Dipartimento di Ingegneria e Scienza dell'Informazione

Transportation in Trentino - KDI Lab 2019-2020

Document data:	**Reference persons:**
dd.mm.2019	Federico, Calabrese – 207463 Giacomo, Callegari – 207468 Bogdan, Kostić - 211827 Fabio, Molignoni – 203201 Evidence, Monday - 204729 Andrea, Montibeller – 203264

Index

Revision History

Revision	Date	Author	Description of Changes
1	11.10.2019	Giacomo	Scenario description
2	14.10.2019	Fabio	Definition of persona n. 4 (Carla)
3	14.10.2019	Federico	Definition of persona n. 3 (John)
4	14.10.2019	Bogdan	Definition of persona n. 2 (Julia & Damian)
5	14.10.2019	Evidence	Definition of persona n. 1 (Marco)
6	14.10.2019	Evidence	Storytelling Definition of persona n. 1, 2, 4 (Student, Salesman, person with special needs)
7	16.10.2019	Andrea	Definition of persona n. 1 (Marco)
8	23.10.2019	Andrea	Section 1 - Queries description
9	23.10.2019	Fabio	Section 2 - Dataset description - part 1
10	04.11.2019	Fabio	Section 3 - Dataset clean/merge - part 1
11	07.11.2019	Giacomo	Section 3 - Dataset clean/merge - part 2
12	07.11.2019	Federico	Section 3 - Dataset clean/merge - part 3
13	12.11.2019	Evidence	Section 3 - Model Design
14	13.11.2019	Fabio	Section 4 - TRANSIT and "DISI" ontology
15	14.11.2019	Federico	Section 4 - "Houda/khemaja..." ontology
16	16.11.2019	Giacomo	Section 4 - "km4City" ontology
17	24.11.2019	Bogdan	Section 3 - Model formalization description
18	3.12.2019	Andrea	Section 3 - Model formalization description
19	3.12.2019	Andrea	Section 3.1-Lexical information upload description
20	4.12.2019	Giacomo	Section 5 - Integration process description
21	4.12.2019	Federico	Section 5 - Integration process description
22	4.12.2019	Fabio	Section 5 - Integration process description
23	5.12.2019	Andrea	Section 1.1 Storytelling definition
24	5.12.2019	Fabio	Add references for integration
25	6.12.2019	Fabio	Section 6 - Knowledge graph description + section 6.1 - open issues
26	7.12.2019	Bogdan	Section 4.1 - Lexical information upload description

27	7.12.2019	Giacomo	Section 4 - Ranking of ontologies
28	8.12.2019	Andrea	Section 5 - Top-level grounding
29	8.12.2019	Andrea	Section 6 - Model visualization
30	9.12.2019	Fabio	Section 1.1 Storytelling definition - Write in tabular form.
31	9.12.2019	Bogdan	Section 4.1 - Lexical information upload description
32	14.12.2019	Bogdan	Section 7 - Final Considerations and Open Issues

Scenario description

Transportation is an important aspect of our society. People move in various ways and with several purposes: going to work or to school, shopping, meeting friends etc. Transportation is also relevant from an ecological point of view, as it is one of the largest contributors to air pollution.

Therefore, since moving around is fundamental in our life, we should try to choose the best options for the environment: avoiding unnecessary trips and using eco-friendly means are two possibilities. This is not an easy task, because people might not know the options or might not have the time to review them.

To solve this problem, we are designing an application that will help the users choose how to move around and possibly do it in an ecological way. This project will focus on transportation in Trentino, considering both public and private services: the target users will be workers, students but also tourists and everyone else interested in a sustainable transportation.

The application will use open data regarding transportation in Trentino to provide its services to the users. Information about buses, trains, cableways, car and bike sharing, road traffic is considered: the users will have access to a wide variety of means of transportation.

Section 1.1 – Storytelling definition

In this section, a more detailed definition of the scenario is provided. Four personas are presented: Marco (student), Julia (saleswoman), John (tourist) and Carla (person with special needs). For each persona, we present an itemized sequence of actions/goals that he/she may require from the system.

Persona	**Real world action**	**System action**
Marco	Go to school with public transportation.	Find how to go to a location using only public transportation.
Marco	Go to school in the least amount of time.	Find the shortest path between two spatial points.
Marco	Go to a concert while spending the least amount of money as possible.	Find the cheapest way to go to a specific place.
Marco	Arrive on time to the next “Aquila Basket” game.	Check the schedule of a specific mean of transportation.
Marco	Buy tickets for public	Find all the possible ways to buy

	transportation.	tickets.
Marco	Go to the Aquila Basket game while avoiding overcrowded buses.	Find how much people there are on a particular bus.
Marco	Go to Gianni's party without using the car.	Find, among all possible means of transportation, the one that brings you closer to a particular point in the least amount of time.
Marco	Go to school using a bike.	Find the nearest bike-sharing station.
Julia	Find a near restaurant on the street to eat.	Search for a restaurant that is on the street that she is following.
Julia	Refill the car.	Find the nearest gas station.
Julia	Arrive to the next client as soon as possible.	Find the fastest street.
Julia	Park the car.	Find parking locations around the current location.
Julia	Travel safely during a winter day.	Look at the street conditions.
John	Find a way to go to the museum using only public transportation.	Return sequence of means of transportation to take to go to a particular place.
John	Ask information to the public transportation staff in my language	Return the language spoken by the staff of a specific public transportation.
John	Go trekking.	Find the trekking paths.
John	Travel all day while spending the least amount of money	Return the cheapest tickets that allows you to travel for the most amount of time.
John	Search for family-discounts on tickets.	Return all the possible discounts, filtered by conditions.
John	Rent a bike to travel in Trento.	Find bike-rental shops.
John	Find a street path to follow which contains many points of interest to visit.	Return, from the current position, a street that passes through several points of interest.

John	Find fastest way to go to the concert.	Find fastest way to go to a particular place.
Carla	Go to the doctor using only public transportation and avoiding to use the wheelchair.	Find a way to go from point A to point B while walking as less as possible.
Carla	Find the nearest bus stop, so that I can avoid move a lot with the wheelchair.	Find the nearest bus stop.
Carla	Visit her friend.	Find if the bus stop is accessible with a wheelchair.
Carla	Visit her friend.	Find if the bus has room to accommodate the wheelchair.
Carla	Move using public transportation.	Return if there are any strikes/interruption of the public transportation service.
Carla	Use public transportation and dismount at a specific bus stop.	Find if a bus has changed its path due to works on the street/manifestations.
Carla	Buy tickets online, so that she has not to go physically at the ticket store	Find how to buy tickets online.

Section 1.2 – Personas description

Marco is a 19-year-old student at the University of Trento. He lives with his parents, as well as his little sister Tatiana, in Pergine Valsugana. In order to get to school in Trento early, Marco chooses to use Trentino Trasporti because it passes through the fastest route. He also takes advantage of getting tickets online, when he does not have to go to school, in order to save some money for his other needs. On some days, Marco is required to go to pick his little sister from school because his mother is not able to. To do this with great ease, he uses Trentino Trasporti as his preferred means of transportation because it is accommodating and has space specifically for little children. Marco is also a huge Aquila Basket fan and never misses the weekend games in Trento. This is essentially made sure because he chooses Trentino Trasporti which navigates through the fastest/shortest route and easily connects his city to the sports palace. This Wednesday, the basketball team will play the *“7DAYS EuroCup”* and Marco is not worried about being on a crowded bus because he will use the Trentino Trasporti services that guarantee an efficient connection.
Marco has a grandma that stays at the elderly home of Meano. He loves to spend time with her; hence, he goes to see her on Sundays. This Friday is a big day because Marco will be attending a party at his friend but this time he will be going there by

using the bike provided with the Trentino Trasporti card. In all, Marco is ensured safe, fast, convenient and accommodating means of transportation using Trentino Trasporti.

Julia and Damian are partners in an effective salesman team. Every week they are assigned to distribute goods and make sales in different cities. This week they were in Rovereto to present their new product to a customer. For this purpose, they needed to know the fastest way to get there as the appointment with the customer was made quite spontaneously. However, the employees of Julia and Damian's company are usually urged to choose to shortest route for reasons of cost-efficiency and sustainability.
After arriving at the customer's office, Julia and Damian found a place to park their car using a designated application. After their presentation at the customer's office, both of them were hungry and wanted to take lunch. Since Julia and Damian don't know about spots to eat in Rovereto, they looked for nearby restaurants on their phone. On their way back to Trento, Julia and Damian had to refuel their car. For cost-efficiency purposes, they headed for the cheapest gas station in the area.

Carla is a 70-year-old pensioner who lives in Mezzocorona with her husband Mattia. She worked all her life as a seamstress in Trento and now she spends most of her time visiting friends and having fun at the "Circolo pensionati di Trento", a local meet-up organization for old people. Unfortunately last winter, while going to the grocery, she broke her femur and now she is on a wheelchair. Due to this accident, Carla is no longer able to drive the car to go regularly to Trento. Since she is no more independent, she has to rely on public transportation in order to move around the region. Fortunately, Mezzocorona is well connected to other locations in Trentino: there are trains (like Trenitalia and Trento-Malè) that connect the town to Trento, then from Trento several buses can be used to move to the desired position inside the city. She is interested in finding ways to go to the various locations of meet-up of the Circolo Anziani using just public transportation. Moreover, she wants to know whether the buses, trains, stations and bus stops are accessible with a wheelchair. Moving through the city with a wheelchair may be a difficult task, therefore Carla is interested in finding the nearest bus stops around her location. Time is not one of her major concerns, therefore she can accept to use several means of transport in order to get to the desired destination. A priority for her is to arrive as near as possible to the destination while performing the least amount of distance using the wheelchair. She may also accept to pick a longer bus route if this means having to move for a shorter distance using the wheelchair.

John is a 47-year-old winemaker, living in Monaco with his wife Anna and his two sons Josh and Emily. John and his family want to spend a weekend in Trento to visit landscapes and for John to visit the Trentino wine houses. Being the first time in Trento, he does not know how to get around the city, so he decides to use an application that allows him to obtain information about Trentino Trasporti, timing and related costs, look for places to go hiking in Trentino and ask for information on the best places to walk or bike. To move around, a card is suggested in which children travel for free and can run both Trento and province for 72 hours, changing transportation services based on their needs, including the rental of bicycles. Wanting to take a bike ride with the family, he checks where the nearest bike rental station is and then they head towards Monte Bondone. On the way back they decide to stay longer and arrive at Mattarello for dinner, so they check if there is also a bicycle depot, because given the tiredness, the shortest way to arrive at the Hotel is the bus. The following day, John decides to visit the Cantine Ferrari for lunch, and the fastest way is by bus from the Hotel taking a panoramic tour. In the afternoon he then goes to Cantine Mezzacorona again by bus, opting for a view of the city. For the return to the hotel, he is advised to use the train plus the bus to get there faster. On the last day, to take advantage of the 72 hours, John and his family decide to spend the morning at Lake Caldonazzo, given the proximity and the limited time available. Having spent the morning, John decides to take time to save the coincidence with the direct train to Monaco, as the application indicates slowdowns of the bus resulting from the traffic, saving time and distance from the station in case they had chosen the bus.

From informal to formal

Section 2 – Queries description

In order to formalise a complete list of competency queries and to derive the query patterns we started with a brief analysis of the storytelling defined for each persona.
The four personas have, because of their uniqueness, different priorities and necessities. Marco, a 19-year-old student, uses public transportation to move to his favourite places and to save money since, as a student, he does not have large finances.
Carla, an elderly woman, uses public transportation in order to easily connect to the hospital or the pharmacy and by subscribing to the pensioners service she has special offers. Last but not least, Trentino Trasporti offers adequate structures on their vehicles.
Julia and John (a business woman and a tourist) have similar priorities. Both are interested in finding the fastest or the cheapest path for their itinerary. However John, being a tourist, might also be interested in finding the most panoramic itinerary, the nearest bus station or to check if the staff of public transportation speaks more than a single language.
Julia, when opting for public transportation, is interested to know if there are different transportation classes, Internet and other services so she can work on board in the most comfortable way.
Once analysed their storytelling, we defined 30 competency question for each. Immediately we noticed that some of the competency questions, such as:

- *"Nearest transport station?"*
- *"Which transport can I take?"*

were not unique for the single personas but shared between them. Thus, before proceeding with the definition of the queries we divided the competency questions according to the following criterion: unique of the single persona or not. In the link attached below we show how all the competency questions in common between Carla, John, Julia and Marco are inserted in a specific group whose cells were not filled with any color. Instead, the unique competency questions of each persona are filled with a specific color.
Having applied the over mentioned criterion, we proceeded producing the query pattern with their labels and attributes. The attributes are keywords that define the competency questions, the labels have the task to link each query with a specific problem to solve.
The work for this task has been equally divided between the members of the group covering the modeling task.
The final output of this section is attached in the link below where the competency questions and query descriptions are shown:
Competency Question and Queries

Section 3 – Model Design

The model design phase produces as main output the informal model. To accomplish to this task we made use of yED[1], which gives a clear overview of the structure of the datasets at a glance. This phase started with an understanding of the datasets gathered by the data integration team and the generalized queries from the section above. Then, decisions were made about the core, common and auxiliary entities based on their level of importance. Along with defining these entities, their relations and attributes were defined as well. Entities represent classes of objects that have properties in common and an autonomous existence, while relations define the relationships between entities in the model structure and attributes are features (properties) owned by entities.

Core entities are "independent elements" returned by the competency questions and queries. In this project we are examining the data from the domain of transportation. Hence, the core entities are transportation and location. That is, all queries from different domains return transportation, location, ticket etc, as their base.
For example, let's analyze the competency query "Is a shared bike available at this location?". A possible instance is:

```
                                                SELECT Public Transportation
Is shared bike available at this location?   >     FROM Transportation
                                                   WHERE Shared Bike
                                                   AND available = 1
```

We can observe that in the instance above, the query returned transportation as its base.

The selection of common entities was based on re-usability, i.e. entities that can be re-used across different domains. Some of the common entities in this project are *Itinerary* and *Place*. As seen in the model, any type of Public Transportation will require a structured attribute *Ticket*. Using same format as the example above, we can have:

```
SELECT Public Transportation
FROM Transportation
WHERE Class IS Train
AND id = 519
```

The auxiliary entity types are general patterns that can determine how to filter results. In simple terms, they are entities that can be used to filter through structured results when a search is carried out. The auxiliary entity types in this project include *Train*, *Bus*, *Car*, *Bike*, *Service Area*, *Parking*, *Shared Car Station* etc. For example, Train is associated with 'Public Transportation', while 'Service Area' can be used as a filter to search through a result containing 'Location'.

[1] "yED" https://www.yworks.com/products/yed

So, if we want to find a train with a specific class, we can search using 'train' as our search tag. We have:

SELECT train
FROM Public Transportation
WHERE class = B

As established above, each entity has relationships with other entities. In this project, the different dependencies include type, requirement, possession, availability, route etc. A simple example is represented by core entities 'Transportation', 'Public Transportation' and 'Private Transportation'. A good example is among entities Street, Shared Car Station, and Location.

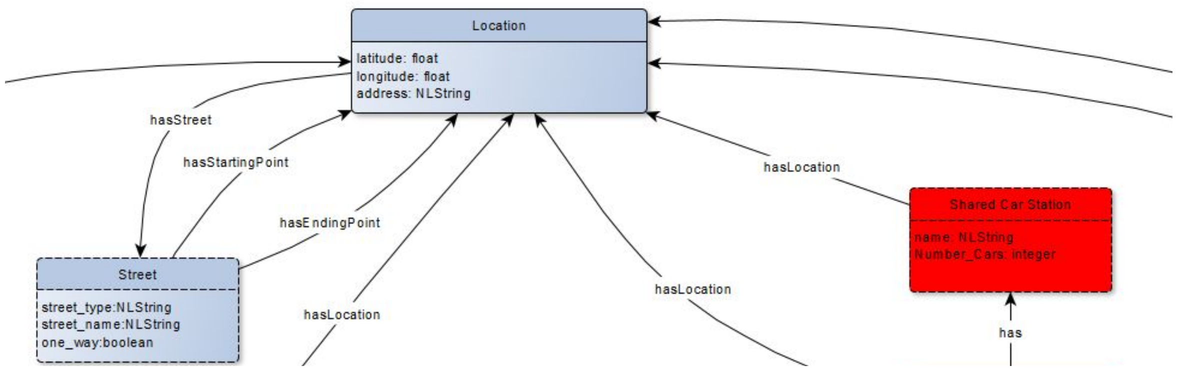


Fig 1. Image showing entities, relations and attributes

From the figure above, we can see the relationship between entities *Location*, *Street* and *Shared Car Station*. These relationships are tagged *hasLocation, hasEndingPoint, hasStartingPoint* and *hasStreet.*

From figure 1 above and the examples explained so far, we notice that each entity has various attributes. The attributes simply represent features of the entities. In this project, the attributes were assigned based on the structure of the dataset. Specifically, from the dataset, each entity has column headers and these are represented as attributes in the model design. For example, the core entity type *Location* has attributes 'latitude', 'longitude' and 'address'. Attributes also have data types which can be boolean, NLString, float etc.

For an overview of the EER model click here.

Section 4 – Model formalization description

To build the ontology for the final knowledge graph, we made use of the software Protégé[2]. Starting from the EER model we had, it was easy to define the first "draft" of our ontology. Fine-tuning our ontology model, together with our EER, was not so immediate. This task requires the precious feedbacks coming from the group taking care of the data integration. More than a single time we noticed that what seemed formally correct for us was not suitable for the integration of the data. Another problem regarded the availability of data in order to cover specific competency queries.
For example, not having any data about the eco-friendly contributions produced by Trentino Trasporti with the money of their customers, we decided not to care about these queries and as consequence to remove the information from the EER and the ontology model.
Instead, we preserved the entities and the properties caring about the queries such as the ones concerning the transportation staff. These ones, in our opinion, may be something that in the near future can be easily integrated adding values to our ontology.
This "ping-pong" phase was fundamental in order to finally obtain the official formal model.
Starting from the final EER that we defined as described above, we began with the construction of ontology classes. The core class of our ontology is *Transportation* with its two subclasses *Private Transportation* and *Public Transportation*. Along with the other entities of our EER model that are represented as classes in the ontology, we chose to define some of the entity properties of our EER model as classes, e.g. *eco_fuel*. This is the list of all classes we made use of in our ontology:

[2] "Protégé" https://protege.stanford.edu

- eco_fuel
- Energy_Source
- Itinerary
- Location
- Place
    - Parking
    - Service_Area
- Schedule
- Shared_Bike_Station
- Shared_Car
- Staff
- Station
- Ticket
- Transportation
    - Private_Transportation
        - Bike
        - Car
    - Public_Transportation
        - Bus
        - Cableway
        - Train
- Transportation_Card
- Transportation_Classes

The second step consisted in defining the object properties. These ones describe the relations between entities of the previously defined classes. They correspond to the edges of the EER model, but also reflect relations between ontology classes that are entities in the EER model and ontology classes that are properties in the EER model. Each of the object properties is assigned a domain and a range, which correspond to one of the classes. This list contains all the object properties we used in our ontology:

- bikeStationHas
- bus_hasEcofuel
- bus_hasEnergySource
- busHasClass
- busHasStaff
- cablewayHasStaff
- car_hasEcofuel
- car_hasEnergySource
- hasBikeStation
- hasItinerary
- hasPlace
- hasSchedule
- isA
- itineraryHas
- placeHas
- placeHasLocation
- requires_card
- requires_ticket
- scheduleHas
- sharedCarHas
- stationHas
- train_hasEcofuel
- train_hasEnergySource
- trainHasClass
- trainHasStaff

Last but not least, we defined the data properties of our ontology. As ranges for the data properties we used standard data types of the XML Schema Definition[3], e.g. *xsd:float*. The domain of each data property corresponds to one of the ontology classes. The data properties coincide mostly to our EER model entity properties. This list shows all the data properties we made use of in our ontology:

[3] "XML Schema Part 2: Datatypes Second Edition" https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/datatypes.html

- condition
- eco-friendly
- eco-friendly_contribution
- eco_fuel
- electric
- electric_plug
- electric_support
- electricity
- energySource
- engine_type
- fastest
- first_departure
- gasoline
- hiking
- ID
- language
- last_departure
- latitude
- leaves_at
- longitude

During the formalization of our model we also had to face some ambiguity introduced by Protégé. One in particular was the fact that if we had one shared property between multiple classes, Protégé was not able to handle it anymore associating it only to one class. In order to solve this problem we defined a class having that single property to which the classes to which it belonged were connected through their object properties.

Section 4.1 – Lexical information upload description

Once defined our model, we processed it with OOPS[4]. This additional step was useful to understand if there were critical or important issues in the ontology. One of them was the lack of lexical information since having lexical information referring to the natural language associated with the ontology is fundamental for its reusability.
In order to provide the correct informations, we made use of WordNet[5]. WordNet is an online tool that provides a large lexical database of English nouns, adjectives, verbs and adverbs. We decided to focus only on the English database because, in our opinion, it makes our ontology more reusable and understandable. Another point is that in English the amount of information is more than enough to not require a further generalization.
Nouns, verbs and adjectives are grouped into sets of cognitive synonyms interlinked by means of conceptual-semantic and lexical relations. These sets of cognitive synonyms are called synsets.
Starting from our ontology, we annotated each class, object property and data property looking for their definition in WordNet. During this process we not only made our model better defined but we also denoised it, specializing it and reducing the level of ambiguity.
One problem that we faced while annotating our ontology was that WordNet utilizes other lexical terms for some of the concepts we used. For example, none of WordNet's concepts for *street* fit our concept for the *Street* class, so we had to profoundly look for another concept that would better fit our purpose. The final concept we used for our *Street* class was *road (ID: 04103160)*, which fits very well our idea of the concept.

[4] Poveda-Villalón, María, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. "Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation." *International Journal on Semantic Web and Information Systems (IJSWIS)* 10.2 (2014): 7-34.
[5] Princeton University "About WordNet." WordNet. Princeton University. 2010.

Another issue we had to deal with was the fact that not every concept we use in our ontology is available in WordNet. To fix this problem, we thought about several approaches, e.g. leaving out the lexical information for the missing entries or changing our ontology in order that every class and property has a respective entry in WordNet. However, to ensure that our ontology is precise enough, in the end we decided to extend WordNet's database with the missing concepts. This extension was done by providing a definition for the missing concepts and connecting them through the hypernym relation to other synsets of the database.
The extension can be found in the file *wordnet_extension.tsv* which contains the newly created ID of the inserted entry, the lexical entry itself, the part-of-speech tag of the entry, the ID of the entry's hypernym as well as the entry's definition.
The following list contains all the synsets used to precisely describe our ontology:

Synset ID	One of the synonyms in WordNet
13580156	source
08633886	itinerary
00037365	location
08682181	place
13800883	parking
08666356	service area
06507319	schedule
08456947	staff
04313218	station
04103160	road
06530710	ticket
04480667	transportation
04025345	public transit
02636270	have
00757790	require
08508037	address
00362929	discount
15205769	arrival time
00184353	available

13943868	condition
13472977	consumption
14710042	delay
15158573	duration
04262144	socket
11470903	electricity
05848697	type
00979699	fast
14710042	diesel
14710074	gasoline
00289814	hiking
07285306	identifier
06293304	language
08613276	latitude
15205929	departure time
06344646	name
08614224	longitude
08077282	manufacturer
05128718	number
00235970	one-way
02571853	panoramic
05152365	price
15137796	time
15144316	travel time
04602355	WiFi
02837983	bike
02961779	car

02927500	bus
04475240	train
02937835	cable car
KDI01	eco-fuel
KDI02	shared bike station
KDI03	shared car station
KDI04	transportation card
KDI05	transportation class
KDI06	bike support
KDI07	continuous transportation
KDI08	eco-friendly
KDI09	eco-friendly contribution
KDI10	electric support
KDI11	engine type
KDI12	shared car
KDI13	stop ID
KDI14	transportation ID
KDI15	transportation card type
KDI16	private transportation type
KDI17	wheelchair accessible
KDI18	first departure
KDI19	last departure
KDI20	private transportation
KDI21	other services

Section 5 – Top-Level Grounding

This section maps the formal model we obtained with the TRANSIT[6] top-level ontology model that we found in Linked Open Vocabularies[7]. We merged both the models linking ours with the top-level ontology as well as declaring some entities as equivalent. This is done to enable reusability and optimization of our ontology with the top level one. Essentially, a user upon utilizing our ontology can also access entities and properties present in the top-level ontology. Part of the output obtained upon mapping is, as seen in the yED diagram, below:

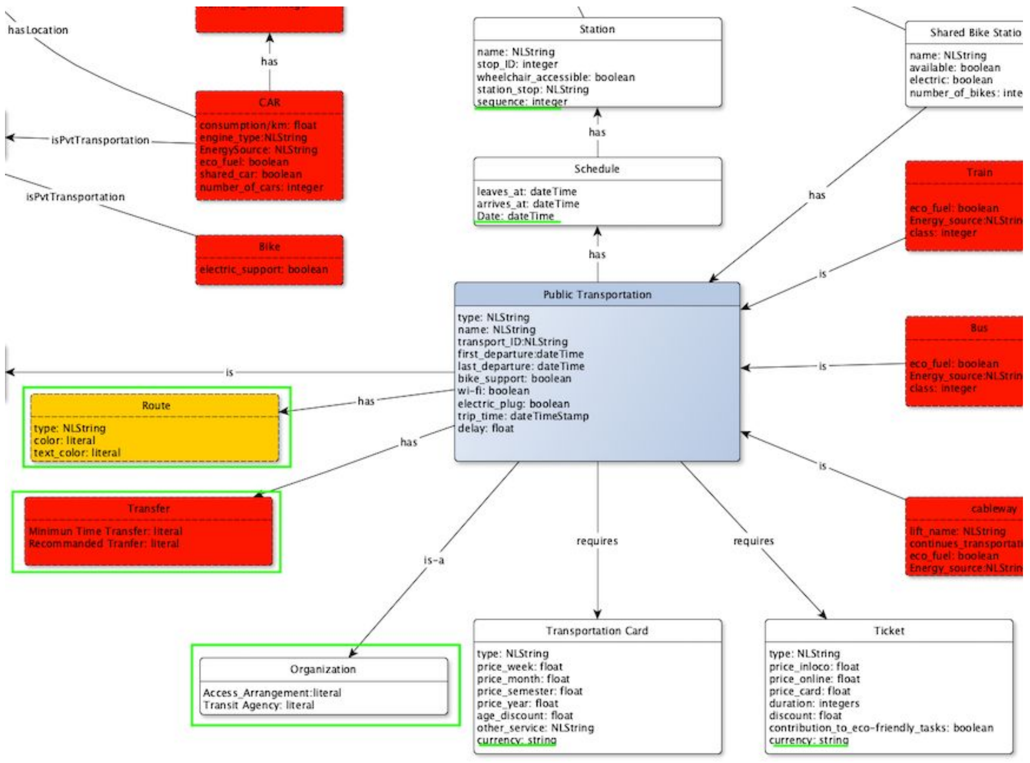


Fig 2. Image showing entities added after mapping

The information surrounded by a green line were part of the top level ontology and has been added in the process.

Accomplishing this task was not easy. First of all we had to find a top-level ontology in which similar entities were present or that shared our same goal. In fact our purpose was not only to define a model for transportation, but to enhance the values of the public and sustainable ones.
Once found the top level ontology, it was important to understand, looking at the lexical information of both models (ours and the top-level), which entity classes were similar or identical. In case we had identical classes also in terms of attributes, these two could become a single one. Instead if the lexical information was the same but the attributes were only partially shared, we could merge them together in order to obtain,

[6] "TRANSIT: A vocabulary for describing transit systems and routes" https://vocab.org/transit/
[7] "Linked Open Vocabularies" https://lov.linkeddata.es/dataset/lov/

as in the previous case, a single class.
In the top-level ontology we noticed that the class *"Service"* was by definition our class *"Public Transportation"*. Starting from this one it was easy to map the other entity classes of the top-level ontology to our ontology. For the class *"Schedule"* or *"Station"* instead we found equal definition with *"Schedule"* and *"Transit Station"* (in the top-level ontology), but different attributes. Thus, joining them, we composed two equivalent classes *"Schedule"* and *"Station",* having all the attributes.
One last problem was due to the fact that not all entity classes of the top-level ontology were useful for us. Therefore, in order to solve this problem we decided to remove them from our final ontology since these ones were not adding any useful information for our purpose. Entity classes such as *agent*, *license*, *person*, *document*, *text* and *work* were deleted from our final output.

The table below gives an overview of how the mapping was done.

Top level Entities	**Mapping relationship**	**Domain roots**
Schedule	Equivalent To	Schedule
Service	Equivalent To	Public Transportation
Transit Station	Equivalent To	Station
Fare	Equivalent To	Transportation Card & Ticket

Domain roots	**Mapping relationship**	**Top level Entities**
Public Transportation	has	Transit route
Public Transportation	is-a	Organization

We applied same reasoning also for data properties:

Top Level Data Properties	**Mapping relationship**	**Domain Data Properties**
Sequence	Equivalent To	stop_ID
Arrival_time	Equivalent To	arrive_at
Departure time	Equivalent To	leaves_at

The same reasoning holds here as in the entity mapping. Click on the links below for a better overview of the yED diagram and the merged ontology.
merged owl ontology, merged EER diagram .

Section 6 – Model Visualization

The final output, that is possible to visualize and use by clicking the two links above, is shown in figures 3 and 4 below.

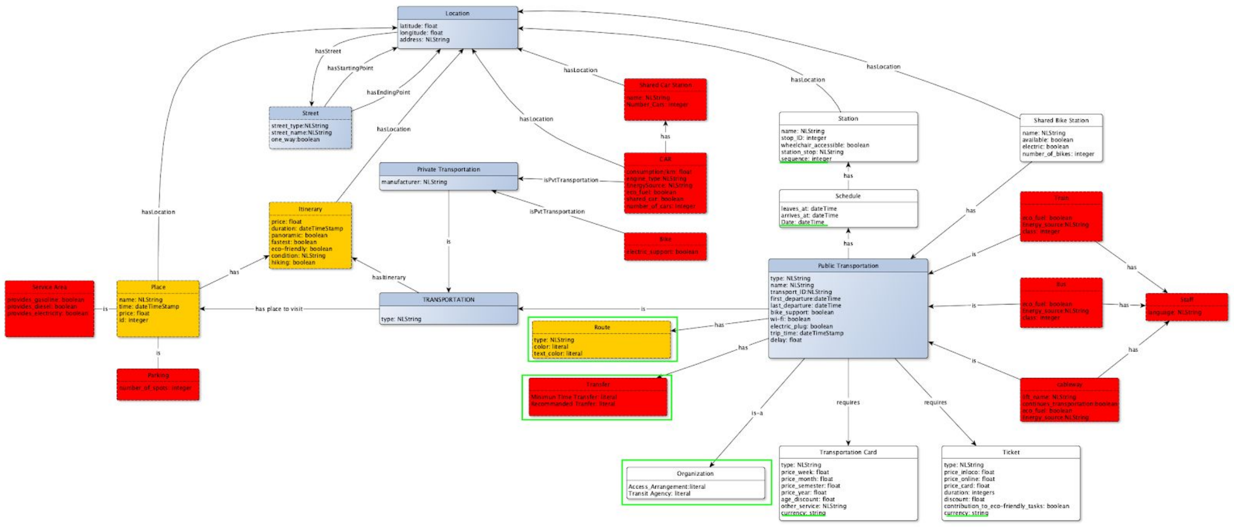


Fig 3. EER final model

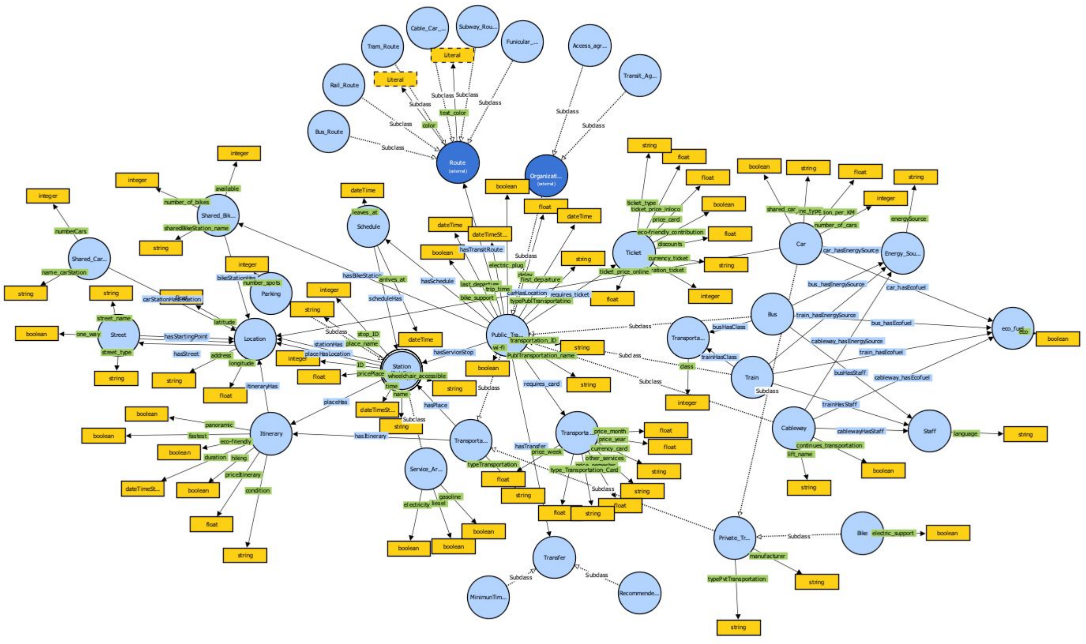


Fig 4. Final ontology

As already mentioned, being our goal the possibility to generate an ontology for Transportation able to cover as best as it can the entire field in Trentino, we had to carefully select the classes and the attributes able to add the values that we need to our formal model.

The choice to maintain classes such as *"Transit Route", "Transfer"* and *"Organization"* was given by the fact that all of these were adding extra information, useful to accomplish to our goal.

"Transfer", for instance, tells what is the minimum amount of time between arrival and departure to assure a connection, *"Transit Route"* specifies the line to follow for a specific connection and *"Organization"* states who will provide the connections.

This, together with the addition of *"currency"* to the classes *Ticket and Transportation Card,* as well as *"sequence"* and *"Date"* to *Station* and *Schedule*, let us reduce the level of ambiguity of our model, also making it more precise and less ambiguous. In this way we are able to provide a useful and reusable ontology that can be used to cover both public and private transportation providing, once integrated with the used database, information about what is the best route to follow, the nearest station or public transport, the consumption of the private vehicles, the cheapest itinerary, other information about transportation card or tickets and finally what is the more sustainable way to move.

The final model may be divided in two main parts, both seen as "children" of the core entity transportation. The first one is "Private Transportation". Here we try to formalize all the queries referring to the use of a private car and bike, indicating the manufacturer, the consume per kilometer and other peculiar attributes of the two types of private transportation cited above. Apart from this, it is also included the concept of shared car station in which a user can retrieve information about this service. The second "child" of the main class "Transportation" is "Public Transportation". Here we decided to take care of multiple types of transportation: Bus, Train and Cableway. This entity class is particularly full of attributes because of the huge amount of data that we were able to retrieve. We can provide information about fundamental things such as the price of the ticket or transportation card, the ID of the transportation, the time schedule, station, etc.

In the end the one of the most complex part was understand how to correctly model the "Itinerary", "Place", "Location" and "Street".

These entities not only add information about the price, the duration, the condition of the street or the type of the itinerary, etc. but also provide spatial information useful for navigation.

In the end, with this ontology we tried to provide a tool not only useful to the user but at the same time the most complete as possible to afford the task of transportation, whatever the type was (private and public) with an eye on eco-friendly and sustainable solutions.

Section 7 – Final Consideration and Open Issues

After the integration of our formal model with the data provided by the integration group, we can be satisfied by the final results and its functioning. The ontology represents well the set of queries defined in Section; only one of them is not well represented in this moment. Furthermore, the choice of the top-level ontology permits us to extend our model and to complete it giving extra value to our work. More general evaluation on the overall project were done with the data integration group. The result of this analysis can be read in the Section 7 of the data integration part.

As we said, one open issue was for example that we are not able to represent the attribute related to the spoken languages for the auxiliary class "Staff". This is due to the fact that this information is not available yet. Being able to retrieve it in the future may add to the usefulness of the knowledge graph.
Moreover, some means of transportation that can be used to move through the region of Trentino are not represented by our ontology, for example taxi. Adding these missing types of transportation would allow to use our ontology in more situations and therefore make it more complete.
Another improvement of our ontology would be to expand the area we consider from the region of Trentino to for example all of the area of Italy or an even larger area. For this purpose we would have to consider all existing types of transportation in that designated area and extend our ontology with the missing entities and attributes. This extension would enable our ontology to be used by a lot more people.

Data Integration

Section 2 – Datasets extraction and generation description

First of all, in order to find all the data needed for the application, we analyzed the storytelling, persona descriptions and competency questions. From these resources we extracted a list of datasets that we had to provide. We divided the data into three main areas: "points of interest" (i.e. useful facilities and locations), "public transport" (containing all data concerning transportation using means of transport such as trains and buses) and "other" (which contain some useful generic datasets). All the data along with the scripts used were pushed to the GitHub repository. In the following part, a list of all the downloaded data is provided, along with the explanation of the extraction process. The three main resources that we used to retrieve data are OpenStreetMap[8], OPENData Trentino[9] and Trentino Trasporti[10].

Points of interest

- **bike sharing stations:** downloaded from here.
- **carsharing positions:** downloaded from here**.**
- **fuel stations**: obtained from OpenStreetMap by querying the Overpass API. Additional data has been downloaded from here and here.
- **panoramic:** downloaded from here and here.
- **trekking:** downloaded from here, here and here.
- **street indications:** obtained from OpenStreetMap.
- **parking**: obtained from OpenStreetMap.
- **charging stations**: obtained from OpenStreetMap.

Public transportation

- **Position and accessibility of bus stops and train stations, schedule and path of buses and trains:** for Trentino Trasporti the data was downloaded from here. Unfortunately, Trenitalia does not provide open data. In order to get the data we had to create a Python script and extract information from the ViaggiaTreno API. The scripts used to scrape the data can be found in the GitHub repository, with the names "scrape_trains.py" and "train_detail.py".
- **delay of buses:** data was scraped from the "Viaggiare in Trentino" API using the script "tt_delay.py" that can be found in the GitHub repository.
- **Cableway stations:** downloaded from here**.**
- **Cost of tickets**: downloaded from here.

Other

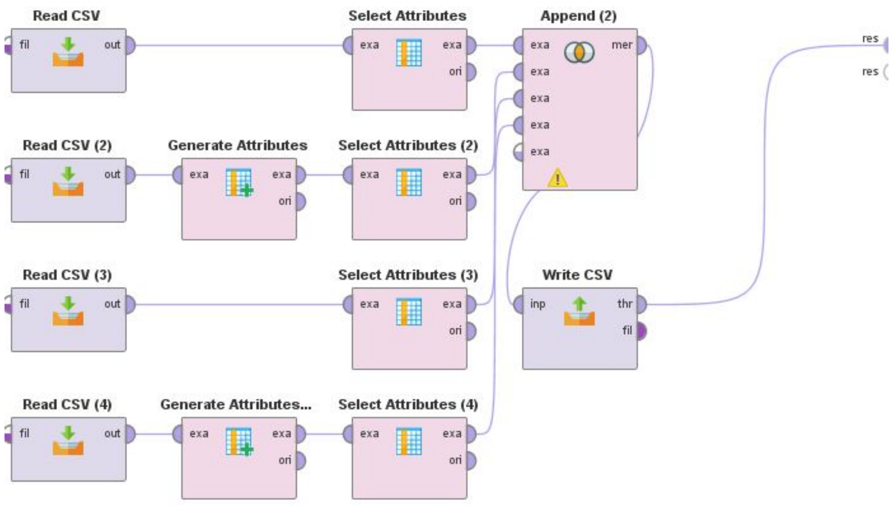
- **Traffic**: downloaded from here.

[8] "OpenStreetMap." https://www.openstreetmap.org/.
[9] " Dati Trentino." https://dati.trentino.it/.
[10] "Trentino trasporti." https://www.trentinotrasporti.it/.

- **Bike Paths:** downloaded from here.
- **City:** downloaded from here.
- **Street:** downloaded from here and here.

Section 3 – Datasets cleaning, merging and analysis description

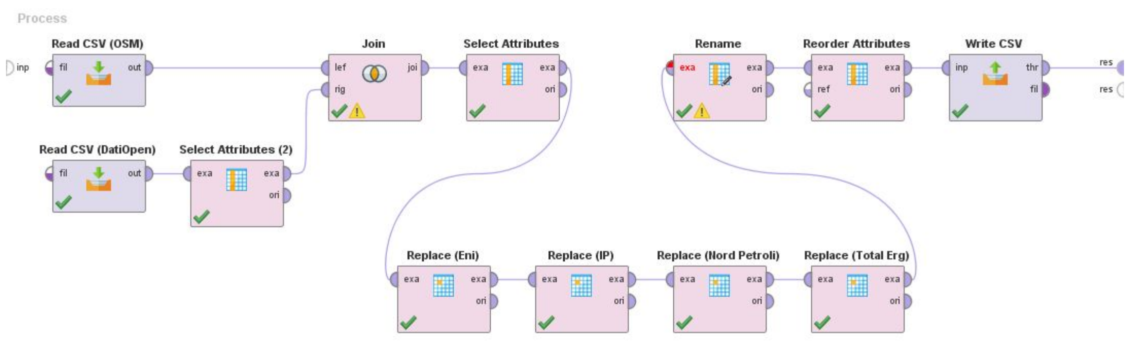
Once all the relevant datasets were downloaded, we started cleaning and merging data in order to produce the final datasets that would be used in the integration phase. We used "RapidMiner"[11] in order to work on the data in an easy and fast way. All the final datasets can be found in the GitHub repository.



The generation of the "cableway.csv" dataset inside RapidMiner

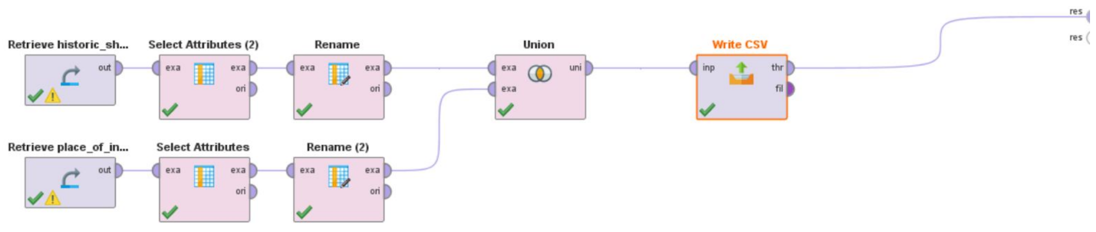
In the image above it is presented an example of computation performed through RapidMiner. In particular, the tool was used to merge together four different datasets, each containing a specific type of cableway. Two datasets missed a column, and the value of that column was inserted by analyzing the content of the dataset (i.e. the "tipo veicolo" column for the "cableway_va_vieni.csv" was not present, but it contained only cableways of type "va e vieni", so the value was inferred). Once all datasets had the same attributes (filtered using the "Select Attributes" tab) the final merge was performed in order to obtain one single dataset.

[11] "RapidMiner." https://rapidminer.com/.



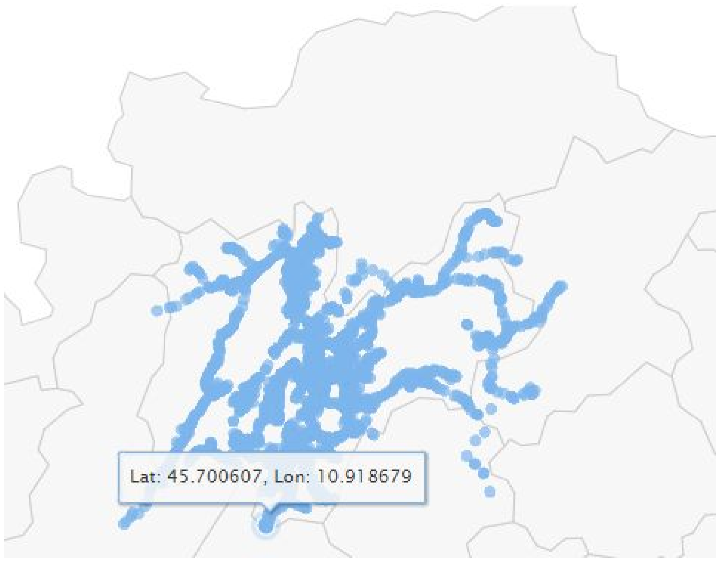
The generation of the "fuel.csv" dataset inside RapidMiner

Another cleaning and merging process, addressing service stations in Trentino, is shown in the picture above. A dataset from DatiOpen has been joined with another obtained from OpenStreetMap: this was possible because both resources contained the OpenStreetMap ID of service stations. After merging the datasets, the relevant attributes have been selected and some cleaning has been performed: this activity focused on the fuel brands of the service stations, which where specified in a heterogeneous way. For instance, all "Agip" entries have been replaced with "Eni", which is the current name of the brand. Finally, the attributes have been renamed and reordered, saving the processed dataset as a CSV file.



The generation of the "point_of_interest.csv" dataset inside RapidMiner

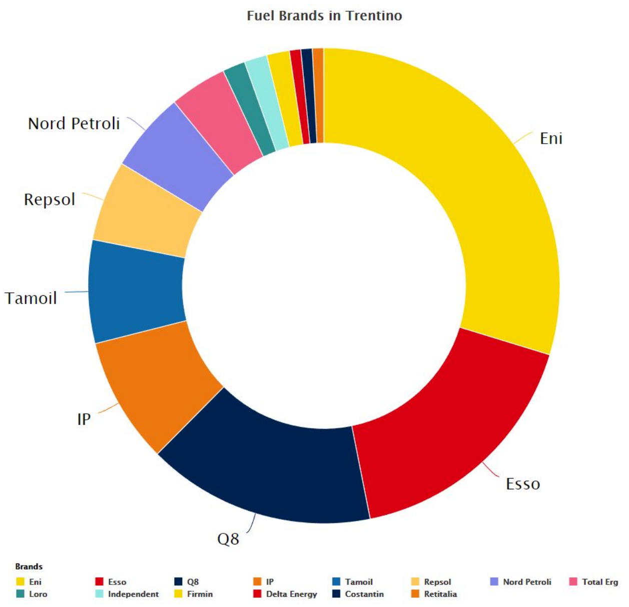
The image above shows the last example of a cleaning and joining process carried out with RapidMiner. In particular, it is shown how the data related to points of interest and historic places of Trentino has been extrapolated from *dati.trentino.it* and *europeandataportal.eu* and then analyzed. The two datasets have different attributes that are related to the same values, so once the most relevant attributes have been selected, they have been renamed in order to get a better definition, after which the two datasets containing similar information are merged and the process is saved in a CSV file.



Public transportation stops in Trentino seen in RapidMiner

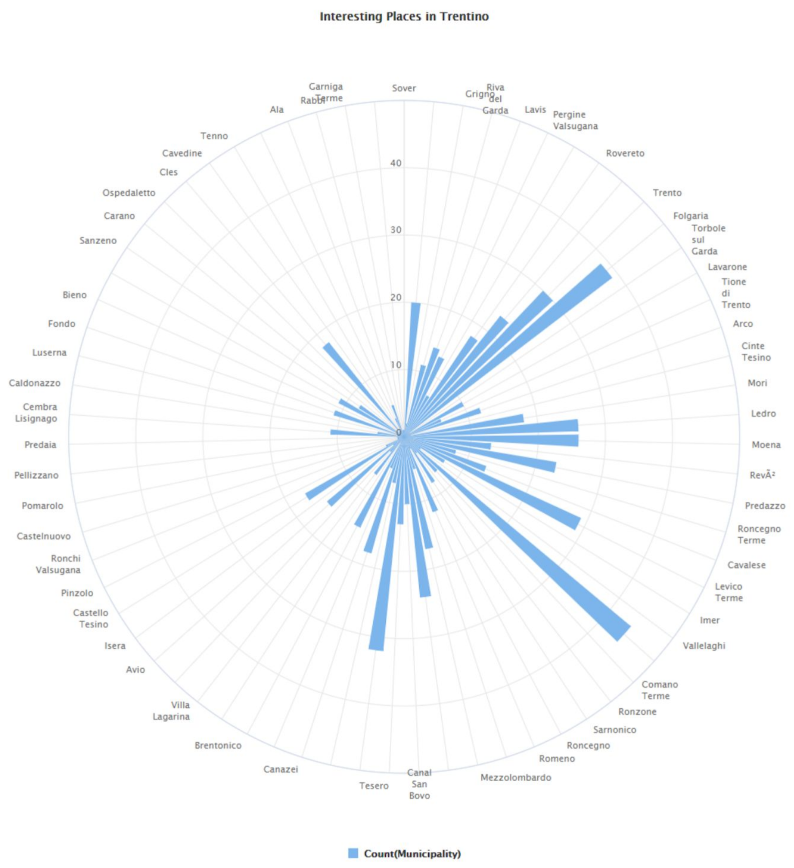
Using RapidMiner it is also possible to perform analysis in an immediate way. For example, the tool was used to verify whether the data regarding public transportation stops was correct. In particular, after the generation of the suburban "stops.csv", inside the "Visualization" tab of RapidMiner we analyzed the data by plotting them on a map. As it possible to see from the image below, the dataset is able to cover correctly all the public transportation stops of Trentino. Note that there are some points that are outside the region; this is due to the fact that some trips starts or end outside the region, and therefore these points are needed in order to maintain coherent trip representations.

Speaking about service stations, a possible analysis is the diffusion of the various brands in Trentino. Again, RapidMiner's Visualization feature has been employed for this purpose: the output of the analysis is a pie chart representing the fuel brands.



Fuel brands in Trentino

With respect to the places of interest and history of Trentino it is possible through RapidMiner's visualization features to see the number of interesting places that are in each locality of Trentino through the following representation:



Number of points of interest per city in Trentino

Section 4 – Related ontology proposal

In this section, several ontologies related to the transportation field are analyzed. We present, for each of them, a brief explanation of the main choices that the authors made in the generation of the ontology. In the last part of the section, we perform a comparison between the four presented approaches, analyzing strengths and weaknesses and identifying the one that appears to better suit our demands.

The first ontology that we present is defined in the paper "An Ontology for Transportation Systems"[12], written by Durgesh Nandini and Gautam Kishore Shahi at the University of Trento. The idea of this paper is to provide an ontology developed from the perspective of traveler users. The ontology also provides a simple vocabulary for describing a generic transportation system. The focus of this ontology

[12] "An Ontology for Transportation System.",Durgesh Nandini and Gautam Kishore Shahi 25 mag. 2019, available at https://easychair.org/publications/paper/8Fls.

is slightly different from ours, since in the development the authors assumed to have to integrate generic cites, from metropoles, such as London, to smaller cities like Trento. In our ontology, instead, the focus is more fine-grained, since we have "Trentino" as our only domain of interest. In the development, the ontology was aligned with the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), one of the top-level ontologies. In the ontology, two main classes are defined: "event" and "physicalObject". Inside the "NonAgentivePhysicalObject", a subcategory of physicalObject, three main classes are defined: "conveyance", "structure" and "physicalPlace". The *conveyance* class contains all the vehicles that can be used to move from one point to another. For example, "aircraft", "bus", "taxi", "motorcycle" and "train" are all subcategories of conveyance. The *structure* class contains all the structures that are related to transportation, such as airports, bus stations and ticket stores. The last main class, *physicalPlace*, contains all the classes that define particular physical areas, such as city and seaport. This ontology was clearly meant to be used in order to provide information on transportation systems. Our project is actually quite similar to the scenario provided in this paper: indeed one of our four personas, John, is a traveler. Our ontology appears to cover a broader area than the one just described: this is due to the fact that our competency queries encompasse three other personas, while the presented paper just focuses on the needs of travelers.

The second resource that we decided to analyze is "TRANSIT: A vocabulary for describing transit systems and routes"[13]. TRANSIT is a vocabulary that describes transit systems, routes, stops and schedules based on the "General Transit Feed Specification" published by Google. The GTFS is a format used for public transportation schedules and associated geographic information. We decided to analyze this vocabulary because some of the datasets handled in the previous sections were based on a GTFS format. The transit vocabulary is based on 6 core classes. "Route" defines a public transportation route, such as the one performed by a bus. Each route has an "Agency", which is the organization responsible for operating services on transit routes. Each route is served by a "Service", which is a public transportation service that operates a route on a given schedule. For example, one parameter of this class is "headsign", which identifies the name of the service. Each service has a "Schedule", which is a list of dates on which a particular service operates, and it is identified by a date. Each Service has also several "ServiceStop", which identify a stop that has been included in the specific service. This class defines the arrival time, the departure time and the sequence of following stops. Each ServiceStop is associated to a "Stop", which defines the actual location where passengers board or disembark from a transit vehicle. It is clear, from what has been described, that this vocabulary is one of the main resources that has to be used when handling public transportation schedule. For example, during our work we saw that Trentino Trasporti handles its data, both for the suburban and urban, using the GTFS format. It would then be straightforward to work on this data using the TRANSIT vocabulary. Unfortunately, this resource just describes public transportation, but has no details for other types of transportation, such as car. For this reason, it appears that

[13] "TRANSIT: A vocabulary for describing transit systems and routes." 28 mar. 2011, http://vocab.org/transit/.

this vocabulary is not well suited for our application, since our work has to cover all types of transportation in Trentino, and not just the public ones.

Another ontology considered is the one described in the paper "A public transportation ontology to support user travel planning[14]" written by Mnasser Houda, Maha Khemaja (University of Sousse) and Kathia Oliveira, Mourad Abed (Univ Lille Nord de France). The ontology presented concentrates on obtaining the necessary information that a person needs to identify the best path in order to reach a destination B starting from point A, the domain considered is that of transportation considering a multimodality of transport and interests of passengers. The ontology has been defined in four phases: "purpose" to facilitate the request of information in the planning of a vacation in a city or in a nation, so that the person can choose which transport and itinerary is more suitable for personal requests; "conceptualization" definition of concepts, relationships and constraints and a glossary that contains all the specific attributes and concepts; "formalization" expressing the ontology in specific tools, has been formalized in OWL 1.0 and Protégé; as the last "validation" to instantiate the ontology in real contexts, namely Paris and its surroundings. In conclusion, this paper presents the definition of the process of an ontology for public transportation starting from the definition of the concepts up to the specification and formalization of the axioms inherent to the domain. In relation to the domain of our interest in this project with this ontology a good part of aspects are omitted, that is, there is a lack of different types of transport: charsharing; lack of information between public and car connections, and lastly parking, service stations and eco-friendly solutions are not considered.

Finally, we have considered the km4City[15] ontology developed by Pierfrancesco Bellini, Monica Benigni, Riccardo Billero, Paolo Nesi and Nadia Rauch from the Department of Information Engineering of the University of Florence. The main purpose of this ontology is the reconciliation of public data in smart cities, such as roads, transportation and points of interest. The structure is organized into several macro-classes: *Administration*, *Street-guide*, *Point of Interest*, *Local Public Transport*, *Sensors*, *Temporal* and *Metadata*. For our project, the most relevant areas are Local Public Transport (information about buses and trains) and Sensors (monitoring of parking and traffic), but Street-Guide is also important to represent the road system. One of the main advantages of the km4City ontology is the reuse of popular vocabularies and upper ontologies, such as Ontology of Transportation Networks (OTN) or WGS84: the reference to well-known resources can improve the interoperability. The ontology has been built from data about Florence and the region of Tuscany, therefore the scale is similar to the one of our project. Another point of strength is the consideration of both public and private transport, which was lacking in the previous ontologies. A possible drawback is related to precision, as the km4City ontology doesn't only cover transportation, but also other domains related to cities.

[14] "A public transportation ontology to support user travel planning.", Mnasser Houda ; Maha Khemaja ; Kathia Oliveira ; Mourad Abed, available at https://ieeexplore.ieee.org/document/5507372.

[15] "Km4City App." https://www.km4city.org/.

After analyzing the ontologies, we have ranked them from best to worst according to the domain of transportation in Trentino:

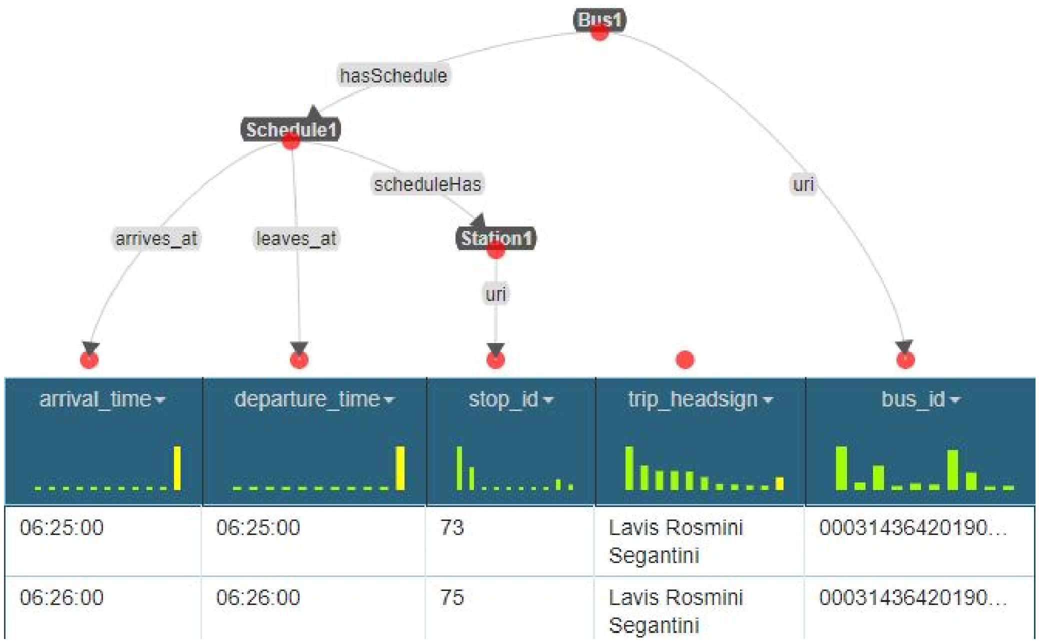
1. "km4City" ontology: both public and private transportation are addressed and there are references to well-known resources (OTN, WGS84).
2. "An Ontology for Transportation Systems": the coverage is good and a top-level ontology has been reused.
3. "TRANSIT: A vocabulary for describing transit systems and routes": a very popular standard, but limited to public transportation.
4. "A public transportation ontology to support user travel planning": many important concepts of transportation are lacking.

Section 5 – Integration process description

Data integration is one of the most important and complex tasks of the methodology, as it merges the formal model with the collected data. This is an iterative process, because inconsistencies between the model and the data are very likely to arise and the ontology must be progressively updated to correct these issues.

We have used Karma[16] to perform data integration: it requires to load the ontology and the datasets that must be integrated. Each attribute of a dataset must be mapped to the ontology by selecting the correct attribute or relation of the entity type; additionally, new columns may be generated for the dataset, for instance specifying the URI of each example. For each dataset, a "model" was produced. A model can be considered as a description of how the dataset columns are linked to the ontology. Using this approach, each member of the group defined the models of a subset of the datasets. Finally, once all the models were defined, we imported all the datasets in a single context using the models defined previously. Finally, using the tool "OpenRDF" inside Karma, the final RDF was generated, using as format "XML/RDF". The final output is published at the following link. An RDF/XML version is also provided here.

[16] "Karma: A Data Integration Tool - Center on Knowledge Graphs." https://usc-isi-i2.github.io/karma/.



arrival_time	departure_time	stop_id	trip_headsign	bus_id
06:25:00	06:25:00	73	Lavis Rosmini Segantini	00031436420190...
06:26:00	06:26:00	75	Lavis Rosmini Segantini	00031436420190...

Integration of the urban_schedule.csv dataset

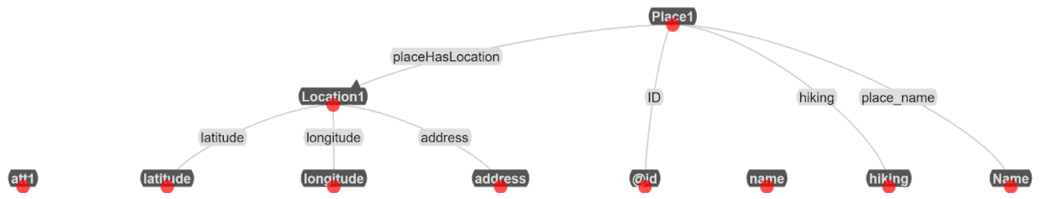
The image above shows the integration done in Karma for the "urban_schedule.csv" dataset. For each mean of transportation, three different datasets were provided: "stops", which contained all the possible stations, "trip", containing all possible trips and "schedule", containing the arrival and departure time of each trip for every station. Clearly, the schedule has a relation both on the trip and on the stops datasets. In particular, each schedule was defined by the tuple <stop_id, transportation_id, arrival_time, departure_time>. In order to represent this relation, we defined stop_id and transportation_id as URI of the entity. In this way, all the three entities were linked together. Some columns, such as "trip_headsign" were not represented inside the ontology and therefore they were also ignored during the integration process.

During the integration we noticed that the "Location" entity was defined by three attributes: latitude, longitude and address. This represented a problem since most of our data contained only the spatial location, but no information about the actual address of the position. In order to address this problem, rather than asking to remove the property, we decided to create a Python script which, given latitude and longitude, returned the address of that point. The script can be found in the GitHub repository, inside the Scripts folder, with the name findAddress.py. This scripts uses "geopy" a library of Python which allows you to query an external API to gather spatial data. This script receives as input parameter the location of a CSV file and, for each row, it adds information about the address. The main problem of this script is that, since it uses an external API, it has to submit to some constraints, such as the fact of performing one query per second. This represented a issue for the streets dataset, which is formed by ~200000 entries. This means that inserting the address for the roads would have required ~55 hours of computation. We decided therefore to add the address parameter to all our spatial datasets except for the roads.

During the integration of our datasets with the formal model, we have found some minor inconsistencies that had to be corrected by updating the ontology. A possible issue was the absence of some attributes in the formal model: they had to be added in order to support the integration of specific datasets. We also encountered the opposite situation, with attributes of the formal model that did not appear in the collected data: we decided to keep this additional information, as it may be useful for future developments where more complete data is provided. Sometimes, entire entity types had to be updated to better reflect the data.

A concrete example of the integration process is related to car sharing. Originally, the formal model had a “Shared_Car” entity type that had attributes “path”, “destination” and was connected to the classes “Car”, “Location” via specific relationships. Our data, however, described car sharing by listing the car sharing stations in Trentino: we had to remove “Shared_Car” and replace it with “Shared_car_station” in order to integrate the dataset. This new class has attributes “name_station” and “numberCars”, but also a relationship towards “Location”.

As described above, the data obtained in the previous phases was found to be missing or unnecessary in some cases, so there was a need to regenerate new data that reflected what was required of the formal model. Given the premise, a second example of the integration process with KARMA is described from the entity "Place" of the formal model: this entity is connected by relationships with two other entities "Location" and "Itinerary", while the attributes it possesses refer to *time*, *ID*, *placeName*, *pricePlace* and *hiking*. On the basis of the available dataset, it was possible to evaluate how some relationships between Places and Itinerary entities could be superfluous, as Location can be considered as an entity that manages the itinerary because it needs its information to be able to obtain it: in this way it can be defined that this relationship is not considered necessary for the modeling of these entities, thus maintaining the unique relationship "placeHasLocation" between Places and Locations allowing identification of the itinerary. Starting from these considerations the dataset was modeled according to the parameters provided by KARMA to obtain the related RDF and R2RML files. The final model obtained for the integration of Places is shown below, where it is possible to see that some previously declared attributes are not mapped because it was not possible to obtain them from the actual data, therefore they were considered useful for future integration.

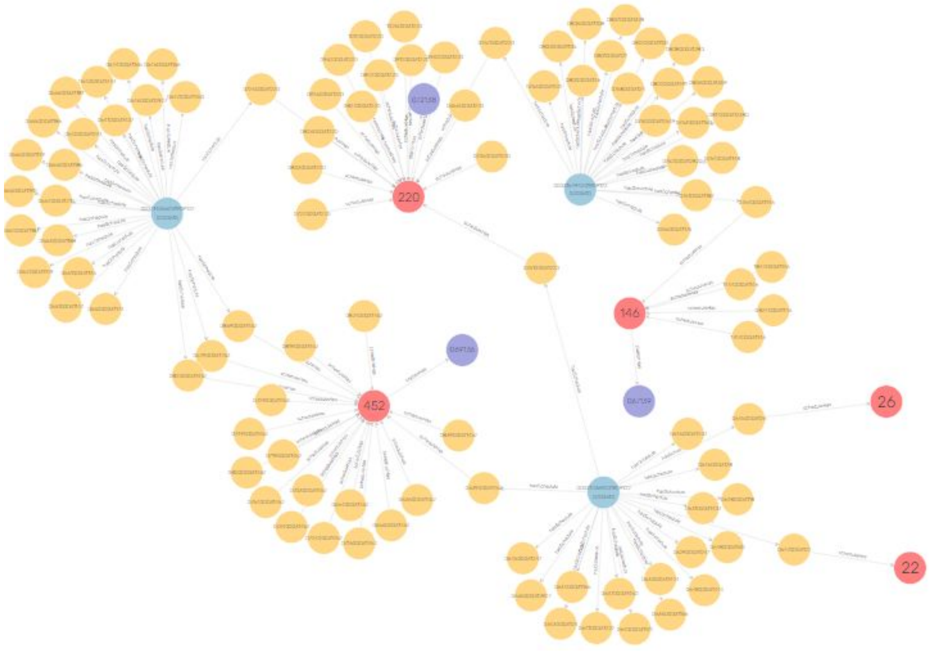


Integration of the hiking.csv dataset

Section 6 – Knowledge Graph description

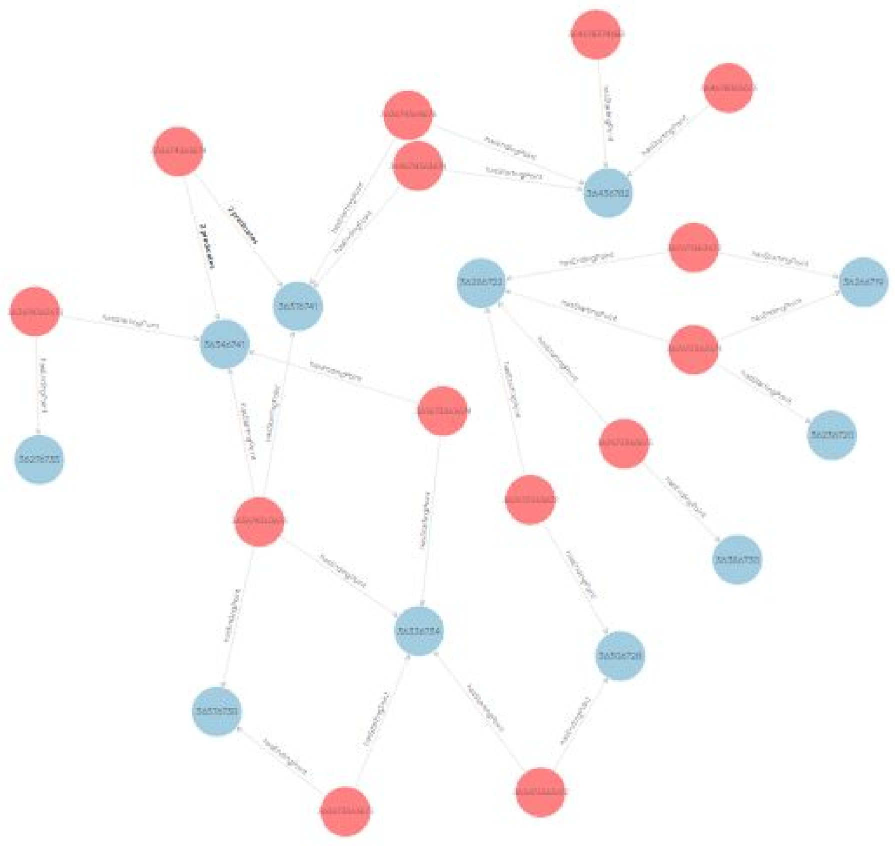
In this final part of the report we analyze the output produced using Karma. In particular, after the final definition of the complete RDF, two main assessments have been done in order to evaluate the quality of the result. First of all, a subset of the knowledge graph has been analyzed using GraphDB[17]. Since the tool only allows to import graphs smaller than 200MB, we had to import just sub-sections of our output.

In the image below is presented one example of visualization of some entities of our Knowledge Graph. The blue entities are some among the urban buses of Trento: the entity with id "0003143662019091220200610" is bus number 5, which allows students to come to the university of Povo each day, while entity with id "0003143692019091220200610" is bus number 9, which goes to Cognola. The yellow entities are the schedules related to the bus. Each schedule defines the time the bus is expected to arrive to a particular station. The red entities are the stations (for example entity with id "452" is the station of "Venezia - Corallo". As it is possible to see, this station has connections to schedules both for bus number 5 and number 9, since in that station both buses stop. This graph represents an important feature of our integration: we had as inputs of the integration three different CSV datasets (one for the buses, one for the stations and one for the schedule), but inside our knowledge graph this data is all integrated together.



Some urban buses of Trento represented in the Knowledge Graph

[17] "GraphDB" http://graphdb.ontotext.com/



A subset of the streets of Trentino

Another important aspect of our model is street navigation: this task requires the use of a graph, where vertices are locations and edges represent streets that connect locations. The image above depicts a part of the knowledge graph related to the area of Val di Sole, where blue nodes represent start/finish locations and red nodes are streets between locations. For instance, the blue node of entity 36286722 is a location in Pejo with coordinates (46.3628816,10.6722171): it is connected to several red nodes representing streets, and therefore it represents an intersection. This is another example of useful information that we can obtain from the combination of different concepts in the knowledge graph.

After the visualization we decided to perform an additional analysis in order to check whether the generated knowledge graph is able to solve the competency questions defined at the beginning of the project. In order to do so, we selected a subset of significative competency questions and we tried to solve them by extracting data from our knowledge graph using SPARQL. This part has been done inside the OpenRDF tool of Karma. Clearly some competency questions cannot be solved just by performing a simple query. For example, in order to respond to the competency question “Can I find the fastest path to go from point A to point B” we should adopt an algorithm that, given the streets, is able to compute the fastest path. Our knowledge graph is able to return the streets of Trentino, along with the type of road (primary, secondary, etc.) and some extra information (if it is a one-way street). This data represents all the inputs the algorithm requires in order to solve the problem. Below, we present two among the SPARQL queries that we defined in order to solve the competency questions.

```
PREFIX ontology:
<http://www.semanticweb.org/andreamontibeller/ontologies/Tran
sportation#>
SELECT ?arrival_time ?name ?latitude ?longitude
WHERE {
?Bus ontology:PublTransportation_name ?"Piazza Dante P.Fiera
Povo Oltrecastello"@it.
?Bus ontology:hasSchedule ?Schedule.
?Schedule ontology:arrives_at ?arrival_time.
?Schedule ontology:scheduleHas ?Station.
?Station ontology:name ?name.
?Station ontology:stationHas ?Location.
?Location ontology:latitude ?latitude.
?Location ontology:longitude ?longitude.
}
ORDER BY ASC(?arrival_time)
```

Above it is presented the SPARQL query used to solve the competency questions "I want to know the timeline of bus number 5" and "I want to know the location of a bus stop". In particular, this query returns the schedule of bus number 5 along with the name of the stops and their latitude and longitude. Without a knowledge graph these competency questions would have required computations on three different datasets, while with our integration we are able to respond to the two questions using just one query.

```
PREFIX ontology:
<http://www.semanticweb.org/andreamontibeller/ontologies/Tran
sportation#>
SELECT ?name ?latitude ?longitude
WHERE {
?Shared_Bike_Station ontology:sharedBikeStation_name ?name.
?Shared_Bike_Station ontology:bikeStationHas ?Location.
?Location ontology:longitude ?longitude.
?Location ontology:latitude ?latitude.
}
```

The query above responds to one of Marco's competency questions: "How far away is the nearest bike sharing station from my final destination?". The result of this SPARQL query are the name, longitude and latitude of each bike sharing station: then, given the current position of Marco, the distance from the station can be computed.

Section 7 – Final Consideration and Open Issues

Once we verified the performance of our knowledge graph, we reflected on the final result, analysing any issues that are present in the final output and that have yet to be solved. We also reflected on what tasks should be addressed in the next iteration of the project.

An issue that we encountered was the lack of open data about transportation in Trentino. Trentino Trasporti provided us one of the main sources, since all data was publicly available both on their website and on the OpenData Trentino repository. We had instead a lot of issues retrieving data about trains: Trenitalia does not provide any type of data, therefore we had to scrape them from their own mobile application API. Clearly, this type of data is not qualitatively as good as the one produced by Trentino Trasporti: for example, inside the data retrieved from the "ViaggiaTreno" API we were not able to extract the data about the train station of Povo-Mesiano. Moreover, in the analysis of trains we had not the possibility to consider private train companies (such as "Italo") since not only they do not provide any open source data, but we were also not able to scrape them from external resources. This lack of open data is present also for other means of transportation. In Trentino there are ways to move that, even if not conventional, should be represented inside our knowledge graph. For example, on Garda Lake there is the possibility to move using boats, while in Trento there is the small airport "G. Caproni". These types of transportation are isolated to small centers and are not used as a general mean of transportation in all Trentino. We tried anyway to search for data both for boats and airplanes in Trentino, but we were not able to find any. For this reason, currently our knowledge graph does not cover entirely the possible ways of transportation in Trentino. Another issue that is currently present in the final output is that not every location has an "address" attribute. As said in the previous sections, this problem has not to do with lack of data, but is instead a problem of computational time. Since the computation of the address for each location would have required more than two days of processing we decided to avoid inserting this attribute for the streets. This leaves the knowledge graph a bit incomplete, since the address name of the streets is unknown. With more time on our hand we may have filled this information as well.

Finally, we reflected on how we could improve the knowledge graph in a next iteration of work. We think the output could be improved in two ways. The first one is to improve the quality of the data that is already present. As seen, there are some missing attributes (such as the address) that we could fill, improving in this way the overall quality of the graph. The second way to improve our output is to introduce new, more specialized, ways of transportation. As seen, for some means of transportation (such as private trains) it is currently difficult to extract any information. We may therefore decide to introduce in our graph more specific ways of transportation. For example, it may be interesting to extend our knowledge graph also considering data about taxis and data about services such as "BlaBlaCar" and "FlixBus".