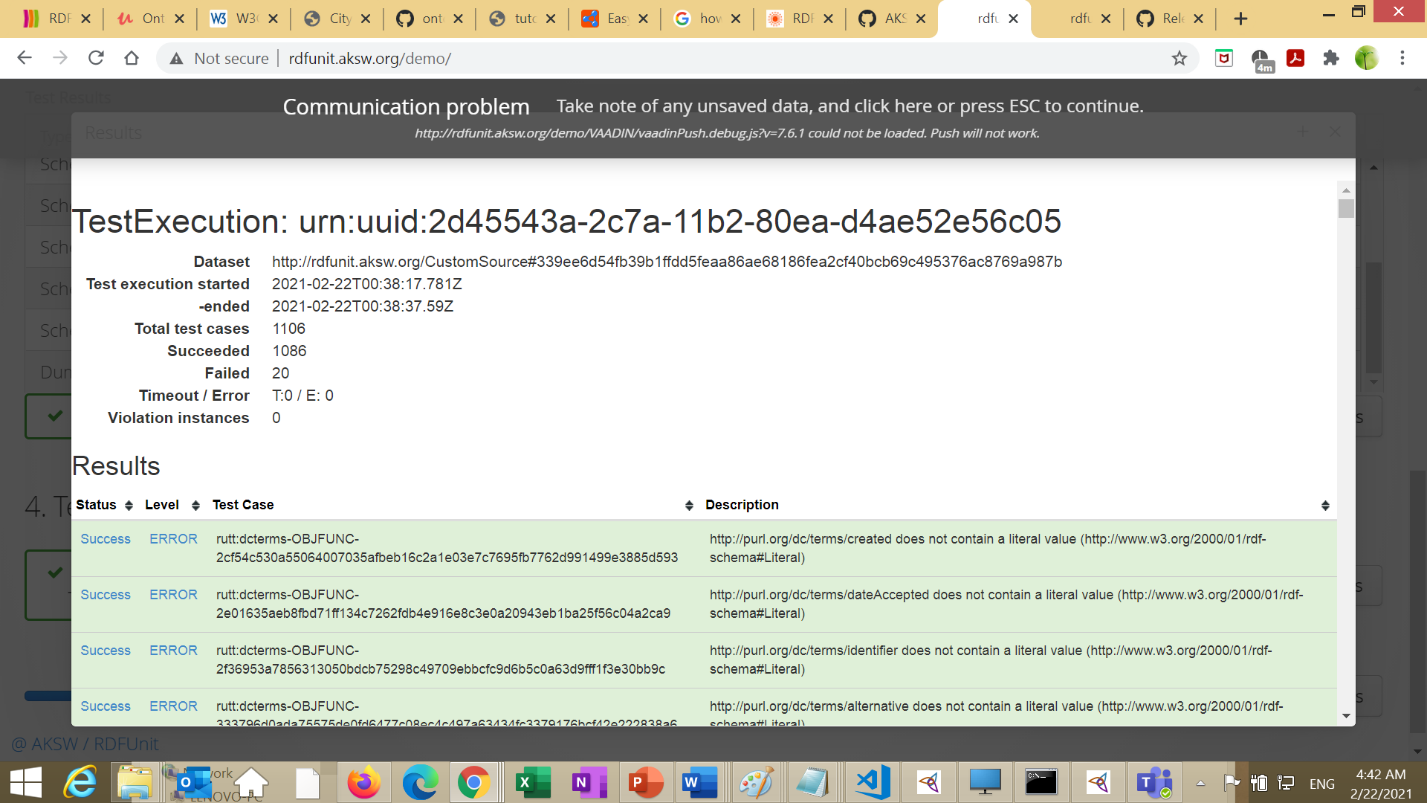
This was corrected by deleting these entities and re-importing them, and then map them with a newly defined entity that capture the same semantic.

* Invalid value for the Dublin core “creator”

Below are test results before fixing the issue.



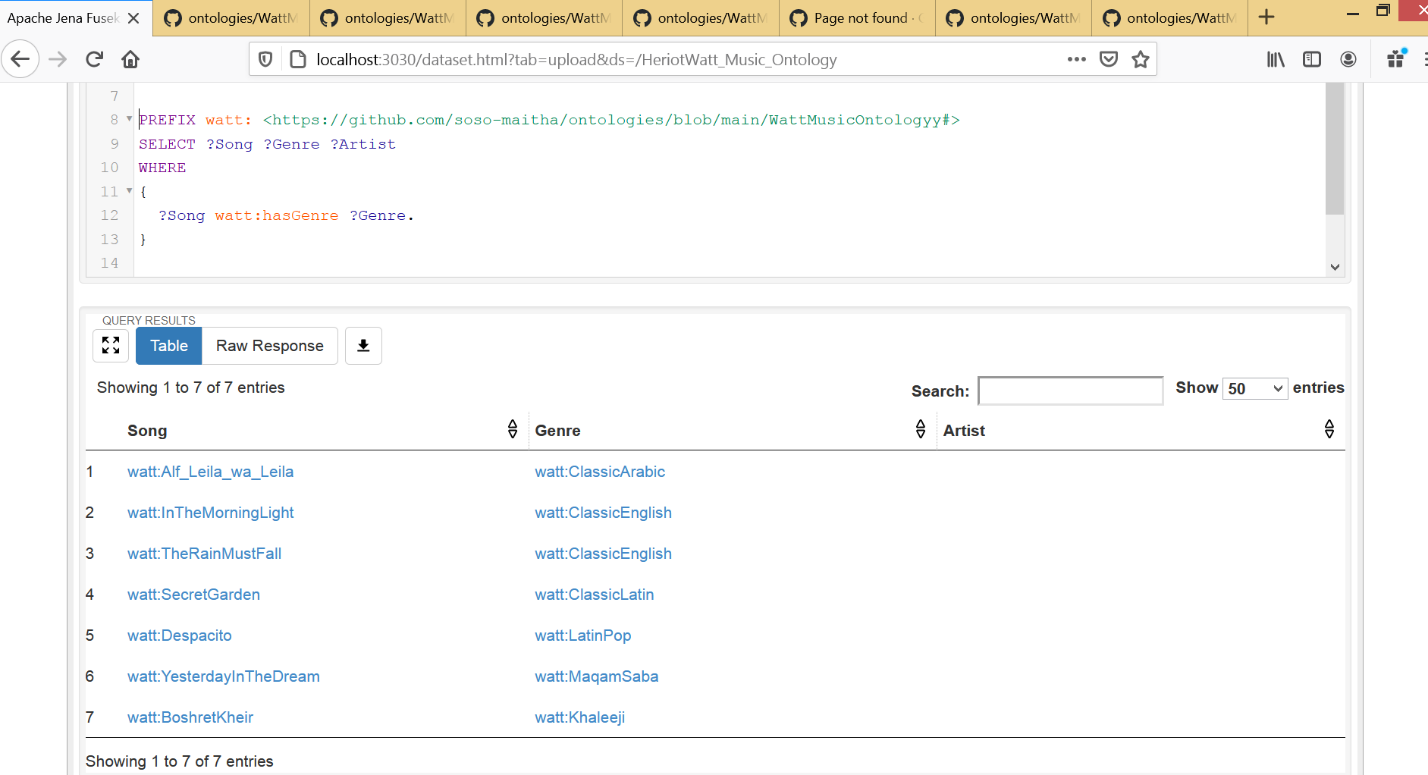
After fixing the issue.

****

**Requirements / Questions / Queries and Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Expected result** | **Sparql Query** | **Result** | **Success / Fail** |
| What are the songs played in HW Radio and their genres | List of songs and their genres | PREFIX watt: <https://github.com/soso-maitha/ontologies/blob/main/WattMusicOntologyy#>  SELECT ?Song ?Genre ?Artist  WHERE  {  ?Song watt:hasGenre ?Genre.  } | Figure-1 | Success |
| What are the songs and artists who sang them | Lists of songs and artists names who sang them | PREFIX watt: <https://github.com/soso-maitha/ontologies/blob/main/WattMusicOntologyy#>  SELECT \*  WHERE {  ?subject ?predicate ?object.  FILTER (?predicate IN (watt:sings, watt:sang, watt:sungBy, watt:sang) )  } | Figure-2 | Success |
| What are all the tracks played Heriot Watt Music Radio | List of tracks and when they were played | PREFIX watt: <https://github.com/soso-maitha/ontologies/blob/main/WattMusicOntologyy#>  SELECT ?track ?date ?DJ  WHERE {  ?track ?predicate ?date.  OPTIONAL {?DJ ?predicate ?track.}  FILTER (?predicate IN (watt:wasPlayedOnDate, watt:played) )  }  order by ?date | Figure-3 | Success |
| What is the most played song? | Details of the song that was played the most | PREFIX watt: <https://github.com/soso-maitha/ontologies/blob/main/WattMusicOntologyy#>  SELECT ?track (COUNT(?someDate) as ?count) ?song  WHERE{  ?track watt:wasPlayedOnDate ?someDate.  ?track watt:isTrackFor ?song.  }  GROUP BY ?track ?song  ORDER BY DESC(?count) | Figure-4 | Success |

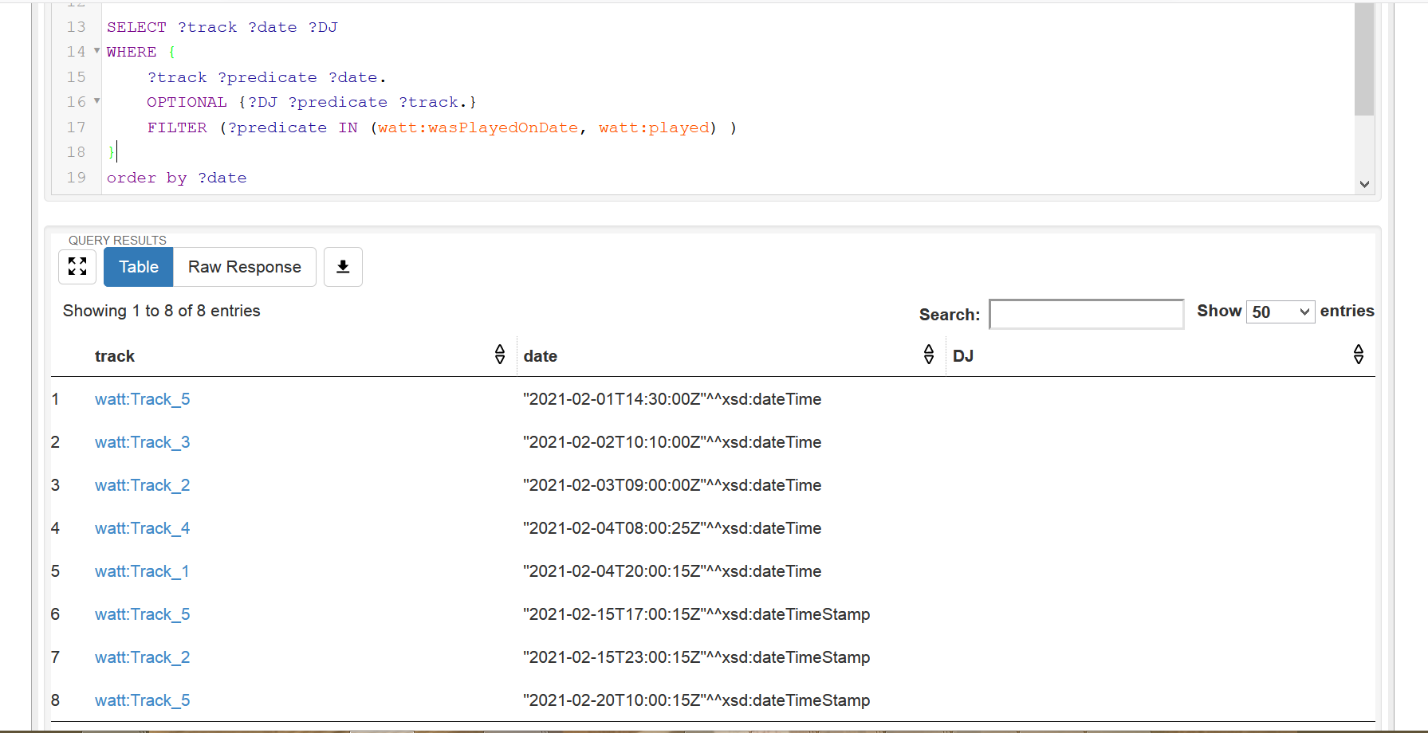
**Figure-1** What are all the tracks played Heriot Watt Music Radio

****

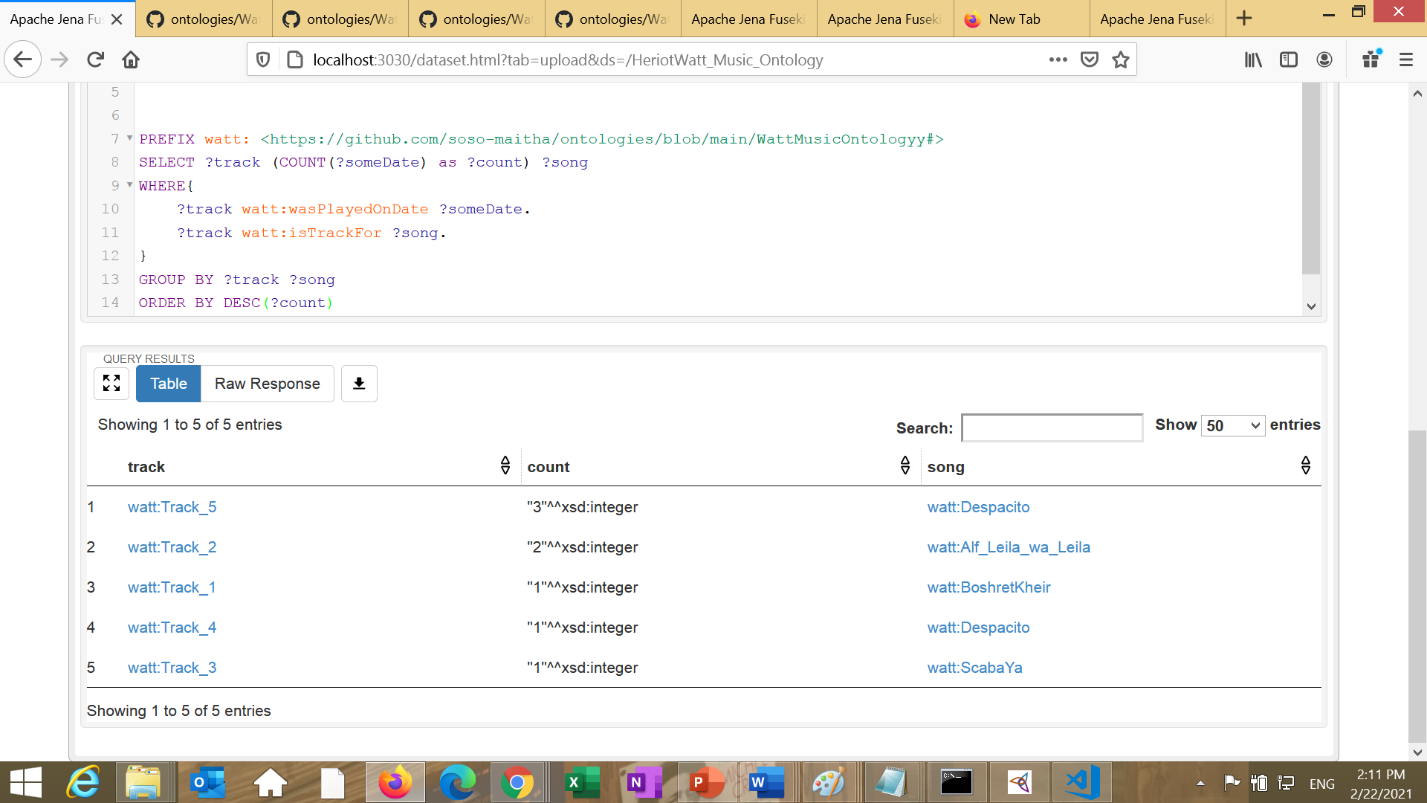
**Figure-2 (**What are the songs and artists who sang them)

****

**Figure-3 -** What are all the tracks played Heriot Watt Music Radio

****

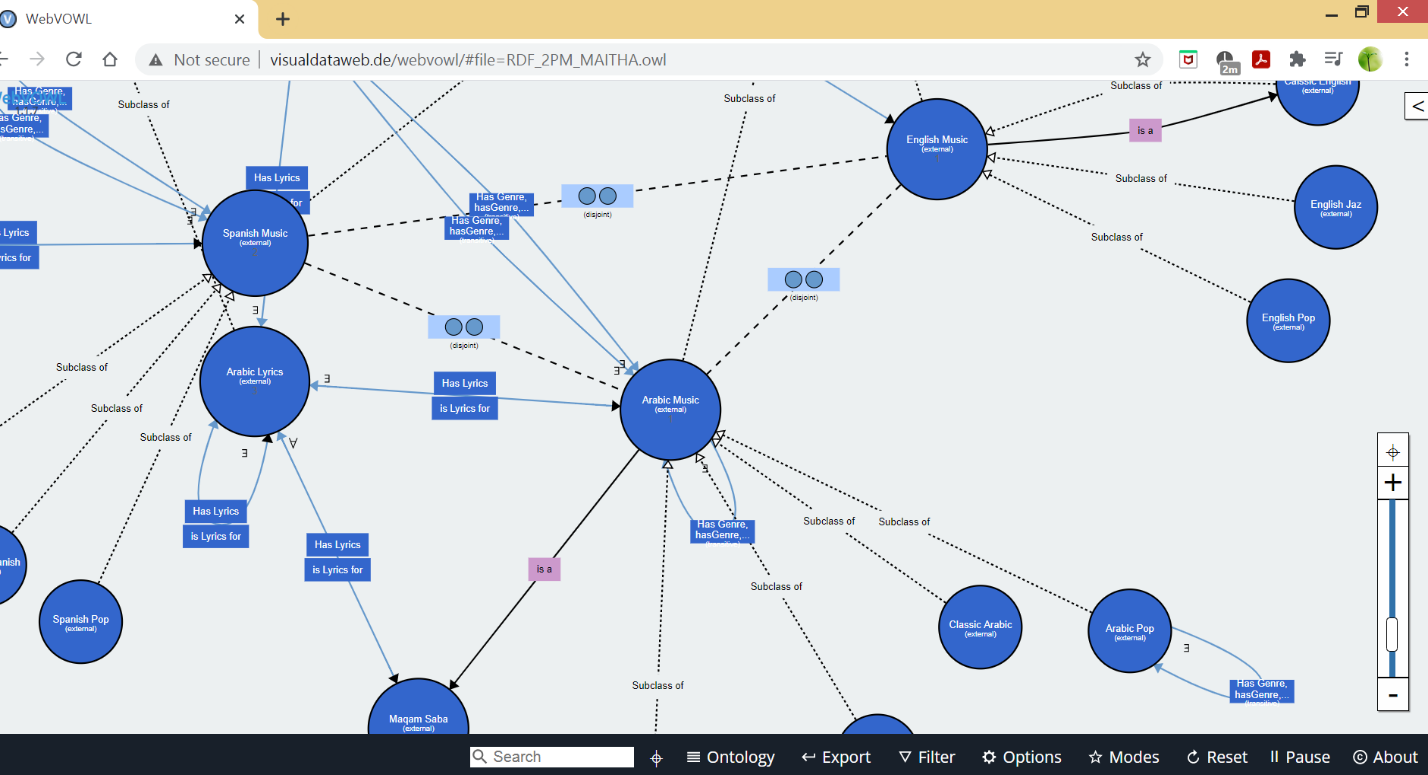
**Figure-4 (**What is the most played song?)

****

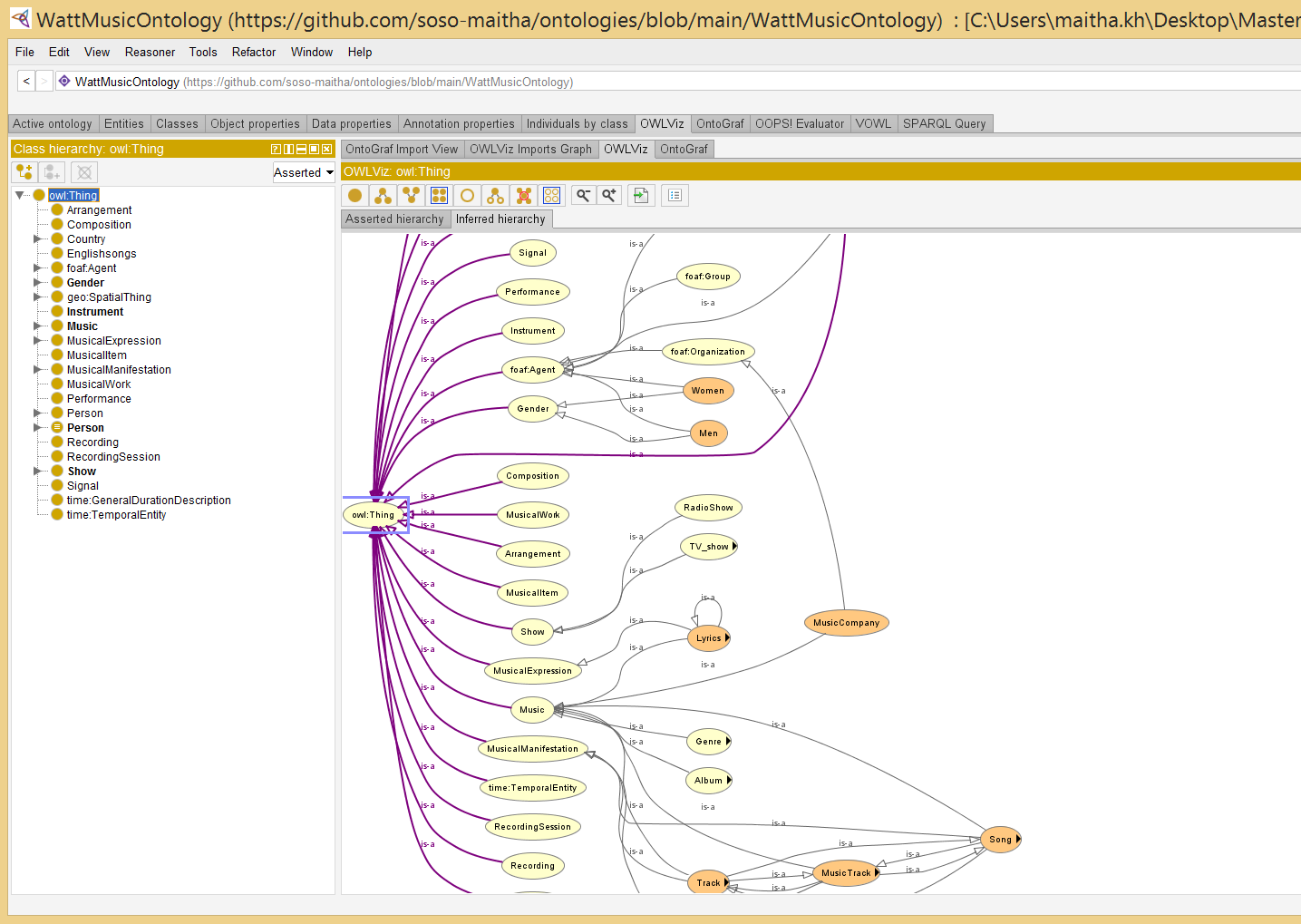
**Ontology Visualization**

Several open source out of the box visualization tools were used to visualize our ontology, some of them were used through Protégé plugins while others are web-based tools. Following are outputs for some of these tools

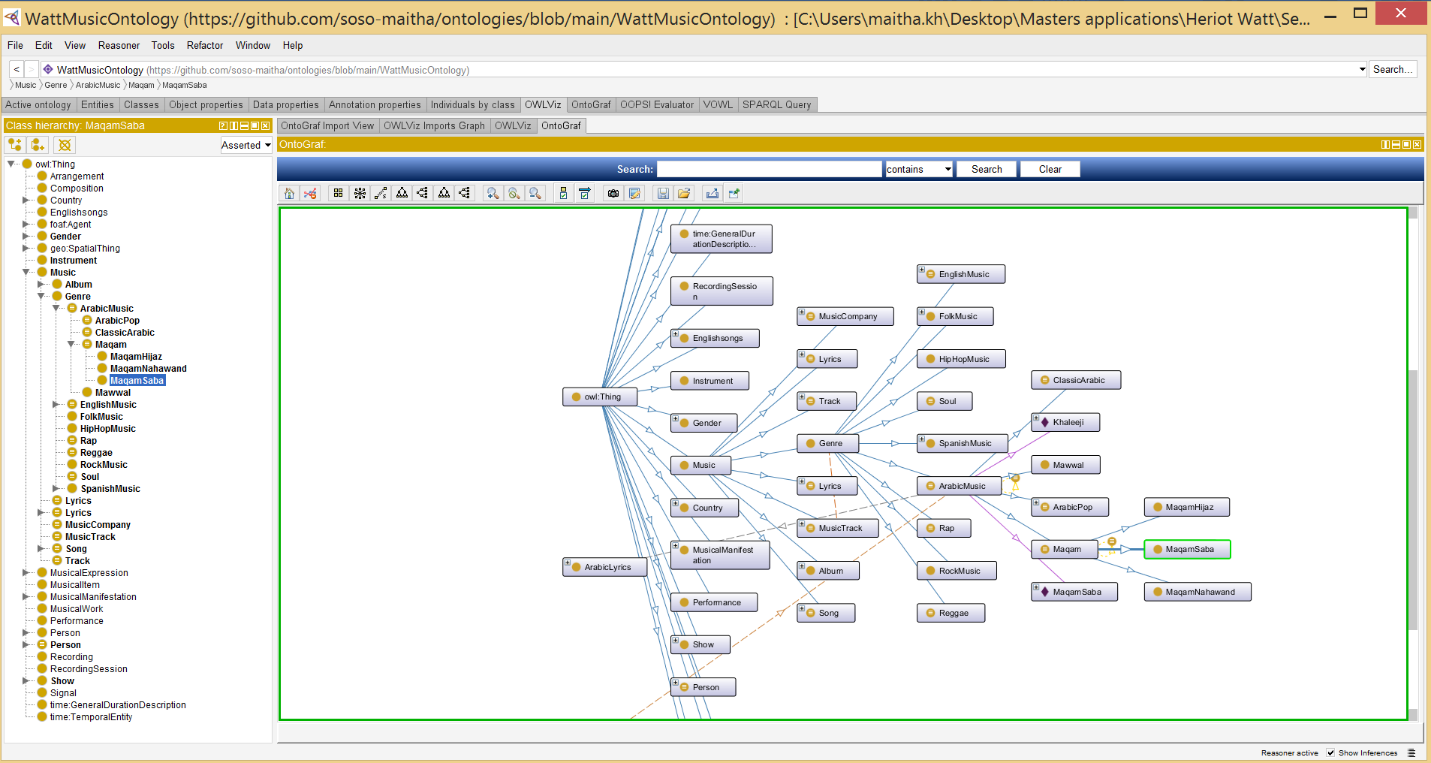
**VOWL**



**OWLViz**



**OntoGraph**



# References

Horridge, M. (n.d.). Retrieved from http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4\_v1\_2.pdf

Wikipedia. (2019). *Open-world assumption*. Retrieved Feb 22, 2021, from Wikipedia: https://en.wikipedia.org/wiki/Open-world\_assumption#:~:text=In%20a%20formal%20system%20of,is%20known%20to%20be%20true.

1. **Bakhshy, Zuhair**

# Protégé Ontology Report February 20th, 2021

Our group presents an Ontology where we have incorporated various ideas to express the DJ’s music library for WattRadio. Given the diversity of the university students that tune in, it was only fair for the DJ to have a collection of songs in multiple languages to cater to the numerous cultures.

The group struggled significantly in combining the individual ontologies as our class hierarchy was very different. In addition, each group member presumed different meanings for common object properties which meant that using the “equivalent to” function was of little use. You will find Zuhair’s ontology below which focused mainly on Rap, Reggae, & Soul music.

Timeline

Description automatically generatedGraphical user interface, application

Description automatically generated

The class hierarchy describes personalities classified by their nationalities, gender and genre. The rationale behind subclasses such as film star is used due to the many of the artist also having acting careers. This gives the ontology a bit of deeper meaning and introduces further connections.

In the individual version, I have not wrapped the WattRadio DJ’s schedule/library around the ontology. Instead this is done in the combined version. I would like to draw your attention to the inferences made by Protégé after using a series of different object properties. Here I have used a series of different characteristics to describe the nationality, language, gender, and profession using only the object properties. The data properties are used to describe the individual’s success using a range of different measures. This can be put into the context of high number of subscribers, net worth, low age, and number of active years in the industry. Below are some of the queries.

Query 1 –

This is a simple query run from the inbuilt SPARQL tab on Protégé. The purpose of this query was to list all the PlayTimes for the data available on the ontology. Due to the way my ontology was created, I am partially restricted to querying data properties. This is due to most of the object properties and class identification being inferred by Protégé as can be observed above. Of course, this meant that songs and albums were not differentiated when querying playtime.

Graphical user interface, text, application

Description automatically generated

Query 2 –

The following is a more interesting query. I have tried to create a measure of success for the personalities using various filters. The best combination for success from the existing data can be seen below, however, it is merely describing young, wealthy, popular talents – to measure success in their career. In addition, I have also added “activeSince” to provide an overview of how long each has been in the industry for.

Graphical user interface, text, application

Description automatically generated

With reference to the above data, it can be inferred that Ariana Grande has experienced a sharper incline toward wealth & fame compared to Miley Cyrus. This is also evidenced by the number of awards accumulated by Ariana Grande in her relatively shorter career.Chart, diagram

Description automatically generated

|  |  |
| --- | --- |
| Data Properties | Descriptions |
| activeSince | Year when personality entered industry |
| age | Current age |
| awards | Number of awards accumulated |
| netdollarWorth | Current net worth (2021) |
| playTime | Track length in minutes & seconds |
| releasedOn | Date the album was released |
| subscribers | Number of youtube subscribers |
| youtubeViews | Number of track views on youtube |

|  |  |
| --- | --- |
| Object Properties | Descriptions |
| contains | Album contains what song (inverse of partOf) |
| featuresIn | Which artist features in what song (inverse featuring) |
| from | Describes nationality |
| hasGender | Personality has what gender |
| hasGenre | Album has what genre |
| released | Artist released what album (releasedBy) |
| sang | Artist/singer sand what song (inverse sungBy) |
| showingOn | Which platform telecasts a show/film |
| starring | Which show/film stars X film star (inverse starredIn) |