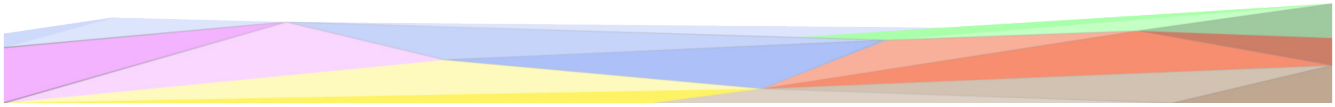
Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

Transportation KG [KDI 2020-21]

Document Data:	Reference Persons:
- date -	- authors -



Contents

Revision History:

Revision	Date	Author	Description of Changes
1.0	17.10.2020	Fivos Kapidis, Antonio Stefani	Draft of the Scope Definition
2.0	18.10.2020	Fivos Kapidis	Draft of the Scenario and of the Storytelling Definition
3.0	19.10.2020	Fivos Kapidis	Draft of the CQs
4.0	19.10.2020	Fivos Kapidis, Antonio Stefani	Draft of the Inception Schema
5.0	19.10.2020	Omid Jadidi	Draft of the dataset description
3.1	20.10.2020	Fivos Kapidis, Antonio Stefani	Final revision of the CQs
6.0	20.10.2020	Alberto Carbognin	Draft of the metadata documentation
2.1	21.10.2020	Antonio Stefani	Final revision of the Scenario Description and of the Storytelling Definition
3.2	21.10.2020	Antonio Stefani	Final revision of the CQs
4.1	21.10.2020	Antonio Stefani	Final revision of the Inception Schema
6.1	21.10.2020	Alberto Carbognin	Final revision of the metadata documentation
5.1	21.10.2020	Omid Jadidi	Final revision of the dataset description
7.0	03.11.2020	Antonio Stefani	Draft Informal Modeling Schema
8.0	03.11.2020	Fivos Kapidis	Draft variance respect the defined CQs
9.0	04.11.2020	Antonio Stefani	Draft ETypes and attributes
10.0	04.11.2020	Omid Jadidi	Draft metadata documentation
11.0	04.11.2020	Alberto Carbognin	Draft datasets management process
7.1	05.11.2020	Antonio Stefani, Fivos Kapidis	Final revision of ETypes and attributes, Informal Modeling schema
10.1	05.11.2020	Omid Jadidi, Alberto Carbognin	Final revision datasets management process
11.1	05.11.2020	Omid Jadidi, Alberto Carbognin	Final revision metadata documentation
8.1	05.11.2020	Antonio Stefani, Fivos Kapidis	Final revision of the variance respect the defined CQs
12.0	18.11.2020	Antonio Stefani	Draft Ontology documentation
13.0	25.11.2020	Omid Jadid	Draft datasets management
14.0	25.11.2020	Alberto Carbognin	Draft variance respect previous datasets
12.1	26.11.2020	Antonio Stefani, Fivos Kapidis	Final revision Ontology documentation
13.1	26.11.2020	Omid Jadidi, Alberto Carbognin	Final revision datasets management
14.1	26.11.2020	Omid Jadidi, Alberto Carbognin	Final revision variance respect previous datasets
15.0	15.11.2020	Antonio Stefani	Revision of the whole documents
16.0	16.12.2020	Fivos Kapidis	Draft evaluation
17.0	16.12.2020	Antonio Stefani	Draft codebook
18.0	16.12.2020	Omid Jadidi	Draft data processing
19.0	16.12.2020	Alberto Carbognin	Draft karmalinker and process documentation
16.1	17.12.2020	Antonio Stefani	Final revision evaluation
17.1	17.12.2020	Antonio Stefani	Final Revision codebook, draft presentations
19.1	17.12.2020	Omid Jadidi, Alberto Carbognin	Final revision karmalinker and process documentation
18.1	17.12.2020	Omid Jadidi, Alberto Carbognin	Final revision data processing
20.0	18.12.2020	All	Final revision presentation and documents

1 Knowledge Graph Codebook

In the first chapter of this report we are going to resume all the activities done during the project. Each step will be then illustrated into details in the second chapter.

1.1 Knowledge Graph general description

Improving the quality of traveling is essential to live in an easier way: you always lose time moving from one place to another and you lose even much more time looking for a route or mode of transport to get somewhere as fast as possible and often spending as little as possible. Our project wants to solve this problem: it works under the transportation domain and it goals is to answer to those questions regarding the road system in Trentino, this include of course the railway system, the mountain paths and the cycling routes as well. So, more in general, we want to provide a system able to suggest to the users the fastest and/or (or whichever information they want to know) cheapest way to get to another place by respecting their will: using a specific public mode of transport or a private one.

1.2 Data level

In the final version of the datalevel datasets have been broken in many pieces to be as much as possible the same as schema. For instance one dataset might have many fields and attributes in it, so we devided these attributes in classes that have been described in the schema **(Address, Location, ...)** and defined a key for each connection between classes. In this sense hierarchy of the classes or etypes have been saved through key indexes. Later we will explane how we import these data in karma linker and define relation between nodes. After importing data in karmalinker and defining relations we extract the final data as rdf format and all have been saved in github directory.

1.2.1 Datasets general details

- **Trentino public transport Urban TTE**: this dataset contains the *core data* about the city of Trento;
- **Trentino public transport Extra-Urban TTE**: this dataset contains the *core data* of routes in the Trentino province;
- **Trentino public transport rates for Urban TTE**: contains the *core data* of prices of routes in Trento;
- **Trentino public transport rates for Extra-Urban TTE**: contains the *core data* of prices of routes in Trentino Province;
- **Cycle paths**: contains the *common data* of cycle paths routes;
- **Cycling points of interest**: contains the *common data* of cycle point of interest;
- **Italian Parking Areas**: contains the *contextual data* of the parking places;
- **Railway Stations**: contains the *common data* of the railway's stations;
- **Car sharing (Open Data)**: contains the *common data* of the Car sharing information.;
- **Map of petrol stations in Italy**: contains the *contextual data* of the gas stations around Italy;

- **Mountain Paths**: contains the *common data* of the mountain paths;
- **Campsite and other accommodation facilities**: contains the *contextual data* of the campsites and their relative prices.

1. Trentino public transport Urban TTE.
   This dataset is beased on the GTFS standard and have eleven different attributes (agency, calendar, calendar_dates, feed_info, routes, shapes, stop_times, stops, stoplevel, transfers, trips). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	**Description**
Dataset Identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Dataset Publisher	Name: Public Transport Service, IPAVAT Code: 0OK0PZ
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Holder	Autonomous Province of Trento
Author	Name: Public Transport Service, IPA VAT: 0OK0PZ
Url	https://www.trentinotrasporti.it/opendata/google_transit_urbano_tte.zip

2. Trentino public transport Extra-Urban TTE.
   This dataset is beased on the GTFS standard and have eleven different attributes (agency, calendar, calendar_dates, feed_info, routes, shapes, stop_times, stops, stoplevel, transfers, trips). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	**Description**
Dataset Identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Dataset Publisher	Name: Public Transport Service, IPA / VAT Code: 0OK0PZ
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Languages of the dataset	Italian
Holder	Autonomous Province of Trento
Author	Name: Public Transport Service, IPA / VAT: 0OK0PZ
Url	https://www.trentinotrasporti.it/opendata/google_transit_extraurbano_tte.zip

3. Trentino public transport rates for Urban TTE.
   This dataset is beased on the GTFS standard and have seven different attributes (fare_attributes_urbano.txt, fare_attributes_urbano_cartascalare, fare_attributes_urbano_mobile, fare_rules_urbano, fare_rules_urbano_cartascalare, fare_rules_urbano_mobile, zones_urbano). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	Description
Dataset identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Dataset Publisher	Name: Public Transport Service, IPA / VAT Code: 0OK0PZ
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Holder	Autonomous Province of Trento
Author	Name: Public Transport Service, IPA / VAT: 0OK0PZ
Url	https://dati.trentino.it/dataset/6d5c2000-972e-4c21-aef6-fdbba94418a8/resource/44efc0bd-223a-49c7-b3b0-16128e32813c/download/tariffegtfsurbano.zip

4. Trentino public transport rates for Extra-Urban TTE.
   This dataset is beased on the GTFS standard and have seven different attributes (fare_attributes_urbano.txt, fare_attributes_urbano_cartascalare, fare_attributes_urbano_mobile, fare_rules_urbano, fare_rules_urbano_cartascalare, fare_rules_urbano_mobile, zones_urbano). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	Description
Identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Publisher	Name: Public Transport Service, IPA / VAT Code: 0OK0PZ
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Holder	Autonomous Province of Trento
Author	Name: Public Transport Service, IPA / VAT: 0OK0PZ
Url	https://dati.trentino.it/dataset/6d5c2000-972e-4c21-aef6-fdbba94418a8/resource/10e93dd1-3463-4664-8c24-300a7403780a/download/tariffegtfsextraurbano.zip

5. Piste ciclabili (Cycle paths).
   General description of this dataset is in table below:

Info	Description
Identifiers	p_TN:3ff5db13-8a3d-4bd8-8d6f-9b2fdf1aeb41_resource
Url	https://siat.provincia.tn.it/IDT/vector/public/p_tn_3ff5db13-8a3d-4bd8-8d6f-9b2fdf1aeb41.zip
Created	26.09.2008
Coordinates	[[[10.41, 46.6], [11.97, 46.6], [11.97, 45.6], [10.41, 45.6], [10.41, 46.6]]] Type: Polygon

6. Punti di interesse ciclabili (Cycling points of interest).

Representation of the punctual elements present on the Trentino cycle paths: bicigrill (refreshment point, assistance and information), counters (instrumentation for detecting pedestrian and cycle paths), cippi km and fountains. General description of this dataset is in table below:

Info	Description
Identifiers	p_TN:0211d261-70d8-485e-9265-b1c27b1a84e1_resource
Url	https://siat.provincia.tn.it/IDT/vector/public/p_tn_0211d261-70d8-485e-9265-b1c27b1a84e1.zip
Coordinates	[[[10.41, 45.6], [10.41, 46.6], [11.97, 46.6], [11.97, 45.6], [10.41, 45.6]]] Tipo: Polygon
Contact	mailto: serv.naturambiente@provincia.tn.it

7. Parking map in Italy.
   Archive, which can be represented on the map, which contains the non-exhaustive list of over 21,000 car parks in Italy. The source of the data is OpenStreetMap.org which has been assigned with automated procedures the classification by municipality, province and region. Data updated on: 23 February 2016 The data was created by DatiOpen.it on 23 February 2016

Info	Description
Publisher	Open.it
Author	OpenStreetMap http://www.datiopen.it/it/catalogo-opendata/openstreetmap-org
Url	http://www.datiopen.it/it/opendata/Mappa_dei_parcheggi_in_Italia
Contact	mailto: info@datiopen.it

8. Train stations (Open data).
   Station of the railway stations in the municipal area of Trento. It includes the Brenner railway, the Trento_Malè_Marilleva railway and the Valsugana railway. Data provided by Trentino Trasporti.

Info	Description
Publisher	Trentino Trasporti
Author	Trentino Trasporti https://www.trentinotrasporti.it/
Url	https://www.comune.trento.it/Aree-tematiche/Cartografia/Download/Stazioni-treno
Contact	mailto: Servizio.innovazionedigitale@comune.trento.it

9. Car sharing (Open Data).
   Location of Car sharing stalls Parking spaces dedicated to the collection and delivery of Car sharing vehicles. Data taken directly from the site https://www.carsharing.tn.it Car sharing allows you to have a car suitable for family or business needs without owning one and without incurring fixed costs (road tax, insurance, maintenance, garage or parking), but paying only in proportion to use.

Info	Description
Author	Name: Municipality of Trento IPA / VAT: c_l378
Published	Trentino Trasporti
Url	https://dati.trentino.it/dataset/car-sharing-open-data
Contact	mailto: Servizio.innovazionedigitale@comune.trento.it

10. Map of petrol stations in Italy.
    Archive, which can be represented on the map, which contains the non-exhaustive list of over 13,000 petrol stations in Italy. The source of the data is OpenStreetMap.org which has been assigned with automated procedures the classification by municipality, province and region.

Info	Description
Author	OpenStreetMap
Publisher	DatiOpen.it
Url	http://www.datiopen.it/it/opendata/Mappa_dei_distributori_di_carburante_in_italia
Contact	mailto: Servizio.innovazionedigitale@comune.trento.it

11. Province of Trento Paths.
    Paths of the entire network of the Società degli Alpinisti Tridentini (SAT) that insist on the territory of the Autonomous Province of Trento: each path consists of the spatial coordinates that allow it to be correctly positioned on the territory and presents a series of additional textual information (attributes) that describe.

Info	Description
Author	SAT (Society of Tridentine Alpinists)
Published	DatiOpen.it
Url	http://www.datiopen.it/it/opendata/Provincia_di_Trento_Sentieri
Contact	mailto: sentieri@sat.tn.it

12. Autonomous Province of Trento List of non-hotel structures.
    The archive contains information on non-hotel accommodation facilities in the territory of the Autonomous Province of Trento: rural businesses, bed-and-breakfasts, campsites, hostels, holiday homes, etc. Where available, the data contains the address, telephone, e-mail address, website and other information.

Info	Description
Author	Trentino Alto Adige Region
Published	DatiOpen.it
Url	http://www.datiopen.it/it/opendata/Provincia_Autonoma_di_Trento_Elenco_strutture_extra_alberghiere?metadati=showall
Contact	mailto: info@open.it

1.2.2 Datasets metadata documentation

All the metadeta at attribute level can be found in the json format in our Github repository.

1.3 Ontology level

In this section we are going to describe the final ontology defining our Knowledge Graph illustrating in particular all parties involved.

1.3.1 Ontology general details

As previously said in the introduction, our ontology aims to model the transportation domain. We have built an ontology as modular as possible taking into consideration not only the road system but also the railway system, the cycling routes and the mountain paths going to cover almost all the ways and modes of transport. This way of proceeding gave us the chance to not only providing an ontology working with different datasets from those of departure but also one which can be easily integrated into other ontologies describing different problems: for instance a Geo-Spatial ontology can integrate our one just by adding links to the class "Location" (i.e. the coordinates of a point).
Our work was done starting from a blank sheet of paper in order to develop a schema as modular and complete as possible. This leaded us to not consider any ontology already existing in our domain but just hiring some constructs of them. In particular one thing we focused on was the GTFS Format structure, indeed we used connection among the classes guided by the format itself (Calendar ⇒ CalendarDates).

1.3.2 Ontology metadata documentation

In this section all the metadata describing each element of the ontology are reported, in particular we illustrate firstly the ETypes used and the concept which they represent, then we show all the attributes of a circumstantial class added to define the type of certain elements (i.e. the class Enumeration comprising the several lists of the modes of transport, the facilities and the fuel types used by vehicles)

EType	GID	Concept	Data Properties	Object Properties
Address	36400	A sign in front of a house or business carrying the conventional form by which its location is described	has City (45988) has Number (34489) has Province (46567) has Street Address (45807) has Zip Code (34110)	has Location (132)
Agency	45084	An administrative unit of government	has Agency ID (120057) has Agency Language (120060) has Agency Name (120058) has Agency Timezone (120059)	has Contact (39136)

EType	GID	Concept	Data Properties	Object Properties
Calendar	44719	A tabular array of the days	has Start Date (120091) has End Date (120090) has Monday (80758) has Tuesday (80759) has Wednesday (80760) has Thursday (80761) has Friday (80762) has Saturday (80763) has Sunday (80757) has Service ID (120056)	has Exception (31741)
Calendar Dates	120045	Tabular array of dates which are associated to a specific events	has Date (103420) has Service ID (120056) has Exception (31741)	
Contact	120109	All useful information to get in touch with someone	has Phone (34485) has Website (34126) has Email (105296)	
Duration	80581	The property of enduring or continuing in time		has Time Stamp (120046)
Enumeration	34789	A numbered list		
Facility	3012	Something designed and created to serve a particular function and to afford a particular convenience or service		Is Specified By (120047) has Calendar (44719) has Facility Address (36400) has Facility Calendar (44719) has Facility Contact (39136) has Facility Price (28431)
Location	132	A point or extent in space	has Altitude (28272) has Latitude (46263) has Longitude (46270)	
Mode Of Transport	120044	Way in which transportation happens		Is Specified By (120048)
Path	46379	An established line of travel or access		has Road (22592)
Price	28431	Value measured by what must be given or done or undergone to obtain something	has Cost (70407) has Currency Type (120061)	

EType	GID	Concept	Data Properties	Object Properties
Private Transport	120043	Personal or individual use of transportation vehicle	has Cost (70407) has Currency Type (120061)	has Individual Price (28431) has Fuel Type (120049)
Public Transport	22138	Conveyance for passengers or mail or freight		has Agency (45084) has Stop Time (120042) has Ticket (111874)
Road	22592	An open way (generally public) for travel or transportation	has Length (28259) has Road ID (120051) has Speed Limit (35726)	has Address (36400) has Duration (80581) has Facility (3012) has Mode Of Transport (120044)
Stop	5446	A brief stay in the course of a journey	has Stop Code (120053) has Stop ID (120051) has Stop Name (120054) has Wheel Chair Boarding (120055)	has Location (132)
Stop Time	120042	Temporal parameters for a specific stop	has Cost (70407) has Currency Type (120061)	has Calendar (44719) has Stop Time (5446) has Time Stamp (120046)
Ticket	111874	Provide with a ticket for passage or admission	has Fare ID (70599) has Payment Method (120061) has Time Table Duration (120088)	has Price (28431)
Time Stamp	120046	Hour, minutes and seconds of a duration	has Hour (81114) has Minutes (81154) has Seconds (72173)	

Table 13: Ontology Elements Metadata

The class Enumeration is a class needed to collect all those lists of elements which can be selected to specify a particular attribute of a class. In particular we have designed three different lists:

Parent EType	Element	GID	Concept
ModeOftransport_GID-120044			
	Bicycle	15188	A wheeled vehicle that has two wheels and is moved by foot pedals
	Bus	15732	A vehicle carrying many passengers

Parent EType	Element	GID	Concept
	CableCar	15797	A conveyance for passengers or freight on a cable railway
	Car	15945	A wheeled vehicle adapted to the rails of railroad
	Foot	1429	The act of traveling by foot
	Train	18679	Wheelwork consisting of a connected set of rotating gears by which force is transmitted or motion or torque is changed
Facility_GID-3012			
	BikeSharing	843	The act of maneuvering a vehicle into a location where it can be left temporarily
	BusStation	15745	A terminal that serves bus passengers
	CampsiteParking	45940	A site where people on holiday can pitch a tent
	FuelStation	18641	A service station that sells gasoline
	ParkingArea	46375	A lot where cars are parked
	RailwayStation	22321	Terminal where trains load or unload passengers or goods
PrivateTransport_GID-120043			
	Diesel	17309	An internal combustion engine that burns heavy oil
	Electric	61771	A physical phenomenon associated with stationary or moving electrons and protons
	Gas	79121	A fluid in the gaseous state having neither independent shape nor volume and being able to expand indefinitely
	Methane	79566	A colorless odorless gas used as a fuel
	Petrol	78042	A volatile flammable mixture of hydrocarbons (hexane and heptane and octane etc.) derived from petroleum

Table 14: Enumeration Metadata

1.4 Knowledge Graph Evaluation

In this final section we are going to show the results obtained by computing the evaluation metrics. In order to understand how much is good our ontology we have compared it with several others found in the web, here we are reporting just two of them, please look at the section xxx to have more information.
In particular we computed 4 different metrics: Coverage, Flexibility, Extensiveness and Sparsity.The first reference schema is taken by :

Information	Data
Name	km4city
URL	http://wlode.disit.org/WLODE/extract?url=http://www.disit.org/km4city/schema#Support{_}activities{_}for{_}transportation
Ontology IRI	http://www.disit.org/km4city/schema
Authors	DISIT lab
Publisher	DISIT Lab, University of Florence, Italy, http://www.km4city.org
Coverage	0.01
Flexibility	0.02
Extensiveness	0.01
Sparsity	0.97

Comparing our ontology with this one, the first thing to say is that there is a very huge difference in the number of ETypes considered: the ontology provided by the DISIT Lab is indeed composed by 667 ETypes while our one just 18. In this case the most important metric is the Sparsity: it indicates that there is an important difference between the ETypes defined by us and those defined in the km4city ontology. The second thing to highlight is the similarity among the other metrics, indeed it indicates that our ontology is not well represented by the other schema.

Information	Data
Name	Tickets Ontology
URL	http://www.heppnetz.de/ontologies/tio/ns#TransportationService
Ontology IRI	
Authors	Martin Hepp
Publisher	
Coverage	0.26
Flexibility	0.42
Extensiveness	0.11
Sparsity	0.63

In this case it is possible to see as Sparsity is lower than in the case before, this is due to the amount of classes: in this case indeed the schema took as reference is composed by 42 ETypes and this means that there is no a huge difference between the schemes. In addition an higher Coverage value indicates that the ontology above is more similar to our one, this means that if going ahead in exploring the domain we could obtain a very interesting ontology. A good indicator is instead the flexibility, being almost at 50 means that our schema could potentially become a very good graph if integrated in order to explore more into details the domain.

2 Knowledge Graph Development Process

The second chapter of this document aims to describe, in a detailed way, the KG development process. The sections below describe each phase of the KG building project, reporting for each phase, the description of the datasets and their evolution respect the previous phases, the schema construction which will generate the KG ontology in the

end, as well as the description of the procedures adopted to manage the data and finally achieve those results. Moreover for each phase is reported an evaluation section, which aims to evaluate the quality of the results achieved at the end of each phase.

2.1 Scope Definition

Even if working, studying, visiting a city or a relative, practicing sports, tacking a trip seem totally different activities they have one common thing: they take us out of our houses. To get to our workplace, school or gym we have to move and this makes us spend one of the greatest parts of our day on transportation. To save even a little of our time we often spend much of it looking for a faster and a cheaper way to get to our destination. Transportation involves several parameters that each traveler evaluate carefully based on his needs. The main parameters taken into account are:

- routes
- modes of transport
- time spent
- cost

Our system wants to provide a solution able to solve two different problems in the Trentino area: giving all the useful information on the way of transport and providing all those useful facilities related to the road system. To do this our solution is based on an integration of data regarding the road system like routes, time spent on each path, public transports schedules, and its relative facilities like parking areas, petrol pumps, campsites.
To better link all the available datasets and create a system able to give important information instantaneously, the best way to proceed is using a Knowledge Graph. To construct it is very important to have a standardized methodology like the iTelos one. Thanks to the iTelos Methodology indeed, it is possible to divide the problem in several sub-problems being able to solve each one in an easier way. Those parts are, in particular:

- Inception, in this step the goal is to define the queries that could be posed and how to shape up the system itself.
- Informal Modeling, in this step the goal is to design an EER Model such to highlight all possible relations among data and entities.
- Formal Modeling, in this step the goal is to build up the real KG provided of a SKG and a LKG.
- Data Integration, in this step the goal is to finalize all the system providing a final DKG with the reference to the LKG and the SKG.

But before proceeding with the system design and its implementation there is a step 0 called "Scope Definition". The goal of this step is to provide a description of the scenario. To achieve this goal the idea is to describe several situations in which it is shown how the system can be extremely relevant to solve certain problem.

2.1.1 Scenario Description

In order to show in a better way the application scenario and how the system interact with people, let define some personas which could be used to generate examples of competency queries.

Claudio: he is currently studying at the High School and since he is still too young to drive a vehicle he always moves using public transport. In particular he makes use of it to go to his school which is located in the centre of the city while he lives in its suburbs. Claudio has always practiced basketball and three times a week he goes training and once a week he have a match with other locals basketball teams. In the weekend he enjoys spending time with his friends at some locals parks or hanging out to a nearby basketball court.
Andrea: he is currently working as salesman of a big company so he travels a lot all around the Trentino area. To move he usually makes use of the company car but sometimes, if his destination is really close to a railway station or a bus station he takes the public transport. In his free time he enjoys going hiking in the mountains and exploring new paths.
Elisa: she is a university off-premise student. To reach her faculty she often makes use of public transportation, she makes use of it also to go back home. She lives in the center of the city so, to move easier, she uses her bicycle or during the summer she likes walking. Elisa is very attached with her family so once every two weeks she goes back home to visiting them. In her free time she likes visiting the historical city centers of the nearby cities, while when it is warmer she likes exploring cycle paths.
Maria: she is currently working in Bolzano but she lives in Trento with her husband and her children. She is a very ordinary mother: she is always in rush. During the holiday she likes to plan her family trips and usually they travel by camper. Once a month they usually go to visit her mother in Molveno.

2.1.2 Storytelling Definition

Claudio studies at the Trento Scientific Highschool "Leonardo Da Vinci" and he lives in Gardolo, due to this every morning he must take the bus to go to the school. Claudio enjoys basketball very much and since he was a child he has always played with a team at the Sanbàpolis gym on the south of Trento.
His team is currently competing in a regional basketball tournament so once a week he plays against other teams. Sometimes the match location us pretty far so he has to take public transports to get there and the evaluating of the several routes and modes of transport could be very annoying and time-consuming. In order to save some of his time, Claudio often asks for a lift to his parents but they are not always available.
In his limited free time Claudio likes hanging out with his friends spending some hours along the Adige shore or in some other parks near the city center, therefore he often checks the bus schedule to get there.

Andrea is a salesman at "Dolomiti Energia", an energy provider company based on Trento. Since he has started working there, he spends much of his working day travelling around to the several agencies collocated in: Mezzolombardo, Pergine Valsugana, Rovereto and many others. To do these tours a company car has been provided to him.
Another main part of Andrea's job consists in writing emails and check stuff on the web, for this reason when he has a lot of work to do and he needs to use his laptop he often decides to take trains in order to feel sufficiently comfortable to work. Since the tours change from one day to another he spends a lot of time checking the train schedule and this is often the most annoying part of his job.
Andrea is a mountain lover and in his free time he likes hiking in order to explore new mountains paths of nearby mountains. Since he does not have its own car he often makes use of public transports as buses to get there.

Elisa is studying at the University of Trento, since she is from Tuscany she is currently living in Rovereto as off-site student even if she studies Sociology in Trento. All her activities are carried out in the city of Trento and often she

involves also her boyfriend, they likes very much walking on the Adige riverside and often they take the cable car to go to the Bondone.
Elisa is an eco-friendly person, she really likes going around using just her bike and indeed she uses it to go to the supermarket, to the library, to the coffee and all those other places which she enjoys. Even when she has not the possibility to take her bike she often use the "e-motion" service to move around. But while for her ordinary movement she takes the bike, to go to the university or to go back home she has always to use the public transports and since her schedules change everyday she spends a lot of time to plan very well every trip in order to save both time and money.
In her free time she likes very much camping with her friends and since she has acquired all the skills needed to plan a cheap tour she is always picked as planner of their trips. Elisa is very good at it in fact she always evaluate the cheapest and shortest way to get to the selected campsite, moreover she always think to what they could need during the trip as hospitals, supermarkets, fuel station.

Maria is a 58 years old woman who for the last three years has worked in Bolzano. She has always lived in Trento with her husband and her children and even if the movement between Trento and Bolzano has always been annoying she has never wanted to move there. To respect her daily routine she has always preferred to move autonomously using the family car but when her husband needs it she use the car sharing service. Due to all this rush she is always careful on the best route to take and she usually spend a lot of time by studying the several possible paths to get her destination.
During the family holidays she often rent a camper in order to visit all the Trentino and sometimes to move also to other countries. Even if she knows that this kind of holiday can be more expensive than staying in an hotel she really likes planning the several routes to be taken and the evaluating of the several rest areas.
One of her children really likes the cable cars, so during the weekend she often goes with her family to Sardagna. Moreover, once a month, they all go to visit her mother who lives on the shore of the Molveno Lake and there they often take a walk all around the lake.

2.1.3 People Description

In the following table we want to resume the main interests of the people described in the storytelling section highlighting in particular the usage of the system we want to provide.

Name	Age	Interest	Usage
Claudio	17	Go to school, to his basketball training or matches all around the Trentino. Hang out with his friends	Check the public transportation schedules and the position of the facility he needs
Andrea	35	Go to the work by car or sometimes by public transports. Go hiking in the Trentino mountains	Check the fastest ways to get his several destination. Planning his hiking
Elisa	23	Go to the university, go cycling and camping with friends	Check cycling routes and facilities, check public transportation schedules and cheaper campsites
Maria	58	Go to the work by using her car or the car-sharing service, save as much time as possible. Visit the Trentino during the holidays	Check fastest and cheapest routes, check the schedule of the public transports, check the position of all the facilities she needs

2.2 Inception

2.2.1 CQs definition

Starting from the people described in the scenario we have imagined which could be their needs and tried to identify those possible questions which could be done to our system. The following table contains person doing the questions and the possible answer that the system should provide.

Person	CQ #	Question	Answer
Claudio	1.1	List all the possible paths from Gardolo to Trento	Return all the routes from Gardolo Square to Trento Square highlighting modes of transport, time spent on each route, prices of each path, useful facilities available encountered in the path
Claudio	1.2	List each path from Gardolo to Trento duration	Return the "duration" associated to each path from Gardolo Square to Trento Square
Claudio	1.3	List all the train itineraries from Gardolo to Trento	Return paths with associated route, time spent, price, Railway Stations and mode of transportation that has field type: "train" from the closest Railway Station to Gardolo Square to the closest Railway Station to Trento Square
Claudio	1.4	Give the cost of the cheapest path from Gardolo to Trento	Find the path from Gardolo Square to Trento Square with the associated smallest price and return the value of "price"
Claudio	1.5	List all the bus itineraries from Gardolo to the High School "Leonardo Da Vinci"	Return the paths with associated route, time spent, price, buses stops which have the mode of transport field as: "bus" from Gardolo Square to the High School "Leonardo Da Vinci"
Claudio	1.6	Give the time spent of the fastest path from Gardolo to Trento	Find the path from Gardolo Square to Trento Square with the associated smallest time spent and return the value of "time spent"
Claudio	1.7	List all the costs from Gardolo to Trento	Return the associated "price" value of each path from Gardolo Square to Trento Square
Claudio	1.8	List all the Bus Stops from Gardolo to Trento	Find all the paths from Gardolo Square to Trento Square which have the mode of transport field as: "bus" and return for each path the location, address and schedule of the stops

Person	CQ #	Question	Answer
Claudio	1.9	Give the fastest path from Gardolo to the High School "Leonardo Da Vinci"	Return route, mode of transport, price, time spent and the list of useful facilities with the lower associated field: "duration" from Gardolo Square to the High School "Leonardo Da Vinci"
Claudio	1.10	Give the schedule of a specific bus stop in the weekend	From the stop with the specific "StopID" return the schedule of "Saturday" and "Sunday"
Claudio	1.11	List all the cycling starting and ending points closer to the San Bartolomeo train station	Search in a range of 10km all the cycling points and return road, location (latitude and longitude) and the address associated to the cycling points
Claudio	1.12	Give the closest train stop to a specific location	Search in a range of 5km all the Railway Stations and return location, address, schedule and route associated to the Railway Station in which the value of the route field "length" is the lowest
Claudio	1.13	Give the schedule of the closest bus stop	In a range of $5km$ from all the Routes with arrival point the location of a Bus stop and starting point the current location, return the associated schedule of the stop considering the route with the minimum value of "length"
Claudio	1.14	Give the availability of the pubic transportation for a specific route	Return the several values of the attribute "Type" of the ModeOfTransport for a specific route
Andrea	2.1	Give the fastest path from Trento to Mezzolombardo	Search for the path with the lowest value of the field "duration" from Trento Square to Mezzolombardo Square and return route, mode of transport, time spent, cost and all the associated facilities
Andrea	2.2	List all the possible paths from Trento to Cles	Return route, mode of transport, time spent, price and associated facilities of all the possible paths from Trento to Cles
Andrea	2.3	Give all the fastest train itinerary from Trento to Mezzolombrardo	Return the path from Trento Square to Mezzolombardo Square with associated route, price and Railway Stations which has the field type of the mode of transport as "train" and the lowest value of the "time spent"

Person	CQ #	Question	Answer
Andrea	2.4	List all possible paths from Trento to Pergine Valsugana	Return route,mode,price,time spent and all associated facilities of all possible paths from Trento to Pergine Valsugana
Andrea	2.5	List all the hiking paths in Trentino	Return route, time spent, starting point, final point and associated facilities of all hiking paths in Trentino
Andrea	2.6	Give longest hiking path in Trentino	From all the hiking paths in Trentino, return route, time spent, initial position, final position and facilities associated of the path with the maximum value of the route field "length"
Andrea	2.7	List all the available parking areas close to a specific point	Return contact, locations, addresses and price of the parking areas which has the value of the filed "length" of the associated route $<$ 5km from the specific point
Andrea	2.8	List all the available fuel station between Trento and Tione	From the list of all the paths starting in Trento Sqaure and arriving in Tione Square return all the available facilities of types "Fuel Station" with contact, locations, address, schedule and price
Andrea	2.9	List all the buses itinerary from Trento to Mezzolombardo	Return route, cost, time spent and facilities associated of all the paths starting in Trento Square and arriving in Mezzolombardo which have the field type of the mode of transport as "Bus"
Andrea	2.10	Give the closest parking area to a specific point	Return contact, location, address, route, time spent, price of the path starting in the specific point and arriving in the parking area with the lowest value of the filed "length" of the route
Andrea	2.11	Give the closest fuel station to a specific point	Find the facilities of which the value of the field type is "Fuel Station" and return contact, location and address of the one with the lowest value of the field "length" of the route
Andrea	2.12	Give the length of a specific hiking path	Return the value of the field "length" of the specific path
Andrea	2.13	Give the location of the cheapest fuel station close to Trento	From all the fuel station with the value of the field route $<$ 5km from the Trento Square return the location and the address of the one with the lowest value of the field price

Person	CQ #	Question	Answer
Andrea	2.14	Give the cost of an autonomous transportation from Trento to Mezzolombardo	Return the attribute "Price" of the ModeOfTransport
Andrea	2.15	Give the closest bus Stop from a specific location with the least waiting time	From all the routes with starting point the specific location and arrival point the location of the bus stop return the one with associated route with the minimum value of "Length" and least waiting time
Elisa	3.1	List all the possible paths from Rovereto to Trento	Return route, time spent, prices and mode of transportation of all the possible paths starting from Rovereto Square and arriving to Trento Square
Elisa	3.2	List all the bus itineraries from Rovereto to Trento	Return route, time spent, costs, schedule and bus stops of all the paths with the value of the field type of the associated mode of transportation equal to "bus" from the Rovereto Square to the Trento Square
Elisa	3.3	List all the e-bikes facilities location in Trento	Return location and address of all the e-bikes facilities in Trento
Elisa	3.4	Give the availability of all the e-bikes facilities in Trento	Return location and address of all the available e-bikes facilities
Elisa	3.5	List all the cycling itineraries in Trento	Return route, time spent, starting position, arrival position of all the cycling routes with the distance from the Trento Square $<$ 5km
Elisa	3.6	Give the cost of the cheapest path from Rovereto to Trento	Return the value "price" of the path with the lowest value of the associated price with starting position in Rovereto Sqaure and arrival position in Trento Square
Elisa	3.7	Give the cheapest train itinerary from Trento to Rovereto	Return route, time spent, price and all the facilities associated of the path starting in Roverto Square and arriving in Trento Square with lowest value of the field "price" and the value of the mode of transportation field type equal to "train"
Elisa	3.8	Give the longest cycling route of Trento	Return route, time spent, starting position, arrival position of the cycling routes with the distance from the Trento Square $<$ 5km and the maximum value of the field "length" of the route

Person	CQ #	Question	Answer
Elisa	3.9	Give the closest e-bike location to a specific point	Return location and address of the e-bikes facility with the lowest value of the field "length" of the route between the specific point and the facility location
Elisa	3.10	List all the campsites close to Trento	Return contacts, location, address, and price of all the campsites with the value of the field route < 5km starting from Trento Square
Elisa	3.11	Give the closest campsite to a specific point	Return contact, locations, addresses and price of the campsite which the lowest value of the filed "length" of the associated route starting from the specific point
Elisa	3.12	List the cost of all the campsites close to Trento	Return the value of the field "price" of all the campsites with the value of the field route < 5km starting from Trento Square
Elisa	3.13	Give the cost of the cheapest campsite close to Trento	Return the value of the field "price" of the campsites with the value of the field route < 5km starting from Trento Square and the lowest value of the field "price"
Elisa	3.14	Give the most economic path from Rovereto to the Sociology Department in Trento	From all the paths starting from the Rovereto Square and arriving to the Sociology Department return route, time spent, price, mode of transport and associated facilities of the one with the lowest value of the field "price"
Elisa	3.15	Give the available campsites on the weekend in Trentino	Return facilities of type: "Campisite" which are available during the weekend
Elisa	3.16	Give the dates of unavailability of a specific bus stop	Return the list of the dates in which the schedule indicates that the specific bus stop is unavailable
Elisa	3.17	Give the next arrival time of a specific bus stop	Return the schedule of a specific bus stop
Maria	4.1	List all the possible paths from Bolzano to Trento	Return route, time spent, price, mode of transport and all the associated facilities of all the paths from Bolzano Square to Trento Square
Maria	4.2	List all the fuel station between Bolzano and Trento	From all the path starting in Bolzano Square and arriving in Trento Square return contact, location, address, price of all the facilities with the value of the field type equal to "Fuel Station"

Person	CQ #	Question	Answer
Maria	4.3	Give the cheapest fuel stations in Bolzano	From all the facilities with the route value < 5km from the Bolzano Square search the one with value of the field type equal to "Fuel Station" and lowest value of the filed "price", return contact, location, address and price of it
Maria	4.4	List all the train itineraries from Trento to Bolzano	From all the paths starting in Trento Square and arriving in Bolzano Square return route, time spent, price and facilities of those with the field type of the mode of transport equal to "train"
Maria	4.5	List all the paths costs from Bolzano to Trento	For each path starting in Bolzano Square and arriving in Trento Square return the value of the field "price"
Maria	4.6	Give the closest fuel station to a specific point	Return contact, location, address, price of the facility with type "Fuel Station" which has the lowest value "Length" of the associated route
Maria	4.7	Give the closest available "car sharing" position to a specific point	Return location and address of the car-sharing facility with availability "1" and the lowest value of the field "length" of the route between the specific point and the facility location
Maria	4.8	Give the fastest path from Bolzano to Trento by using public transports	From the list of paths starting in Bolzano Square and arriving in Trento Square with the field Public of the mode of transport equal to "1" return the one with the lowest value of "time spent"
Maria	4.9	Give the cost of the fastest public transportation path from Bolzano to Trento	From all the paths starting in Bolzano Square and arriving in Trento Square with value of the field Public of the mode of transport equal to "1" search the one with lowest value of the field "time spent" and return its associated "price"
Maria	4.10	List all the possible paths from Trento to Molveno by using the car	Return route, time spent, price and facilities mode of all the possible path from Trento to Molveno with the field Public of mode of transport equal to "0"

Person	CQ #	Question	Answer
Maria	4.11	Give the cable car schedule from Trento to Sardagna	From all the path from Trento Square to Sardagna Square find those with the field type of the mode of transport equal to "cable car" and return the schedule
Maria	4.12	Give the fastest path from Trento to Molveno	From the list of paths starting in Trento Square and arriving in Molveno Square return the one with the lowest value of "time spent"
Maria	4.13	Give the cheapest path from Trento to Molveno	From the list of paths starting in Trento Square and arriving in Molveno Square return the one with the lowest value of "price"
Maria	4.14	Give the schedule of a specific bus stop	Return the schedule of a specific bus stop
Maria	4.15	Give the schedule of the fuel stations of a specific path	From the facilities associated to the specific path return the schedule of all the ones with the field "Type" equal to FuelStation

2.2.1.1 Inception Schema In this paragraph we are going to schematize the several queries. The next tables has 3 properties: the number of the query, the the associated type which is going to be returned to answer to the question and last the properties defining the type.

CQ #	Type	Properties
1:1-4-11 **2**:1-2-4-5-6-9 **3**:3-4-7-9-11-14 **4**:1-4-7-10-11	Generic (Facility)	Type, Availability
1:1 **2**:8-11-13 **3** **4**:2-3-6-15	Fuel Station (Facility)	Price
1:1 **2**:7-10 **3** **4**:	Parking Area (Facility)	Price
1:1-3-14 **2**:3 **3** **4**:	Railway Station (Facility)	
1:1-7-14 **2**:15 **3**:7 **4**:	Buses Station (Facility)	
3:10-11-15	Campsites (Facility)	
1:1-3-5-7-8-11-13 **2**:1-2-3-4-5-6-7-8-9-12-15 **3**:1-2-5-7-8-10-11-12-14 **4**:1-4-7-8-10-12-13	Route	Length, Speed limits
1:1-5-7-11-16 **2**:1-2-3-4-14 **3**:1-8-14 **4**:1-5-13	Mode of transport	Type
1:1-3-5-6-7-9-11 **2**:1-2-3-4-7-8-9-10-11-13-14 **3**:1-2-5-7-6-9-10-12-13-14 **4**:1-3-4-5-6-8-9-10-13	Price	Price
1:1-4-5-10-12-13-14-15 **2**:5-6-7-8-10-11-12-13 **3**:3-4-5-7-8-9-10-11**4**:2-3-6-7-10-12-14	Location	Latitude, Longitude, Altitude

CQ #	Type	Properties
1:1-2-3-5-7-8-11 **2**:1-2-3-4-5-6-7-8-9-10-11 **3**:1-2-5-7-8-10-11-14 **4**:1-4-7-8-9-10-12-13	Time spent	Duration
1:7-10-12-13-14-15 **2**:15 **3**:16-17 **4**:11-14	Stop	Schedule
2:7-8-10-11-13 **3**:3-4-5-7-10-11 **4**:2-3-6-7-10	Contact	Website, Phone, Email address
1:1-4-5-10-12-13-14 **2**:7-8-10-11-13 **3**:3-4-5-7-9-10-11 **4**:2-3-6-7-10-12	Address	Province, City, Village, Street, Number, CAP

2.2.2 Initial Datasets description

In this section are reported the metadata at datasets level involved in the Inception phase, so those metadata regarding the sources, the authors, the collection methods, and other meaningful information. In this step we have identified twelve datasets which are useful to answer to the several quires.

- **Trentino public transport Urban TTE**: this dataset contains the *core data* about the city of Trento;
- **Trentino public transport Extra-Urban TTE**: this dataset contains the *core data* of routes in the Trentino province;
- **Trentino public transport rates for Urban TTE**: contains the *core data* of prices of routes in Trento;
- **Trentino public transport rates for Extra-Urban TTE**: contains the *core data* of prices of routes in Trentino Province;
- **Cycle paths**: contains the *common data* of cycle paths routes;
- **Cycling points of interest**: contains the *common data* of cycle point of interest;
- **Italian Parking Areas**: contains the *contextual data* of the parking places;
- **Railway Stations**: contains the *common data* of the railway's stations;
- **Car sharing (Open Data)**: contains the *common data* of the Car sharing information.;
- **Map of petrol stations in Italy**: contains the *contextual data* of the gas stations around Italy;
- **Mountain Paths**: contains the *common data* of the mountain paths;
- **Campsite and other accommodation facilities**: contains the *contextual data* of the campsites and their relative prices.

1. Trentino public transport Urban TTE.
   This dataset is beased on the GTFS standard and have eleven different attributes (agency, calendar, calendar_dates, feed_info, routes, shapes, stop_times, stops, stoplevel, transfers, trips). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	Description
Dataset Identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Dataset Publisher	Name: Public Transport Service, IPAVAT Code: 0OK0PZ
Date of modification	2017-10-24
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Languages of the dataset	Italian
Holder	Autonomous Province of Trento
Refresh Rate	Half yearly
Author	Name: Public Transport Service, IPA VAT: 0OK0PZ
Url	https://www.trentinotrasporti.it/opendata/google_transit_urbano_tte.zip
License	Creative Commons Attribution 4.0 International (CC BY 4.0)
License_Type	https://w3id.org/italia/controlled-vocabulary/licences/A21_CCBY40
Format	txt

2. Trentino public transport Extra-Urban TTE.
   This dataset is beased on the GTFS standard and have eleven different attributes (agency, calendar, calendar_dates, feed_info, routes, shapes, stop_times, stops, stoplevel, transfers, trips). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	Description
Dataset Identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Dataset Publisher	Name: Public Transport Service, IPA / VAT Code: 0OK0PZ
Date of modification	2017-10-24
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Languages of the dataset	Italian
Holder	Autonomous Province of Trento
Refresh Rate	Half yearly
Author	Name: Public Transport Service, IPA / VAT: 0OK0PZ
Url	https://www.trentinotrasporti.it/opendata/google_transit_extraurbano_tte.zip
License	Creative Commons Attribution 4.0 International (CC BY 4.0)
License_Type	https://w3id.org/italia/controlled-vocabulary/licences/A21_CCBY40
Format	txt

3. Trentino public transport rates for Urban TTE.
   This dataset is beased on the GTFS standard and have seven different attributes (fare_attributes_urbano.txt, fare_attributes_urbano_cartascalare, fare_attributes_urbano_mobile, fare_rules_urbano,

fare_rules_urbano_cartascalare, fare_rules_urbano_mobile, zones_urbano). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	**Description**
Dataset identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Dataset Publisher	Name: Public Transport Service, IPA / VAT Code: 0OK0PZ
Date of modification	2019-05-09
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Languages of the dataset	Italian
Holder	Autonomous Province of Trento
Refresh rate	Half yearly
Author	Name: Public Transport Service, IPA / VAT: 0OK0PZ
Url	https://dati.trentino.it/dataset/6d5c2000-972e-4c21-aef6-fdbba94418a8/resource/44efc0bd-223a-49c7-b3b0-16128e32813c/download/tariffegtfsurbano.zip
License	Creative Commons Attribution 4.0 International (CC BY 4.0)
License_Type	https://w3id.org/italia/controlled-vocabulary/licences/A21_CCBY40
Format	txt

4. Trentino public transport rates for Extra-Urban TTE.
   This dataset is beased on the GTFS standard and have seven different attributes (fare_attributes_urbano.txt, fare_attributes_urbano_cartascalare, fare_attributes_urbano_mobile, fare_rules_urbano, fare_rules_urbano_cartascalare, fare_rules_urbano_mobile, zones_urbano). Each attribute is in separate txt file. General description of this dataset is in table below:

Info	**Description**
Identifier	p_TN: d3c9f167-3271-4a43-b5c1-e0879aa5ad3f
Publisher	Name: Public Transport Service, IPA / VAT Code: 0OK0PZ
Date of modification	2019-05-09
Geographic coverage	Trento
URI of GeoNames	https://www.geonames.org/3165241
Languages of the dataset	Italian
Holder	Autonomous Province of Trento
Refresh rate	Half yearly
Author	Name: Public Transport Service, IPA / VAT: 0OK0PZ
Url	https://dati.trentino.it/dataset/6d5c2000-972e-4c21-aef6-fdbba94418a8/resource/10e93dd1-3463-4664-8c24-300a7403780a/download/tariffegtfsextraurbano.zip

Info	Description
License	Creative Commons Attribution 4.0 International (CC BY 4.0)
License_Type	https://w3id.org/italia/controlled-vocabulary/licences/A21_CCBY40
Format	txt

5. Piste ciclabili (Cycle paths).
   General description of this dataset is in table below:

Info	Description
Landing page	https://siat.provincia.tn.it/IDT/vector/public/p_tn_3ff5db13-8a3d-4bd8-8d6f-9b2fdf1aeb41.zip
Created	26.09.2008
Coordinates	[[[10.41, 46.6], [11.97, 46.6], [11.97, 45.6], [10.41, 45.6], [10.41, 46.6]]] Type: Polygon
Contact	mailto: serv.naturambiente@provincia.tn.it
Identifiers	p_TN:3ff5db13-8a3d-4bd8-8d6f-9b2fdf1aeb41_resource
Format	cpg,dbf,prj,shp,shx

6. Punti di interesse ciclabili (Cycling points of interest).
   Representation of the punctual elements present on the Trentino cycle paths: bicigrill (refreshment point, assistance and information), counters (instrumentation for detecting pedestrian and cycle paths), cippi km and fountains. General description of this dataset is in table below:

Info	Description
Landing page	https://siat.provincia.tn.it/IDT/vector/public/p_tn_0211d261-70d8-485e-9265-b1c27b1a84e1.zip
Created	30.04.2019
Coordinates	[[[10.41, 45.6], [10.41, 46.6], [11.97, 46.6], [11.97, 45.6], [10.41, 45.6]]] Tipo: Polygon
Contact	mailto: serv.naturambiente@provincia.tn.it
Identifiers	p_TN:0211d261-70d8-485e-9265-b1c27b1a84e1_resource
Format	cpg,dbf,prj,shp,shx

7. Parking map in Italy.
   Archive, which can be represented on the map, which contains the non-exhaustive list of over 21,000 car parks in Italy. The source of the data is OpenStreetMap.org which has been assigned with automated procedures the classification by municipality, province and region. Data updated on: 23 February 2016 The data was created by DatiOpen.it on 23 February 2016

Info	Description
Landing page	http://www.datiopen.it/it/opendata/Mappa_dei_parcheggi_in_Italia
Created	23.02.2016
Contact	mailto: info@datiopen.it
AUTHOR	OpenStreetMap http://www.datiopen.it/it/catalogo-opendata/openstreetmap-org
PUBLISHED BY	Open.it
Format	xlsx

8. Train stations (Open data).
   Station of the railway stations in the municipal area of Trento. It includes the Brenner railway, the Trento_Malè_Marilleva railway and the Valsugana railway. Data provided by Trentino Trasporti.

Info	Description
Landing page	https://www.comune.trento.it/Aree-tematiche/Cartografia/Download/Stazioni-treno
Created	01.01.2017
Contact	mailto: Servizio.innovazionedigitale@comune.trento.it
AUTHOR	Trentino Trasporti https://www.trentinotrasporti.it/
PUBLISHED BY	Trentino Trasporti
Format	gml,shp,kml,dxf

9. Car sharing (Open Data).
   Location of Car sharing stalls Parking spaces dedicated to the collection and delivery of Car sharing vehicles. Data taken directly from the site https://www.carsharing.tn.it Car sharing allows you to have a car suitable for family or business needs without owning one and without incurring fixed costs (road tax, insurance, maintenance, garage or parking), but paying only in proportion to use.

Info	Description
Landing page	https://dati.trentino.it/dataset/car-sharing-open-data
Created	04-05-2018
Contact	mailto: Servizio.innovazionedigitale@comune.trento.it
AUTHOR	Name: Municipality of Trento IPA / VAT: c_l378
PUBLISHED BY	Trentino Trasporti
License	http://creativecommons.org/publicdomain/zero/1.0/deed.it
Format	gml,shp,kml,dxf

10. Map of petrol stations in Italy.
    Archive, which can be represented on the map, which contains the non-exhaustive list of over 13,000 petrol stations in Italy. The source of the data is OpenStreetMap.org which has been assigned with automated

procedures the classification by municipality, province and region.

Info	Description
Landing page	http://www.datiopen.it/it/opendata/Mappa_dei_distributori_di_carburante_in_italia
Created	23-02-2016
Contact	mailto: Servizio.innovazionedigitale@comune.trento.it
AUTHOR	OpenStreetMap
PUBLISHED BY	DatiOpen.it
License	http://opendatacommons.org/licenses/odbl/
Format	xlsx

11. Province of Trento Paths.
    Paths of the entire network of the Società degli Alpinisti Tridentini (SAT) that insist on the territory of the Autonomous Province of Trento: each path consists of the spatial coordinates that allow it to be correctly positioned on the territory and presents a series of additional textual information (attributes) that describe.

Info	Description
Landing page	http://www.datiopen.it/it/opendata/Provincia_di_Trento_Sentieri
Created	15-12-2015
Contact	mailto: sentieri@sat.tn.it
AUTHOR	SAT (Society of Tridentine Alpinists)
PUBLISHED BY	DatiOpen.it
License	http://opendatacommons.org/licenses/odbl/
Format	shp,json,xml,csv

12. Autonomous Province of Trento List of non-hotel structures.
    The archive contains information on non-hotel accommodation facilities in the territory of the Autonomous Province of Trento: rural businesses, bed-and-breakfasts, campsites, hostels, holiday homes, etc. Where available, the data contains the address, telephone, e-mail address, website and other information.

Info	Description
Landing page	http://www.datiopen.it/it/opendata/Provincia_Autonoma_di_Trento_Elenco_strutture_extra_alberghiere?metadati=showall
Created	25-10-2017
Contact	mailto: info@open.it
AUTHOR	Trentino Alto Adige Region
PUBLISHED BY	DatiOpen.it
License	http://opendatacommons.org/licenses/odbl/
Format	xls,csv,json

2.2.3 Datasets metadata documentation

Datasets following the GTFS format have a standard metadata definition[1].
Regarding the datasets containing the information for urban and extra urban:

- **agency.txt**: it contains information about the agency - Trentino Trasporti;
- **calendar.txt**: it contains information about the operational days of the week;
- **calendar_dates.txt**: exceptions to the pattern described into *calendar.txt*;
- **feed_info.txt**: information about the publisher;
- **routes.txt**: we will use the information contained into these files to obtain urban and extra-urban paths;
- **shapes.txt**: it contains the path that the bus travels along a route;
- **stop_times.txt**: we can extract all the information regarding travel time and time scheduling;
- **stops.txt**: into this file we can extract all the stops information like name and coordinates;
- **transfers.txt**: information about transfers between lines;
- **trips.txt**: information about service available a certain route.

Moreover, the datasets containing the prices will be integrated with the above information to provide a complete overview over a certain route; all the metadata descriptions about the Trentino Trasporti's datasets are found here. Further meta description and origin of the attributes are available at this github repository.

2.2.4 Datasets collection process

In this section we are going to explain the method used to collect the several datasets. In particular we have divided the process in 4 different steps as described in the iTelos Methodology. Entering into details:

2.2.4.1 Iteration Zero In the first step we verified that datasources were compliant with the information we needed for the competency queries in *common data* typology.
Indeed, we realized that we were missing *the campsite and accomodation dataset*; we then proceed to add this dataset to the datasources sheet.

2.2.4.2 Iteration One In the second step we downloaded the following datasets in *common data* typology: Cycle paths, Cycling points of interest, Railway Stations and Mountain Paths. The kind of information we cared the most were the coordinates of the cycling points.
We then added a metadata description of both the dataset and the attributes to have a more descriptive information.
A lot of **CQs** require locations points that are found in these datasets, so that the reason why we decide to categorized them as *common data*.

[1]Google metadata description: https://developers.google.com/transit/gtfs/reference

2.2.4.3 Iteration Two

In the third iteration we extracted all the data for the *core data* typology. The datasets collected by the datasources were using the GFTS standard, so that metadata description were already available together with the Trentino Trasporti dataset.
In this iteration the following datasets were extracted and categorized as *core data* according to the **CQs**: Trentino public transport Urban TTE, Trentino public transport Extra-Urban TTE, Trentino public transport rates for Urban TTE, Trentino public transport rates for Extra-Urban TTE.

2.2.4.4 Iteration Three

In the last iteration we collected the *contextual data* typology by extracting the following datasets: Italian Parking Areas, Car Sharing, Italian Gas Stations, Campsite and other accommodation facilities.
As long as some prices were missing in some *campsite and accomodations* dataset, we elaborated the data price attribute by calculating **mean** and **variance** over the available information on a specific category and assumed that they the fit a **normal distribution**. We then randomly fill the missing data accordingly to obtain cleaned dataset.

2.2.5 Inception level evaluation

The last section of the Inception phase report the evaluation of the outcomes obtained in this phase, through specif evaluation metrics.

2.3 Informal Modeling

The Informal Modeling step is divided in two main tasks: one at Schema Level and one at Data Level. Concerning the Schema Level the definition of the several ETypes that the KG is going to involve and the providing of the EER Schema are aimed, while on the data level the goal is to highlight the evolution of the datasets (those filtered and those new) and provide all their metadata.

2.3.1 Schema level

The Schema level is divided in two paragraphs in order to highlight in a better way the EER Schema and the evolution from the CQs defined in the Inception step.

2.3.1.1 ETypes and EER Model definition

The first step to design the EER Schema is to define the ETypes and identify which are Core ETypes, Common ETypes or Contextual ETypes and then defined their attributes and the relationship among them.
In addition to the ETypes some structures have been identified in order to specify in a better way some of the attributes defined in the ETypes.
The Core ETypes are the sine qua non entities defining our project. In the next lists we are going to highlight those ETypes considered as the most important for the Transportation Domain:

1. Path: it is the all-encompassing and top level EType and it represents the set of all the routes, mode of transport, prices and related facilities of the trip taken.
    - Route: this attribute is an array of the EType Route. It is the only attribute of the EType Path, this because we considered the Path as a set of routes that could be taken to arrive to a specific point starting from another one, all the other ETypes are so connected to the Route EType.

2. Route: among all, this EType is the most important, it is linked to all the other ETypes considered for achieving our goal and it involves all those attributes which the user can be interested in. Associating to the Route EType all the others it is possible to design a path between two specific points considering from time to time all the users need.

    - RouteID: it is the identifier of the route. Thanks to this it is possible to highlight which is the route to take.
    - SpeedLimit: it is the value of the speed limit of the route.
    - StartingPoint: it is an attribute defined by the EType Address and it is the starting point of the route.
    - ArrivalPoint: it is an attribute defined by the EType Address and it is the arrival point of the route.
    - ModeOfTransport: it is an attribute defined by the EType ModeOfTransport and indicates the mode of transport used to go through the route.
    - TimeSpent: it is an attribute defined by the EType TimeSpent and indicates the amount of time spent to go through the route.
    - Facility: it is an array of the EType Facility and it represents all those facilities associated to the route considered.
    - Length: it is the length of the route (i.e. the distance between the arrival point and the starting point).

3. ModeOfTransport: it is the EType describing the set of all the mode of transport which can be associated to a route. This EType has been divided in two other ETypes linked from an inheritance relationship.

    - Type: it is an attribute defined by the structure TransportEnum and indicates the mode of transport used to go through the associated route.

4. PublicTransport: it is the first child of the ModeOfTransport EType, it is used to describe the public transport which can be associated to the route.

    - Agency: it is an attribute defined by the EType Agency and represents all the specifics of the Agency providing the service to go through the associated route.
    - Ticket: it is an attribute defined by the EType Ticket and indicated the price of the fare to use the transportation service.
    - StopTime: it is an array of the EType StopTimes and it indicates the schedule of the public mode of tranport.

5. AutonomousTransport: it is the other child of the ModeOfTransport EType, it is used to describe the transportation method which do not required a service provided by a transportation agency.

    - FuelType: it is an attribute defined by the structure FuelTypeEnum and it provides the type of fuel used by the autonomous vehicle to go through the associated route.
    - Price: it is an attribute defined by the EType Price and it provides the cost of the mode of transport used.

6. Ticket (GTFS Format): it is the EType which describes all those attributes concerning the cost of the tickets, it is directly connected to public transportation.

- FareID: it is the identifier of the fare.
- Price: it is an attribute defined by the EType Price and it provides the cost of the mode of transport used.
- PaymentMethod: it indicates the payment method as defined by the GTFS Format.
- TransferDuration: it indicates the duration of the validity of the ticket.

7. StopTimes (GTFS Format): it is the EType specifying the time schedule of the stops made by the mode of transport selected, it is directly connected just to the PublicTransport EType.
   - ArrivalTime: it is an attribute defined by the EType TimeStamp and it represents the arrival time at a certain stop.
   - DepartureTime: it is an attribute defined by the EType TimeStamp and it represents the departure time from a certain stop.
   - StopID: it is the identifier of a stop.
   - StopSequence: it indicates the number of the stop during the trip.
   - Calendar: it is an attribute defined by the EType Calendar and it highlights the days in which the service is available for a certain stop.
   - Stop: it is an attribute defined by the EType Stop and it makes reference to all those information about the Stop (shown below).
8. Stop (GTFS Format): it is the EType involving all those attribute needed to characterize a stop of the public mode of transport.
   - StopID: it is the identifier of a stop.
   - StopCode: it is the code identifying a stop.
   - StopName: it is the name of the stop.
   - Location: it is an attribute defined by the EType Location and it represents the location of the stop in the world.
   - WheelchairBoarding: it indicates the presence of the wheelchair boarding along the stop.
9. TimeSpent: it is the EType indicating the time spent to go through a route.
   - TimeSpent: it is an attribute defined by the EType TimeStamp and it indicates the time spent to go through a route.

The Common Etypes are not necessary to design the KG but they are needed to improve the quality of the description of the whole environment in which the system is going to be established. In case of the Transportation Domain they characterize for example the prices, the facilities or the scheduling of the bus stop:

1. Facility: it is the EType defining all the facilities associated to a route, those considered for our system are those useful for the Transportation domain, both public or private.
   - Type: it is an attribute defined by the structure FacilityEnum and it characterizes the nature of the facility.

- Price: it is an array of the EType Price and it contains the prices of the service guarantee by the facility.
- Contact: it is an attribute defined by the EType Contact and it indicates the several ways to get in touch with the facility.
- Ranking: it indicates the service evaluation of the facility.
- Address: it is an attribute defined by the EType Address and it indicates the address and the information about the location of the facility.
- Calendar: it is an attribute defined by the EType Calendar and it indicates the time scheduling and the availability of the facility.

2. Agency (GTFS Format): it is the EType collecting all those information about the agency providing a public transportation service.
   - AgencyID: it is the identifier of the agency.
   - AgencyName: it is the name of the agency.
   - AgencyTimezone: it is the timezone of the agency.
   - AgencyLang: it is the language used by the agency.
   - Contact: it is an attribute defined by the EType Contact and it indicates the several ways to get in touch with the agency.
3. Price: it is the EType describing the costs of a ticket more than the costs of all the services provided by a facility or even the cost of a trip taken by using a private vehicle.
   - Cost: it is the cost.
   - CurrencyType: it is the currency of the payment.
4. Address: it is the EType describing all those information about the location of a facility more than an arrival or starting point of a route.
   - Province: it represents the Province of interest.
   - City: it represents the City of interest.
   - Street: it represents the Street of interest.
   - Number: it represents the Number of interest.
   - CAP: it is the CAP of the City of interest.
   - Location: it is an attribute defined by the EType Location and it represents the location in the world.
5. Calendar (GTFS Format): it represents the time scheduling of a public transport stop or of a facility day per day.
   - ServiceID: it is the service identifier.
   - Monday: it indicates if the service is available on Monday.
   - Tuesday: it indicates if the service is available on Tuesday.
   - Wednesday: it indicates if the service is available on Wednesday.

- Thursday: it indicates if the service is available on Thursday.
- Friday: it indicates if the service is available on Friday.
- Saturday: it indicates if the service is available on Saturday.
- Sunday: it indicates if the service is available on Sunday.
- StartDate: it indicates the service starting date.
- EndDate: it indicates the service ending date.
- Exceptions: it is an array of the EType CalendarDates and it represents those dates in which the service is interrupted.

6. CalendarDates (GTFS Format): it collects all those dates in which a service made by a facility or a public transportation agency is interrupted.
    - ServiceID: it is the service identifier.
    - Date: it indicates the date in which the service is interrupted.
    - ExceptionType: it indicates the reason of the interruption accordingly to the GTFS Standard.

The Contextual ETypes are the less important ones, they are not used to characterized the system designed but they are needed to characterize in a better way the Common and the Core ETypes, usually they store several attributes belonging to the ETypes in order to have a neater design.

1. Contact: it is the EType describing all the ways to get in touch with a facility more than a public transportation agency.
    - Phone: it indicates the main phone number of the facility.
    - Email: it indicates the main email of the facility.
    - Website: it indicates the url of the website used by the facility.
2. Location: it is the EType collecting the information about the world coordinates of an address, more than the location of a stop
    - Latitude: it indicates the latitude coordinate.
    - Longitude: it indicates the longitude coordinate.
    - Altitude: it indicates the altitude coordinate.
3. TimeStamp: it is the EType describing the time information format for the time spent going through a route more than the arrival time or departure time from a stop.
    - Hour: it indicates the hours.
    - Minutes: it indicates the minutes.
    - Seconds: it indicates the seconds.

In addition to the ETypes several structure have been introduce. In particular they describe the type of the fuel characterizing a vehicle, the several modes of transport which can be used to go through a route, the type of the service provided by the facilities.

1. FuelTypeEnum: it is an EType indicating the several fuel types which can be used in an autonomous trip.
    - NoFuel: it indicates when the trip can be taken without using any fuel (in our case when it is possible to go through a route on foot or by bike).
    - Diesel: it indicates when the trip is taken by using a vehicle with diesel engine.
    - Petrol: it indicates when the trip is taken by using a vehicle with petrol engine.
    - Gas: it indicates when the trip is taken by using a vehicle with gas engine.
    - Methane: it indicates when the trip is taken by using a vehicle with methane engine.
    - Electric: it indicates when the trip is taken by using a vehicle with electric engine.
2. TransportEnum: it is the structure indicating the several modes of transport which can be used to move through a route.
    - Train: it indicates the train as mode of transport.
    - Bus: it indicates the bus as mode of transport.
    - Car: it indicates the car as mode of transport.
    - CableCar: it indicates the cable car as mode of transport.
    - Bike: it indicates the bike as mode of transport.
    - Foot: it indicates no mode of transport.
3. FacilityEnum: it is the EType indicating the several facilities involved by our system, in particular those considered are all the facilities related to the road system.
    - FuelStation: it indicates a facility which provides fuel pumps.
    - ParkingAreas: it indicates a parking area.
    - RailwayStation: it indicates a railway station.
    - BusStation: it indicates a bus station.
    - CarSharingPark: it indicates a parking area which is provided by the car sharing service.
    - BikeSharingPark: it indicates a parking area for bikes which is provided by the bike sharing service.
    - Campsite: it indicates an area which provides the possibility of camping or parking campers or caravans.

Starting from the definition of all the ETypes we have designed the EER Schema. To highlight the difference among the Core, the Common and the Contextual ETypes they have been represented by using different colours:

- Core ETypes: blue.
- Common ETypes: yellow.
- Contextual ETypes: red (also for the structures the colour red has been considered).

Moreover several arrows have been used to highlight the links among the ETypes, in particular the head of the arrows have been changed according to the nature of the relationship between two ETypes:

- black dot cursor: it represents the structural attribute.

- full triangle cursor: it represents an inheritance relationship.
- simple arrow cursor: it represents other relationship types, furthermore a label on each arrow has been added so that it is possible to identify the nature of the relation itself.

In order to show in a better way the schema, it has been plotted hierarchically, the starting point is the EType Path that includes the most involved EType Route which in turn is connected to almost all the Core ETypes.
In the schema it is possible to recognize three different branches: the one related to the modes of transport domain, considering their costs and their "high level" specifications as the agency, the fuel type and the mode of transport type; the one related to the space domain, which involves the several specifications about the facilities like the address, the type of the facility, the contact, the location; the one related to the time domain, which includes the stop times of the public transportation, the time spent on each route, the calendar and the calendar dates ETypes.
The Informal Modeling Schema is shown in the figure 1.

2.3.1.2 Variance respect CQs definition

Between the Inception Schema and the Informal Modeling Schema several changes have been made, this in order to satisfy two main milestone of the project:

- Exploiting the already existing standards, our main one is the GTFS Format;
- Being able to create a modular system in order to adapt its functionality to any situation.

In order to describe in a clearer way the several changes we are going to: list all the entities added specifying when they belong to a particular format; list all the changes related to each already existing EType highlighting when necessary the attributes introduced.
Starting from the new entities:

- Ticket (GTFS Format): the ModeOfTransport EType is no longer directly attached to the Price EType but it goes through the Ticket EType, here are described all the attributes related to the fares.
- StopTimes (GTFS Format): thanks to this EType it is possible to totally describe the daily scheduling of each stop, including the arrival time and the departure time and the calendar.
- Agency (GTFS Format): this EType has been introduced in order to describe in a better way the service provided.
- CalendarDates (GTFS Fomrat): directly attached to the Calendar EType the CalendarDates EType is used to identify those particular days in which a service is unavailable.
- FuelTypeEnum: this contextual EType is used to list all the possible fuel types, it has been introduce to give a better qualification of the ModeOfTransport EType.
- FacilityEnum: this EType is involved in an important change. While in the Inception Schema the several types of facilities where treated with an inheritance relationship from the parent EType "Facility", now thanks to this EType it is possible to enclose all of them in just one.

While the changed entities are:

- Path: is no longer the most important EType which involved all the entities hierarchically below, but even if it maintains its role of top level EType now it is directly attached just to the Route EType.

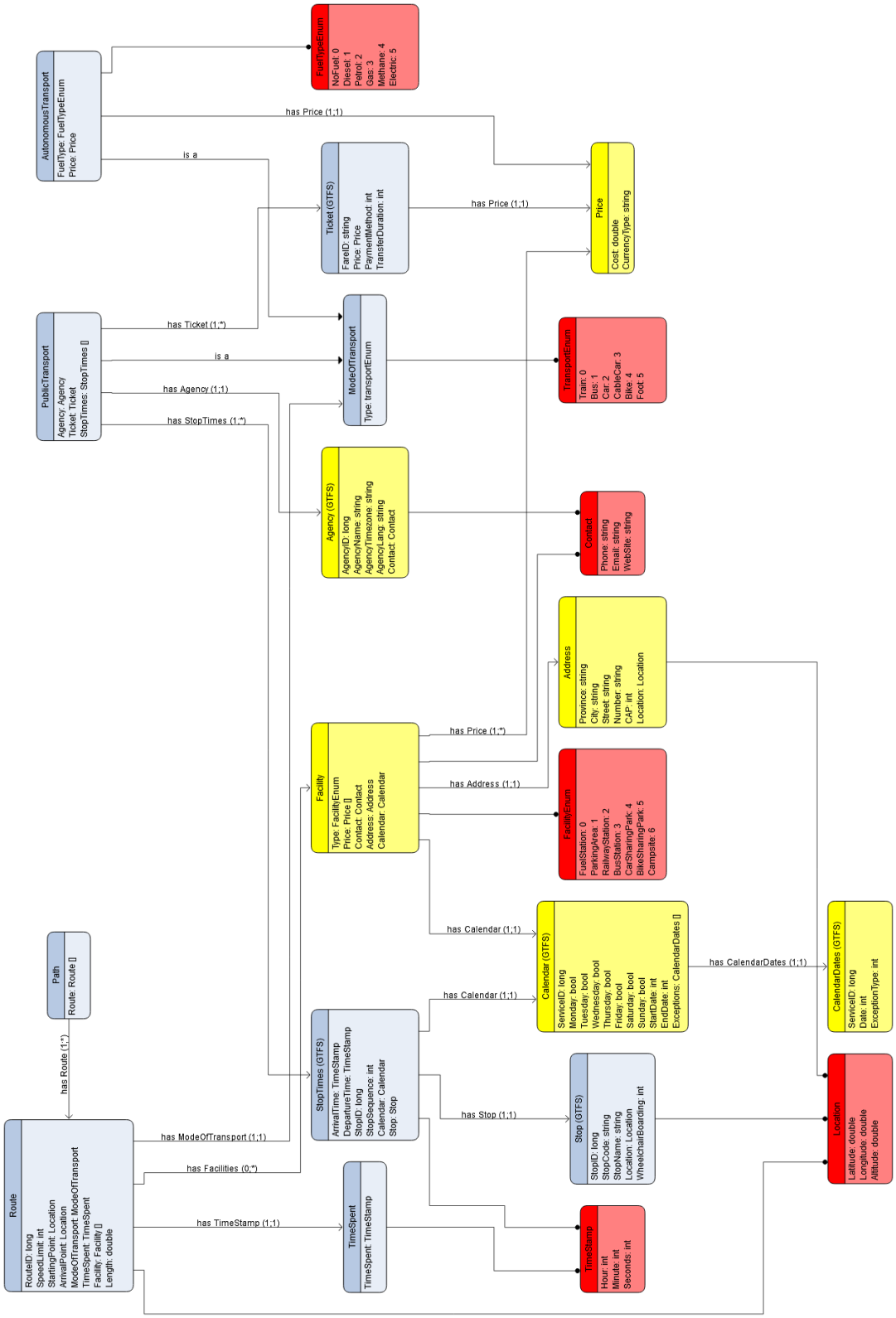


Figure 1: Informal Modeling Schema

- Route: this EType has now become the center of the whole schema, in particular each Route element involves a ModeOfTransport, a TimeSpent, a list of Facility and two Address.
- ModeOfTransport: this EType is the other one playing one of the most important role. While in the Inception Schema it was treated as a single EType now it has been split into two children EType: "PublicTransport" and "AutonomousTransport", in this way it is possible to highlight the difference between the ways of transport adoptable.
- Stop (GTFS Format): this EType already existing has been modified in order to follow the GTFS standard, so several attributes have been introduced qualifying in a better way the Stop EType.
- Facility: as it has already been shown above, the Facility EType has been subject to change so several attributes have been includes, the most important is the "Ranking" one in which it is shown the quality of the service provided.
- Price: to the EType Price it has been introduced the attribute CurrencyType in order to respect the GTFS format and to give an improvement of the description.

2.3.2 Data level

From the *Data Level* point of view we decide to export the **filtered** data into **json** format. The filtering scripts were developed based on the data that we needed from the dataset to model our knowledge. We use python along with different libraries such as pandas to read the csv datasets and filter and rename the different attributes. Moreover, we provided together with the exported json datasets their metadata description in **DCAT** specification. Encoded in both **RDF/Turtle** and **JSON-LD**. The metadata description of the ETypes attributes of a certain dataset is found in the previous phase section.

2.3.2.1 Datasets management process

The first thing that we have done was to check if more data were needed to fulfill the *CQs* requirements. We found and added a new dataset with information about **e-motion** bikesharing service in the Trentino province.
Regarding the data filtering, at the beginning we evaluated the usage of existing libraries to generate the **DCAT** specification and the filtering altogether. We abandoned this path because we found out it was time consuming and checking the **Socrata** library we saw that the online catalog that we should be exploited to *publish* our elaborated dataset was deprecated since July 2020. We then decided a manual approach with python scripts and mostly the *pandas* library to read the csv dataset, filter the needed columns, checking the informal Modeling Schema and exporting the datasets in the json format.
We decided to maintain the datasets in the *GTFS* format as they were because we were concerned that converting them in a single json could slow us down in the next phase.

2.3.2.2 Datasets metadata documentation

As described in the previous section, we generated the **dcat metadata** description of our datasets, both in *RDF/TURTLE* and *JSON-LD* file. We also renamed some columns to make them more explicit, even if they were already described in the metadata in the Inception phase.
Moreover, we have tried to install a Socrata catalog without success, but we were able to manage to install a **CKAN Catalog** on a server, where we are planning to share the datasets. The server is hosted in digitalocean and it is

available at this url http://167.99.139.12/ for the moment. All the metadata files are available in the respective metadata folder of the Informal Schema folder of the github repository.

2.3.2.3 Variance respect Inception datasets

In the **accomodation** dataset we extracted only: *address information*, *typology* and *price information*, renaming the field "Altitudine località turistica" to simply "Altitude"; this dataset contains also synthetic information about prices estimated from available data in the Inception phase.
We keep all the information in the added **e-motion bikesharing** dataset, but we merged all the files into one.
In the **car sharing** dataset we extracted: *nomepos*, *via* and *geometry*.
In the **cycling points** we filtered: *OBJECTID*, *DES_TIPO_E*, *DESCRIPTION* and *GEOMETRY*.
In the **cycling routes** script we extracted from the dataset: *OBJECTID* and *geometry*.
In the **fuel pumps maps** dataset we extracted: **coordinates information**, *address information* and *name*.
In the **mountain paths** dataset we extracted: *Path information*, *difficulty*, *start place* and *end place*.
In the **parking areas** dataset we extracted *coordinates information*, *address information* and *name*.
Finally in the **stazioni** dataset we extracted: *name* and *geometry*.
In the dataset pre-processing we also taking into account the evolution of the Informal Modeling Schema during the different iterations.

2.3.3 Informal Modeling Evaluation

The last section of the Informal Modeling phase report the evaluation of the outcomes obtained in this phase, through specif evaluation metrics.

2.4 Formal Modeling

This section is dedicated to the Formal Modeling phase description. As usual it is divided in two main parts: the schema level, handled by the knowledge engineer and the data level, handled by the data scientist.

2.4.1 Schema level

2.4.1.1 Ontology definition

To design our ontology following the steps illustrated by the iTelos Methodology we have used several tools but the two main ones have been the UKC software and the Protégé software. In particular it is possible to subdivide the work done in the two steps characterized by the use of these software:

1. L1,2: in this step, starting from the schema generated during the Informal Modeling phase, we have checked the presence of our ETypes in the UKC. This software is a vocabulary which assign an identifier to a concept. Doing this it is possible to refer to that concept by using a unique id, but the most important thing is that it is possible to create a link to the concept starting from any language which makes reference to it: the languages barriers are over.

   If a concept taking into consideration while building an EType was not present in the UKC we then searched for the meaning in schema.org and compiled a specific form to enlarge the UKC datase. If in schema.org was in turn not present we looked for the meaning in the "Practical English Usage, 4th Edition Paperback: Michael Swan's guide to problems in English" vocabulary.

2. L4: once associated all our ETypes to their GID we started to design our ontology schema by using the software Protégé 4.0. In particular, thanks to it, we have been able to produce a schema containing all the links between the EType in a more formalized way. Using this software we have been able to highlight 4 different categories: Classes, Data Properties, Object Properties and Enumeration. The classes represent the ETypes of the IM phase, the Data Properties specify the attributes of the ETypes, the Object Properties represent instead the links among the ETypes and last the Enumeration specify those structures listing the specification of some ETypes.

To show in a better way the classes and the relations among them we have subdivide each class and each property in several tables, moreover a screenshot by the Protégé software is provided:

- Classes. The Classes are the core components of the project and of the ontology, they have been taken directly from the Informal Modeling Schema and then assign to each one an identifier accordingly to the concept expressed.

Class	**GID**	**Label**	**Comment**	**Added**
Address_GID-36400	36400	Address	A sign in front of a house or business carrying the conventional form by which its location is described	No
Agency_GID-45084	45084	Agency	An administrative unit of government	No
CalendarDates_GID-120045	120045	CalendarDates	Tabular array of dates which are associated to a specific events	Yes
Calendar_GID-44719	44719	Calendar	A tabular array of the days	No
Contact_GID-120109	120109	Contact	All useful information to get in touch with someone	No
Duration_GID-80581	80581	Duration	The property of enduring or continuing in time	No
Enumeration_GID-34789	34789	Enumeration	A numbered list	No
Facility_GID-3012	3012	Facility	Something designed and created to serve a particular function and to afford a particular convenience or service	No
ModeOfTransport_GID-120044	120044	ModeOfTransport	Way in which transportation happens	Yes
Path_GID-46379	46379	Path	An established line of travel or access	No
Price_GID-28431	28431	Price	Value measured by what must be given or done or undergone to obtain something	No
PrivateTransport_GID-120043	120043	PrivateTransport	Personal or individual use of transportation vehicle	Yes
PublicTransport_GID-22138	22138	PublicTransport	Conveyance for passengers or mail or freight	No

Class	GID	Label	Comment	Added
Road_GID-22592	22592	Road	An open way (generally public) for travel or transportation	No
StopTime_GID-120042	120042	StopTime	Temporal parameters for a specific stop	Yes
Ticket_GID-111874	111874	Ticket	Provide with a ticket for passage or admission	No
TimeStamp_GID-120046	120046	TimeStamp	Hour,minutes and seconds of a duration	Yes
Location_GID-132	132	Location	A point or extent in space	No
Stop_GID-5446	5446	Stop	A brief stay in the course of a journey	No

Table 32: Classes Metadata

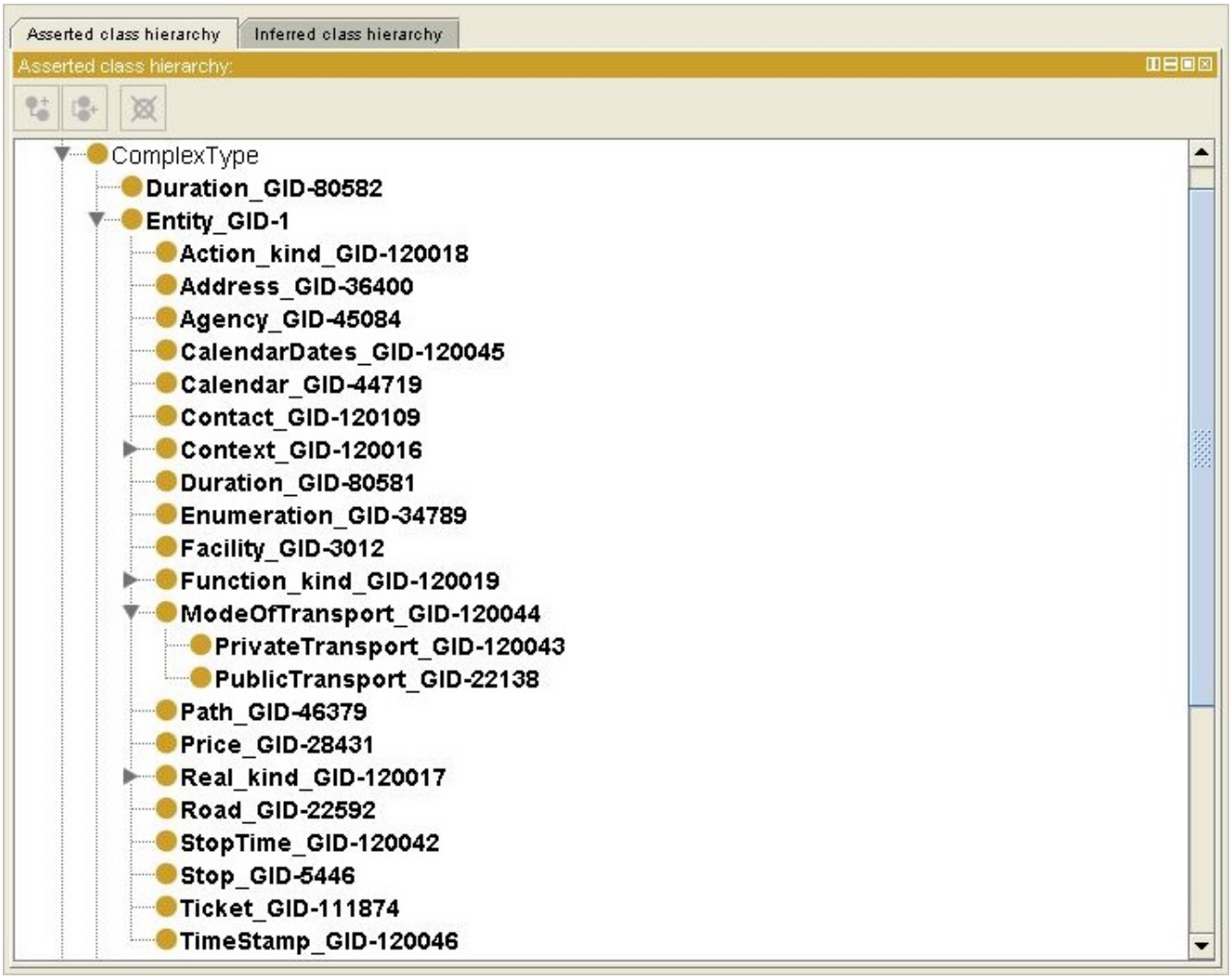


Figure 2: Classes structure in Protégé

- DataProperties. The DataProperties are all those properties characterizing the classes identified, they have a specific attribute which can be a complex or a simple one. In the design of our ontology no complex type has been used to define the classes.

Data Property	GID	Domain	Range	Added
has_AgencyID_GID-120057_Type-45084	120057	Agency_GID-45084	int	Yes
has_AgencyLanguage_GID-120060_Type-45084	120060	Agency_GID-45085	string	Yes
has_AgencyName_GID-120058_Type-45084	120058	Agency_GID-45086	string	Yes
has_AgencyTimezone_GID-120059_Type-45084	120059	Agency_GID-45087	string	Yes
has_Altitude_GID-28272_Type-132	28272	Location_GID-132	double	No
has_Latitude_GID-46263_Type-132	46263	Location_GID-132	double	No
has_Longitude_GID-46270_Type-132	46270	Location_GID-132	double	No
has_City_GID-45988_Type-36400	45988	Address_GID-36400	string	No
has_Number_GID-34489_Type-36400	34489	Address_GID-36400	string	No
has_Province_GID-46567_Type-36400	46567	Address_GID-36400	string	No
has_StreetAddress_GID-45807_Type-36400	45807	Address_GID-36400	string	No
has_ZipCode_GID-34110_Type-36400	34110	Address_GID-36400	int	No
has_Cost_GID-70407_Type-28431	70407	Price_GID-28431	double	No
has_CurrencyType_GID-120061_Type-28431	120061	Price_GID-28431	string	Yes
has_Date_GID-103420_Type-120045	103420	CalendarDates_GID-120045	int	No
has_ServiceID_GID-120056_Type-120045	120056	CalendarDates_GID-120045	int	Yes
has_Exception_GID-31741_Type-120045	31741	CalendarDates_GID-120045	int	No
has_EndDate_GID-120090_Type-44719	120090	Calendar_GID-44719	int	Yes
has_Monday_GID-80758_Type-44719	80758	Calendar_GID-44719	bool	No
has_Tuesday_GID-80759_Type-44719	80759	Calendar_GID-44719	bool	No
has_Wednesday_GID-80760_Type-44719	80760	Calendar_GID-44719	bool	No
has_Thursday_GID-80761_Type-44719	80761	Calendar_GID-44719	bool	No
has_Friday_GID-80762_Type-44719	80762	Calendar_GID-44719	bool	No
has_Saturday_GID-80763_Type-44719	80763	Calendar_GID-44719	bool	No
has_Sunday_GID-80757_Type-44719	80757	Calendar_GID-44719	bool	No
has_ServiceID_GID-120056_Type-44719	120056	Calendar_GID-44719	int	Yes
has_StartDate_GID-120091_Type-44719	120091	Calendar_GID-44719	int	Yes
has_FareID_GID-70599_Type-111874	70599	Ticket_GID-111874	string	No
has_PaymentMethod_GID-120074_Type-111874	120074	Ticket_GID-111874	int	Yes
has_TimeTableDuration_GID-120088_Type-111874	120088	Ticket_GID-111874	int	Yes
has_Hour_GID-81114_Type-120046	81114	TimeStamp_GID-120046	int	No
has_Minutes_GID-81154_Type-120046	81154	TimeStamp_GID-120046	int	No
has_Seconds_GID-72173_Type-120046	72173	TimeStamp_GID-120046	int	No

Data Property	GID	Domain	Range	Added
has_Length_GID-28259_Type-22592	28259	Road_GID-22592	double	No
has_RoadID_GID-120051_Type-22592	120051	Road_GID-22592	int	Yes
has_Phone_GID-34485_Type-39136	34485	Contact_GID-39136	string	No
has_Website_GID-34126_Type-39136	34126	Contact_GID-39136	string	No
has_Email_GID-105296_Type-39136	105296	Contact_GID-39136	string	No
has_SpeedLimit_GID-35726_Type-22592	35726	Road_GID-22592	int	No
has_StopCode_GID-120053_Type-5446	120053	Stop_GID-5446	string	Yes
has_StopID_GID-120051_Type-5446	120051	Stop_GID-5446	int	Yes
has_StopName_GID-120054_Type-5446	120054	Stop_GID-5446	string	Yes
has_WheelChairBoarding_GID-120055_Type-5446	120055	Stop_GID-5446	int	Yes

Table 33: DataProperties Metadata

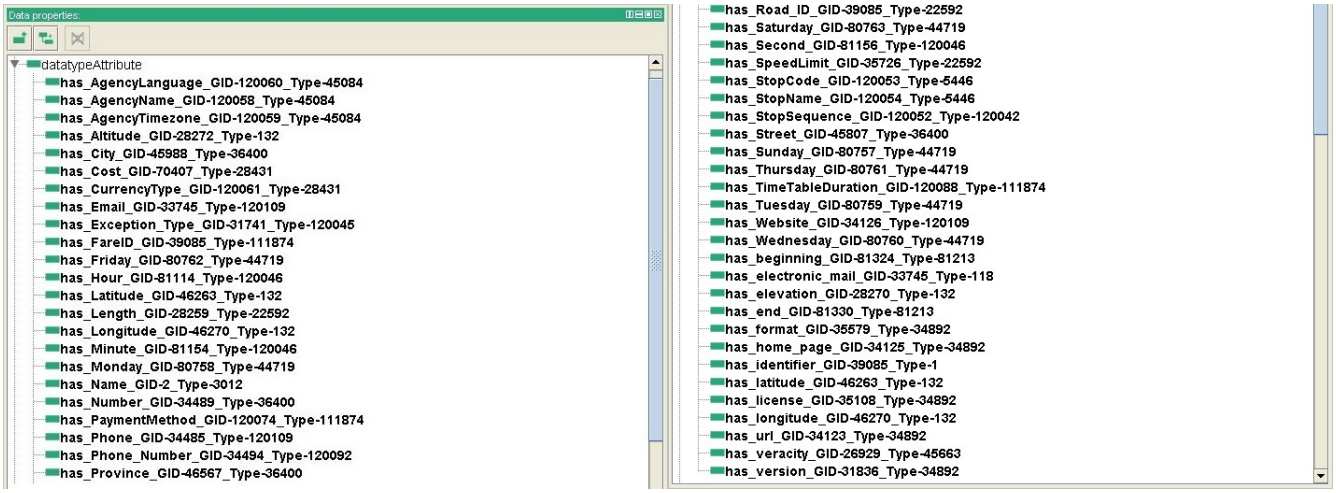


Figure 3: Data Properties structure in Protégé

- ObjectProperties. The Object Properties are those properties which highlight the links among all the classes of the ontology.

Object Property	Domain	Range	Card. Type	Card.
has_FacilityType_GID-103418_Type-3012	Facility_GID-3012	Enumeration_GID-34789	exactly	1
has_FuelType_GID-103418_Type-120043	PrivateTransport_GID-120043	Enumeration_GID-34789	exactly	1
has_ModeOfTransportType_GID-103418_Type-120044	ModeofTransport_GID-120044	Enumeration_GID-34789	exactly	1
has_EndingPoint_GID-46116_Type-22592	Road_GID-22592	Location_GID-132	exactly	1

has_StartingPoint_GID-81326_Type-22592	Road_GID-22592	Location_GID-132	exactly	1
has_Agency_GID-45084_Type-22138	PublicTransport_GID-22138	Agency_GID-45084	exactly	1
has_CalendarDates_GID-120045_Type-44719	Calendar_GID-44719	CalendarDates_GID-120045	exactly	1
has_Calendar_GID-44719_Type-120042	StopTime_GID-120042	Calendar_GID-44719	exactly	1
has_Calendar_GID-44719_Type-3012	Facility_GID-3012	Calendar_GID-44719	exactly	1
has_Contact_GID-39136_Type-45084	Agency_GID-45084	Contact_GID-39136	exactly	1
has_Duration_GID-80581_Type-22592	Road_GID-22592	Duration_GID-80581	exactly	1
has_FacilityAddress_GID-36400_Type-3012	Facility_GID-3012	Address_GID-36400	exactly	1
has_FacilityContact_GID-39136_Type-3012	Facility_GID-3012	Contact_GID-39136	exactly	1
has_Facility_GID-3012_Type-22592	Road_GID-22592	Facility_GID-3012	min	0
has_FacilityPrice-GID-28431_Type-3012	Facility_GID-3012	Price_GID-28431	min	1
has_IndividualPrice_GID-28431_Type-120043	PrivateTransport_GID-120043	Price_GID-28431	exactly	1
has_Location_GID-132_Type-36400	Address_GID-36400	Location_GID-132	exactly	1
has_Location_GID-132_Type-5446	Stop_GID-5446	Location_GID-132	exactly	1
has_ModeOfTransport_GID-120044_Type-22592	Road_GID-22592	ModeofTransport_GID-120044	exactly	1
has_Price_GID-28431_Type-111874	Ticket_GID-111874	Price_GID-28431	exactly	1
has_Road_GID-22592_Type-46379	Path_GID-46379	Road_GID-22592	min	1
has_StopTime_GID-120042_Type-22138	PublicTransport_GID-22138	StopTime_GID-120042	min	1
has_Stop_GID-5446_Type-120042	StopTime_GID-120042	Stop_GID-5446	exactly	1
has_Ticket_GID-111874_Type-22138	PublicTransport_GID-22138	Ticket_GID-111874	min	1

has_TimeStamp_GID-120046_Type-120042	StopTime_GID-120042	TimeStamp_GID-120046	exactly	1
has_TimeStamp_GID-120046_Type-80581	Duration_GID-80581	TimeStamp_GID-120046	exactly	1

Table 34: ObjectProperties Metadata

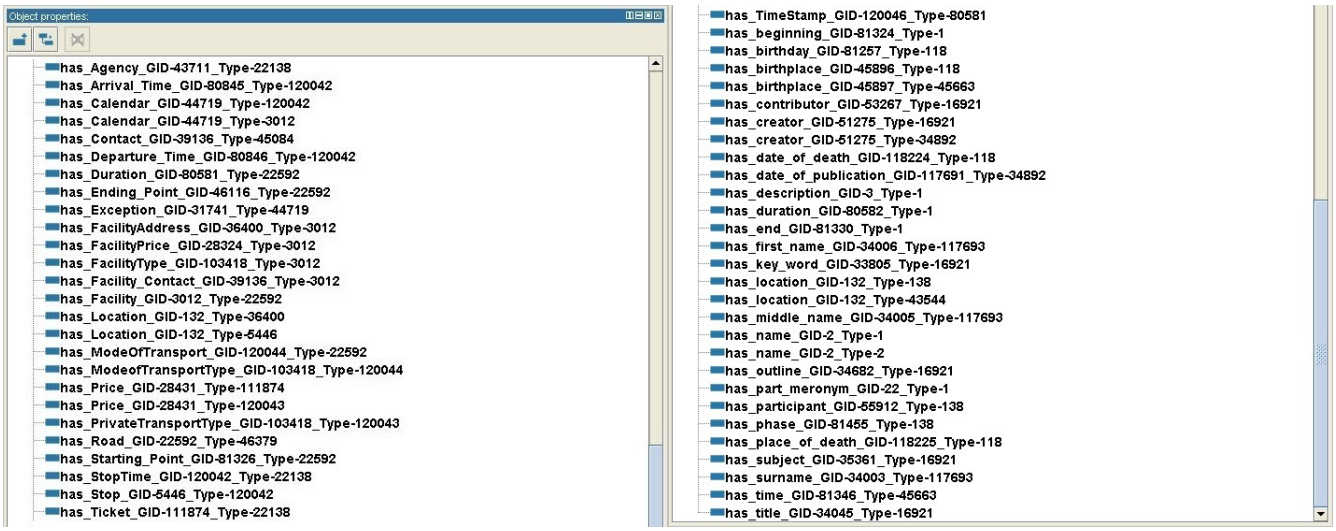


Figure 4: Object Properties structure in Protégé

- Enum. The enumeration are structures useful to define a range of value of an attribute composing a class (DataProperty). To show them in the ontology a class Enumeration has been created at the same level of the main others, then our enum has been treated as its children which in turn have as children the several possible values

Enum	**Element**	**GID**	**Label**	**Comment**	**Added**
TransportEnum_GID-120048		120048			Yes
	Bicycle_GID-15188	15188	Bicycle	A wheeled vehicle that has two wheels and is moved by foot pedals	No
	Bus_GID-15732	15732	Bus	A vehicle carrying many passengers	No
	CableCar_GID-15797	15797	CableCar	A conveyance for passengers or freight on a cable railway	No
	Car_GID-15945	15945	Car	A wheeled vehicle adapted to the rails of railroad	No

Enum	Element	GID	Label	Comment	Added
	Foot_GID-1429	1429	Foot	The act of traveling by foot	No
	Train_GID-18678	18679	Train	Wheelwork consisting of a connected set of rotating gears by which force is transmitted or motion or torque is changed	No
FacilityEnum_GID-120047		120047			Yes
	BikeSharing_GID-843	843	BikeSharing	The act of maneuvering a vehicle into a location where it can be left temporarily	No
	BusStation_GID-15745	15745	BusStation	A terminal that serves bus passengers	No
	CampsiteParking_GID-45940	45940	CampsiteParking	A site where people on holiday can pitch a tent	No
	FuelStation_GID-18641	18641	FuelStation	A service station that sells gasoline	No
	ParkingArea_GID-46375	46375	ParkingArea	A lot where cars are parked	No
	RailwayStation_GID-22321	22321	RailwayStation	Terminal where trains load or unload passengers or goods	No
FuelEnum_GID-120049		120049			Yes
	Diesel_GID-17309	17309	Diesel	An internal combustion engine that burns heavy oil	No
	Electric_GID-61771	61771	ElectricFuel	A physical phenomenon associated with stationary or moving electrons and protons	No
	Gas_GID-79121	79121	Gas	A fluid in the gaseous state having neither independent shape nor volume and being able to expand indefinitely	No

Enum	Element	GID	Label	Comment	Added
	Methane_GID-79566	79566	Methane	A colorless odorless gas used as a fuel	No
	Petrol_GID-78042	78042	Petrol	A volatile flammable mixture of hydrocarbons (hexane and heptane and octane etc.) derived from petroleum	No

Table 35: Enum Metadata

The schema is composed by two different parts, the first one is the structure taken by the KOS software in order to be aligned with the iTelos Methodology and then our ontology. To show the construction of the model we have taken two different screenshots which in turn are more detailed.

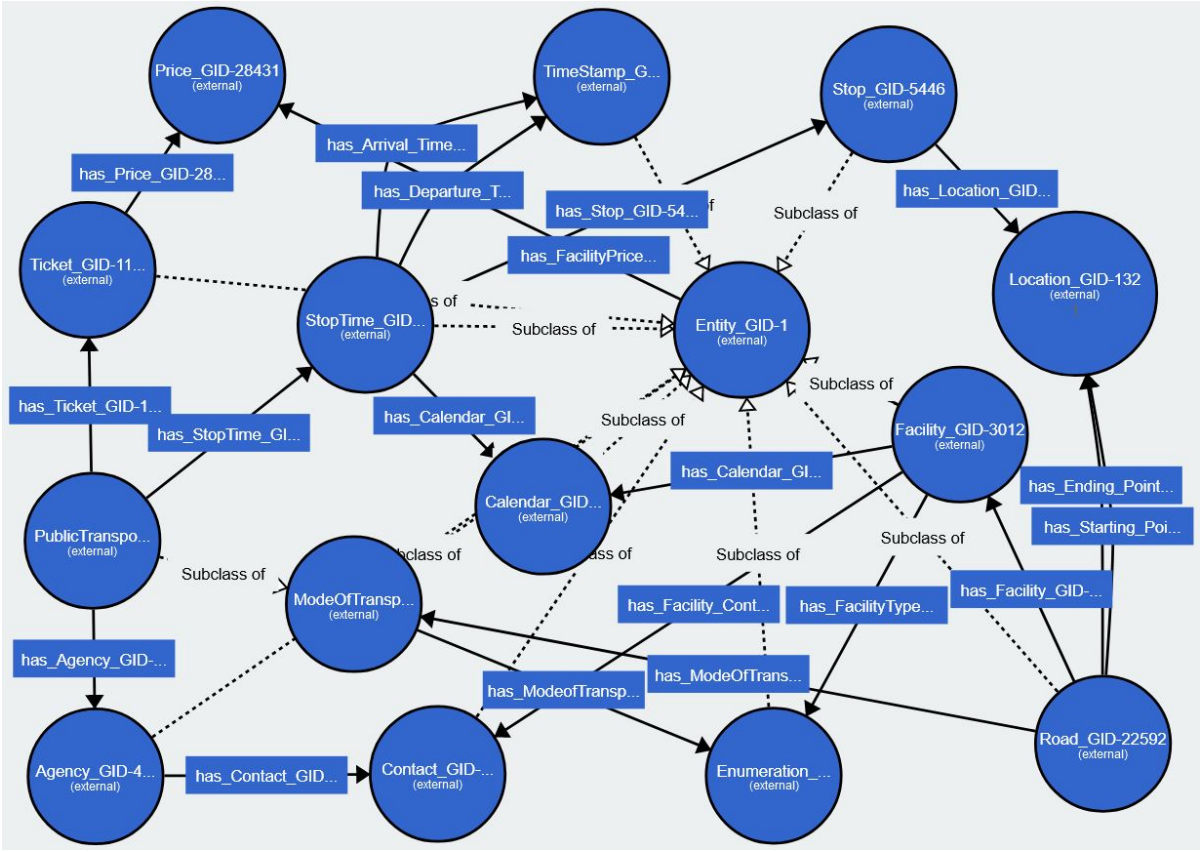


Figure 5: General Ontology Schema

In this screenshot we have kept just the classes defined by us, in particular it is possible to see that the structure of the ontology and the classes names are the same as in the EER model, the only thing changed is that, for simplicity, we have considered just one classes called Enumeration to include all the several EnumTypes attributes which specify the classes PrivateTransport, ModeOfTransport and Facility.

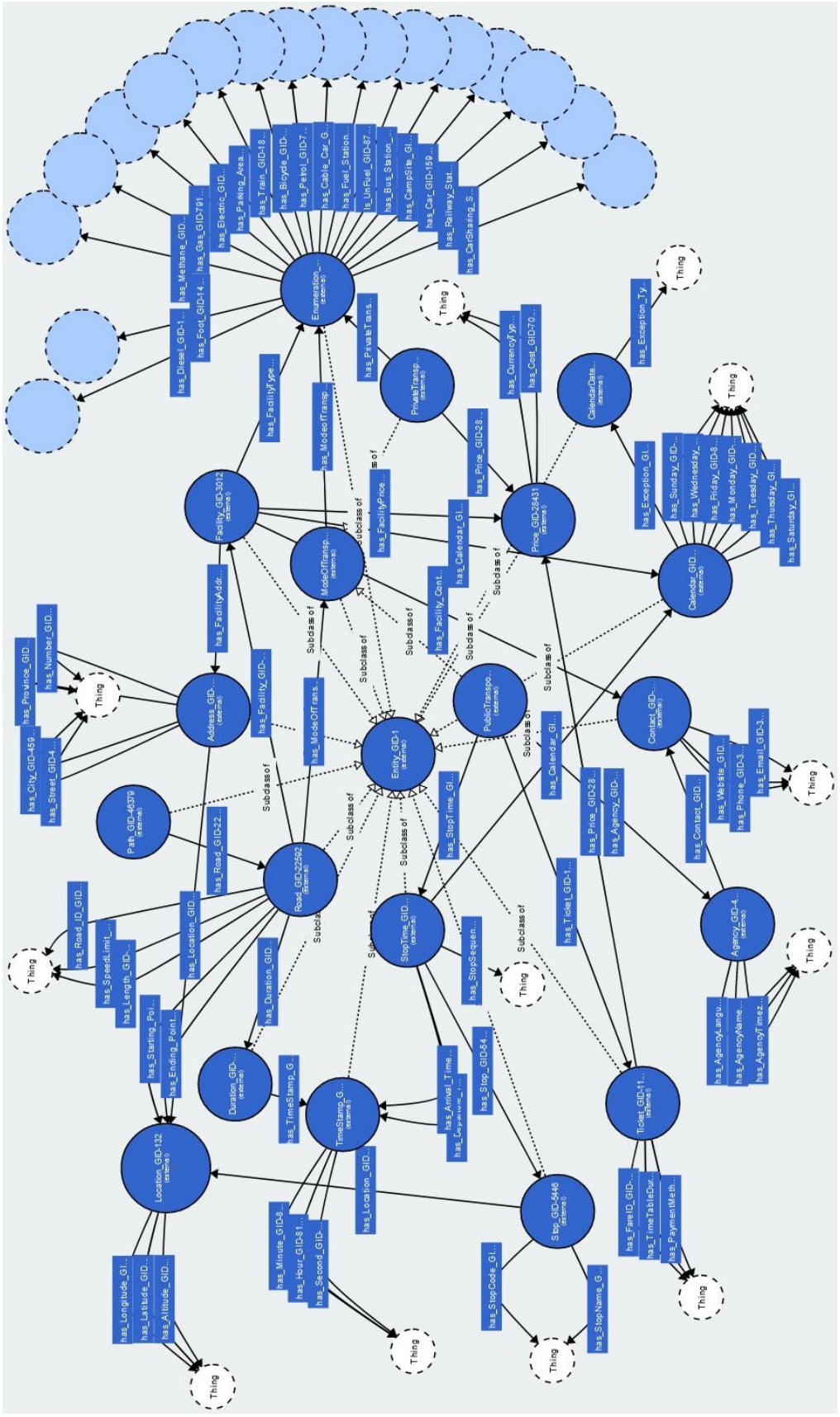


Figure 6: Expanded Ontology Schema

2.4.1.2 Variance respect to the EER Model

When designing the ontology some smaller changes have been done from the Informal Modeling Schema but the structure has been kept the one designed in the stage before. In particular we changed just the name of two ETypes (Route is now Road and TimeSpent is now Duration) in order to be aligned to those words provided in the UKC dataset expressing the same concept.

2.4.2 Data level

In this phase we will focus on two main activities in data layer, Data Alignment and Data Formatting. These operations regard two different aspects, the data alignment respect the ETypes used to represent the data, and the data formatting following the correct data types, the two sub-activities manage these actions respectively.
In data alignment we aim to reduce the gap between the data layer and knowledge layer.
Input of this phase were datasets from informal modeling phase with format of **json, geojson** and output format is **json**.

2.4.2.1 Formal Modeling datasets management

In this section are reported the operations and the tools adopted to format the dataset collected, in order to align them to the ontology definitions generated at schema level. To achieve this we defined a python class for each Etype, containing all attributes that used in formal schema. **Location, Address, Contact, Price, CalendarDates, Calendar, Facility, Road** are the classes that we defined. These classes can be serialized to export data in Json format as output. All the classes are in a file that is named **models.py**.
For data alignment we used python libraries such as **pandas**, **geopandas**, **geopy**, **json**. One of the main tasks in this phase was extracting necessary information from datasets in geojson format. Currently we have 4 datasets which are in geojson format, **stazioni, cycling-points, cycling-routes, carSharing** are datasets which have geometry as attribute with the type of **POINT or LineString**. These special types are used vastly in **GIS (Geographic Information System)** and **geopandas** library have been designed to handle these types of data. geopandas need The **Coordinate Reference System (CRS)** of dataset because the geometric shapes in a GeoSeries or GeoDataFrame object are simply a collection of coordinates in an arbitrary space. A CRS tells Python how those coordinates relate to places on the Earth. For example, one of the most commonly used CRS is the **WGS84 latitude-longitude** projection. This can be referred to using the authority code **"EPSG:4326"**.
First of all we convert EPSG which stands for European Petroleum Survey Group and is an organization that maintains a geodetic parameter database with standard codes. The base code for our datasets is 25832 which contain the area of Europe between 6°E and 12°E. By converting epsg to 4326 we set our coordinate system to World Geodetic System. Now our points can be recognized by all world opensource maps. After that by using geopy library which uses open street map api in backend, we will extract address class attributes which has been defined in **models.py**(province, city, street, number, cap, location). By using these python classes we will be make sure that all data in our datasets are aligned with Etype.
stazioni, cycling-points and carSharing datasets are aligned with the Etype Facility.
Cycling-routes dataset is aligned with Route Etype. At the end geojson datasets are aligned and formated with respect to the value types and exported as json format. On the other hand, datasets such as **accomodotaion, bikesharing, fuel-pumps, mountain-path, parking-areas** have been revised to be aligned with formal schema. Most of these datasets contained very dirty data about for instance the address, so we required to clean them to uniform for every output the very same standard. For instance to extract the number of the address which could be

a simple number or a group of numbers and characters we used a regex expression: **"[0-9]1,4/?[A-Za-z]?"**. Some other cleaning tasks were done in the *email field*; another example was the **fuel stations names** that were following no standard, some row had a *null* coloumn, so that we decided to thight together the facility type with the name: **"FACILITYTYPE - Name"**. By doing this we assure have enough flexibility in the data integration phase. After reading json file we mapped values of the each row to aligned formal value respect to type and class, and thus the informal schema given to us as input. For instance in **accomodation datasets** first we filtered necessary values, then we translate its header from Italian to English. This can be found in **accomodation filtering.py**. Then in **accomodation data cleaning.py** we extract **address, price, contact, facility** from each row. At the end we export datasets as json format. Generally, we named the scripts following this standard: *"dataset_activity.py"* where *activity* could be **filtering** or **data_cleaning**.

2.4.2.2 Datasets metadata documentation

Metadata were basically the same as the previous phase, we just translated the italian attributes to english to uniform the languages used in the datasets and avoid to encounter inconsistencies in the naming of the data in next phase.

2.4.2.3 Variance respect Informal Modeling datasets

This section aims to define the variance between the data elements (datasets and attributes within them) produced in this phase, and the initial datasets collected in the previous phase.
In the **cycling points** dataset we extract address **(province, city, street, number, cap, location)** from **geometry**.
In the **statzioni** dataset we extract address **(province, city, street, number, cap, location)** from **geometry**.
In the **carSharing** dataset we extract address **(province, city, street, number, cap, location)** from **geometry**.
In the **cycling routes** dataset we extract **Road (type, RoadID, SpeedLimit, StartingPoint, ArrivalPoint, ModeOfTransport, Timespent, Facility, Length)** from geometry.
In the **accomodation** dataset we extract the **Facility** class that contains the type of the facility, the price class, the contact class, the address class, the ranking (setted to 1) and the calendar.
In the **bikesharing** dataset we extract the **Facility** class that contains the type of the facility, the price class, the contact class, the address class, the ranking (setted to 1) and the calendar.
In the **fuel pumps** dataset we extract the **Facility** class that contains the type of the facility, the price class, the contact class, the address class, the ranking (setted to 1) and the calendar.
In the **mountain paths** dataset we extract the same value as in the previous phase but renaming them.
In the **parking areas** dataset we extract the **Facility** class that contains the type of the facility, the price class, the contact class, the address class, the ranking (setted to 1) and the calendar.
To have a deeper overview of all the attributes that we used, you may want to take a look at the *models.py* file.

2.4.3 Formal Modeling Evaluation

The last section of the Formal Modeling phase report the evaluation of the outcomes obtained in this phase, through specif evaluation metrics.
In this final section we are going to show the results obtained by computing the evaluation metrics. In order to understand how much is good our ontology we have compared it with several others found in the web, here we are reporting just two of them, please look at the section xxx to have more information.

In particular we computed 4 different metrics: Coverage, Flexibility, Extensiveness and Sparsity.The first reference schema is taken by :

Information	**Data**
Name	km4city
URL	http://wlode.disit.org/WLODE/extract?url=http://www.disit.org/km4city/schema#Support{_}activities{_}for{_}transportation
Ontology IRI	http://www.disit.org/km4city/schema
Authors	DISIT lab
Publisher	DISIT Lab, University of Florence, Italy, http://www.km4city.org
Coverage	0.01
Flexibility	0.02
Extensiveness	0.01
Sparsity	0.97

Comparing our ontology with the first one, the first thing to say is that there is a very huge difference in the number of ETypes considered: the ontology provided by the DISIT Lab is indeed composed by 667 ETypes while our one just 18. In this case the most important metric is the Sparsity: it indicates that there is an important difference between the ETypes defined by us and those defined in the km4city ontology. The second thing to highlight is the similarity among the other metrics, indeed it indicates that our ontology is not well represented by the other schema.

Information	**Data**
Name	Tickets Ontology
URL	http://www.heppnetz.de/ontologies/tio/ns#
Ontology IRI	http://purl.org/tio/ns
Authors	Martin Hepp
Publisher	Hepp Research
Coverage	0.26
Flexibility	0.42
Extensiveness	0.11
Sparsity	0.63

In this case it is possible to see as Sparsity is lower than in the case before, this is due to the amount of classes: in this case indeed the schema took as reference is composed by 42 ETypes and this means that there is no a huge difference between the schemes. In addition an higher Coverage value indicates that the ontology above is more similar to our one, this means that if going ahead in exploring the domain we could obtain a very interesting ontology. A good indicator is instead the flexibility, being almost at 50 means that our schema could potentially become a very good graph if integrated in order to explore more into details the domain.

Information	**Data**
Name	RouteType
URL	http://vocab.gtfs.org/terms#RouteType

Information	Data
Ontology IRI	
Authors	
Publisher	Lod2.eu
Coverage	0.2
Flexibility	0.4
Extensiveness	0.25
Sparsity	0.75

In this case the coverage and flexibility values are very similar with the previous case. We can observe a small difference in the extensiveness and sparsity values, which indicate two main things: slightly higher contribution of our scheme and a slightly better match between the ETypes of the schemes compared.

2.5 Data integration

In this section we are going to illustrate all the steps to integrate the dataset in order to obtain the final outcome of the project. In particular we started by using python in order to prepare the dataset in the format needed by Karmalinker to link them all then we proceed by integrating them in karmalinker and we are going to highlight the URI of all the classes and etypes, the queries for validation ontology and the data.

2.5.1 Data Preprocessing

Before starting integration of data in karmalinker we had to do some transformation in our datasets to be aligned with the formal schema. For instance we need to extract address and many other information that have been introduced in the formal schema from datasets which contains geojson format. For data transformation python libraries such as **numpy, pandas, os, random, geopandas, geopy** have been used. This was due to the fact that the way we elaborated the data in the previous steps was not optimal. Not knowing at the beginning an idea of what would have be the final shape that we needed to achieve to work optimally with Karma Linker we basically needed to re-elaborate the information. The list of the data sets and transformations are as follows:

- **stazioni dataset** After reading dataset from previous section with the help of the geopy library we extracted informations such as **state, city, road, number, cap** extracted from location points. To achieve this we queried latitude and longitude of the point in the opendatamap api with geopy helper. After data extraction seven csv files **Facility, Price, Contact, Address, Location, Calendar, CalendarDates** generated to be exactly the same as formal schema. For keeping the relation among the classes keyid have been used which holds the relation between files. In this case stationi is a facility so *Facility.csv* holds the key id of the *Price, Contact, Address, Location, Calendar, CalendarDates*. Each key id is unique (pandas indexes) so there wont be a conflict between the data in the integration phase.

- **carSharing dataset** After reading dataset from previous section with the help of the geopy library we extracted informations such as **state, city, road, number, cap** extracted from location points. To achieve this we queried latitude and longitude of the point in the opendatamap api with geopy helper. After data extraction seven csv files **Facility, Price, Contact, Address, Location, Calendar, CalendarDates** generated to be exactly the same as formal schema. For keeping the relation among the classes keyid have

been used which holds the relation between files. In this case carSharing is a Facility so *Facility.csv* holds the key id of the *Price, Contact, Address, Location, Calendar, CalendarDates*. Each key id is unique (pandas indexes) so there wont be a conflict between the data in the integration phase.

- **CyclingPoint dataset** After reading dataset from previous section with the help of the geopy library we extracted informations such as **state, city, road, number, cap** extracted from location points. To achieve this we queried latitude and longitude of the point in the opendatamap api with geopy helper. After data extraction seven csv files **Facility, Price, Contact, Address, Location, Calendar, CalendarDates** generated to be exactly the same as formal schema. For keeping the relation among the classes keyid have been used which holds the relation between files. In this case CyclingPoint is a Facility so *Facility.csv* holds the key id of the *Price, Contact, Address, Location, Calendar, CalendarDates*. Each key id is unique (pandas indexes) so there wont be a conflict between the data in the integration phase.

- **CyclingRoute dataset** After reading dataset from previous section with the help of the geopy library we extracted informations such as **Startingpoint, ArivalPoint, Location information**. After data extraction ten csv files **Facility, Price, Contact, Address, Location, Calendar, CalendarDates, Route, TimeSpend, TimeStamp** generated to be exactly the same as formal schema. For keeping the relation among the classes keyid have been used which holds the relation between files. In this case CyclingRoute is a Route so *Route.csv* holds the key id of the *Facility, TimeSpend, ModeOfTransport*. Each key id is unique (pandas indexes) so there wont be a conflict between the data in the integration phase. Also Facility will holds the keyId of *Price, Contact, Address, Location, Calendar, CalendarDates*.

- **Accomodations datataset** After reading the exported **json** assured dataset from the previous step we extracted the useful information to fill with data the ontology. We extracted things like **Locations, Contacts, Addresses and Facilities**; we then saved those information into csv files to import in a easy way the transformed data into Karma Linker. We extracted the data also by assign unique key id for each entry. In this dataset with assign the value 6 to the facility type, which is the **Campsite**.

- **BikeSharing datataset** After reading the exported **json** assured dataset from the previous step we extracted the useful information to fill with data the ontology. We extracted things like **Locations, Contacts, Addresses and Facilities**; we then saved those information into csv files to import in a easy way the transformed data into Karma Linker. We extracted the data also by assign unique key id for each entry. In this dataset with assign the value 5 to the facility type, which is the **BikeSharing Park**.

- **Fuel Pumps datataset** After reading the exported **json** assured dataset from the previous step we extracted the useful information to fill with data the ontology. We extracted things like **Locations, Contacts, Addresses and Facilities**; we then saved those information into csv files to import in a easy way the transformed data into Karma Linker. We extracted the data also by assign unique key id for each entry. In this dataset with assign the value 1 to the facility type, which is the **FuelStation**.

- **Mountain Path datataset** After reading the exported **json** assured dataset from the previous step we extracted the useful information to fill with data the ontology. We extracted things like **Locations, Contacts, Addresses and Facilities**; we then saved those information into csv files to import in a easy way the transformed data into Karma Linker. We extracted the data also by assign unique key id for each entry.

- **Parking Areas datataset** After reading the exported **json** assured dataset from the previous step we extracted the useful information to fill with data the ontology. We extracted things like **Locations, Contacts,**

Addresses and Facilities; we then saved those information into csv files to import in a easy way the transformed data into Karma Linker. We extracted the data also by assign unique key id for each entry. In this dataset with assign the value 1 to the facility type, which is the **Parking Area**.

2.5.2 Data integration operations and tool

This section is dedicated to the description of the usage of the data integration tool that allows to map the datasets generated and well formatted in the previous phases, with the final ontology generated. The last datasets adaptation performed using the tool, as well as the mapping operation are here detailed.

2.5.2.1 Karma Linker We initially tried to install and use Karma Linker 2.4, but the **eml** publish option wasn't available, we then tried other versions of Karma Linker that we found in the wiki of the project in github. Without success in finding that option we tried to publish just the model and the rdf. Gathered the localhost url we configure: the firewall by opening the inbound and outbound port and the router to redirect the traffic to the computer we were using for making the **kos** application able to read the data. Without success and because it was requiring to much time to figure out it was not working we tried to run the Karma Linker on a server at this url: http://135.181.149.179:8080/. We tried by taking a look at the network tab in developers tool the returned code of the http request and it was 504 time out. We repeat the process and tried then to import into the dataset into **Kos application** by providing the a reachable url, but it didn't work again. We then found out the existence Karma Linker that was running on the university servers, we were than able to publish also the **eml**. We tried to import into Kos application by providing the url, but it didn't work due to a kind of parsing error.

2.5.2.2 Karma Linker model transformations In the integration of the data with ontology we have filtered data properly and now is aligned with schema. In each etype we need a unique uri for each model. We decided to use concatenation of the name of the Etype and row index is the best way. Later we will utilize this feature by connecting Etypes to each other with uri link. **ISSUE:** the biggest issue with karmalinker that we faced was understanding of the relation. As we have seen in tutorial videos this relation is defined by identifier but karma was unable to detect this relation through identifier so the relation between entities was unrecognizable for graghDB. When we analyzed we understood karma will assign unique identifier to each row even we define an incoming link for that instance. so two related instances had two diffrent key index instaed of having one as relation key. Generally if two have one way relation one will store the index of the other as value so it can find the instance with that key. But here we had two diffrent values instead of one for the relation key. As we can see in the picture below we used this configuration for relating address to location with identifier but it was unrecognizable.

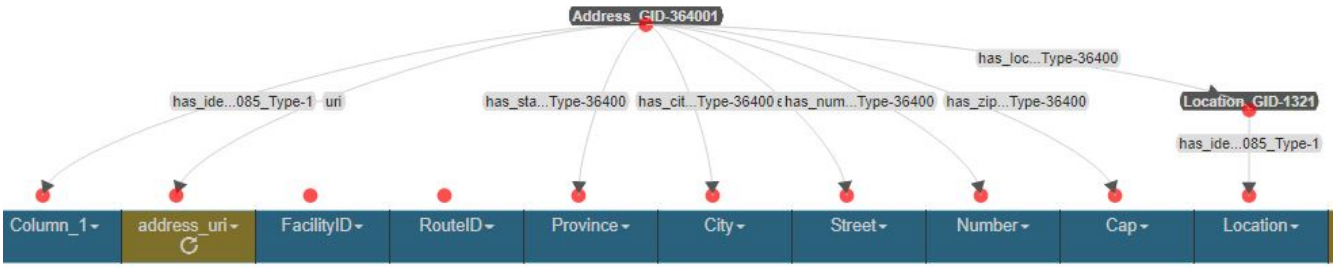


Figure 7: Karma relation through identifier

So we decided to use the URI instead for holding the relation. We already knew that each instances of data had unique row index so we assigned unique URI to each EType and then problem solved.

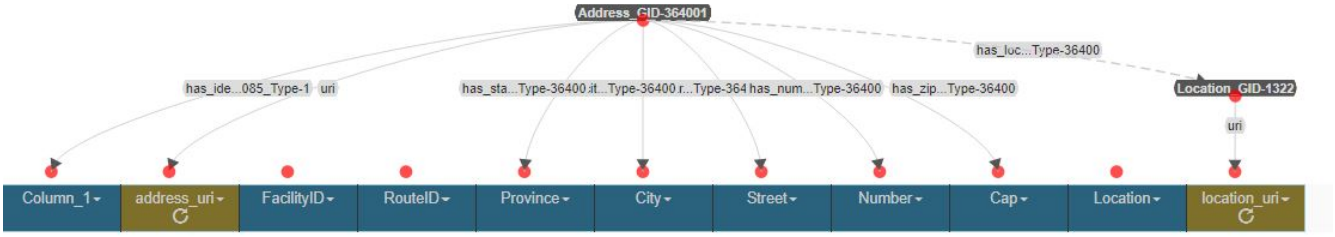


Figure 8: Karma relation through URI

As you can see we transformed *Column_1* because its a unique identifier to *EtypeName_identifierID*. Rest of the data is the same. Transformation only used for URI.

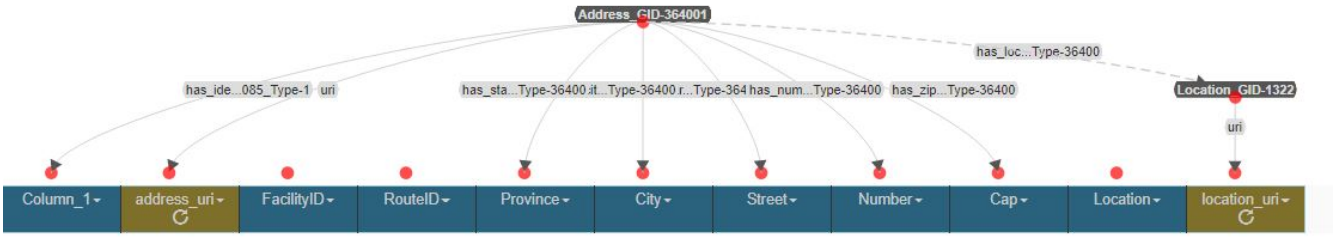


Figure 9: Karma relation through URI

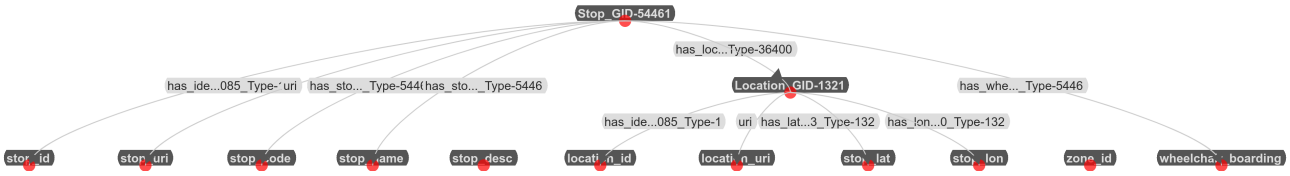


Figure 10: Karma relation through URI for the Stop

2.5.2.3 Solving questions and queries In this section we are going to answer some of the queries that have been defined.

- **query 1:1.11** List all the cycling starting and ending points closer than 10 Km to the San Bartolomeo train station.
  In this query firstly we go through Address to street and filter "San Bartolomeo", then address has location which in turn has latitude and longitude. This will return the coordinates of our target. Now we have to look within a 10 Km distance from the location to get all the routes. We go through road to location, location has latitude and longitude, then by filtering all points less than 10 Km distance from source location we will have 68 result.

Filter query results

⚠ Showing results from 1 to 68 of 68. Query took 0.2s, yesterday at 21:09.

	road	location	latitude	longitude	address	mylat	mylong
1	http://localhost:8	http://localhost:8	"46.035290875 26111"	"11.124597176977254"		"46.047565"	"11.1352074"
2	http://localhost:8	http://localhost:8	"46.035419235 46627"	"11.124589314736 71"		"46.047565"	"11.1352074"
3	http://localhost:8	http://localhost:8	"46.035127025 44549"	"11.122305754251887"		"46.047565"	"11.1352074"
4	http://localhost:8	http://localhost:8	"46.035608035 58216"	"11.122346220288074"		"46.047565"	"11.1352074"
5	http://localhost:8	http://localhost:8	"46.035608035 58216"	"11.122346220288074"		"46.047565"	"11.1352074"
6	http://localhost:8	http://localhost:8	"46.0351606198	"11.1222701405		"46.047565"	"11.1352074"

Figure 11: Query 1 Result

SPARQL Query & Update

Unnamed | Unnamed | Unnamed

```
prefix onto:<http://www.ontotext.com/>
prefix ontology:<http://knowdive.disi.unitn.it/etype#>
PREFIX omgeo:<http://www.ontotext.com/owlim/geo#>

select ?road ?location ?latitude ?longitude
where
{

    {
        #strting facelet for find sanbartalomeo


        select ?mylat ?mylong where {
            ?address ontology:has_street_address_GID-45807_Type-36400 ?street.
            FILTER regex(?street, "Malpensada").
            ?address ontology:has_location_GID-132_Type-36400 ?location.
            ?location ontology:has_latitude_GID-46263_Type-132 ?mylat.
            ?location ontology:has_longitude_GID-46270_Type-132 ?mylong.
        }
    }

    ?road ontology:has_location_GID-132_Type-138 ?location.
    ?location ontology:has_latitude_GID-46263_Type-132 ?latitude.
    ?location ontology:has_longitude_GID-46270_Type-132 ?longitude.
    FILTER( omgeo:distance(?mylat, ?mylong, ?latitude,?longitude) < 5).
}
```

Figure 12: Query 1 code

- **query 2:4.7** Give the closest available "car sharing" position to a specific point.
  In this query firstly we go through Address to street and filter the point specified by the user (e.g. "Malpensada"), then address has location and location has latitude and longitude. This will return the latitude and longitude of our target. Now we have to look within a 10KM distance for facilities with the enum of "CarSharingPark". We go through facility to address, facility has enum and street value, facility to address and address to latitude, longitude. Then by filtring all points less than 10KM distance from source location we will have 7 result.

Filter query results

⚠ Showing results from 1 to 7 of 7. Query took 0.1s, today at 01:45.

	street	latitude	longitude	enumvalue
1	"Via Giovanni Segantini"	"46.0746447"	"11.1217133"	"CarSharingPark"
2	"Via Santa Croce"	"46.0647501"	"11.1232546"	"CarSharingPark"
3	"Piazza Venezia"	"46.0695192"	"11.1276056"	"CarSharingPark"
4	"Piazza Dante"	"46.0712884"	"11.1213118"	"CarSharingPark"
5	"Via dei Solteri"	"46.08609955"	"11.121424367007315"	"CarSharingPark"
6	"Via Don Tommaso Dallafior"	"46.0654572"	"11.1543055"	"CarSharingPark"
7	"Largo Donatori volontari del	"46.0563978"	"11.131146627108679"	"CarSharingPark"

Figure 13: Query 2 result

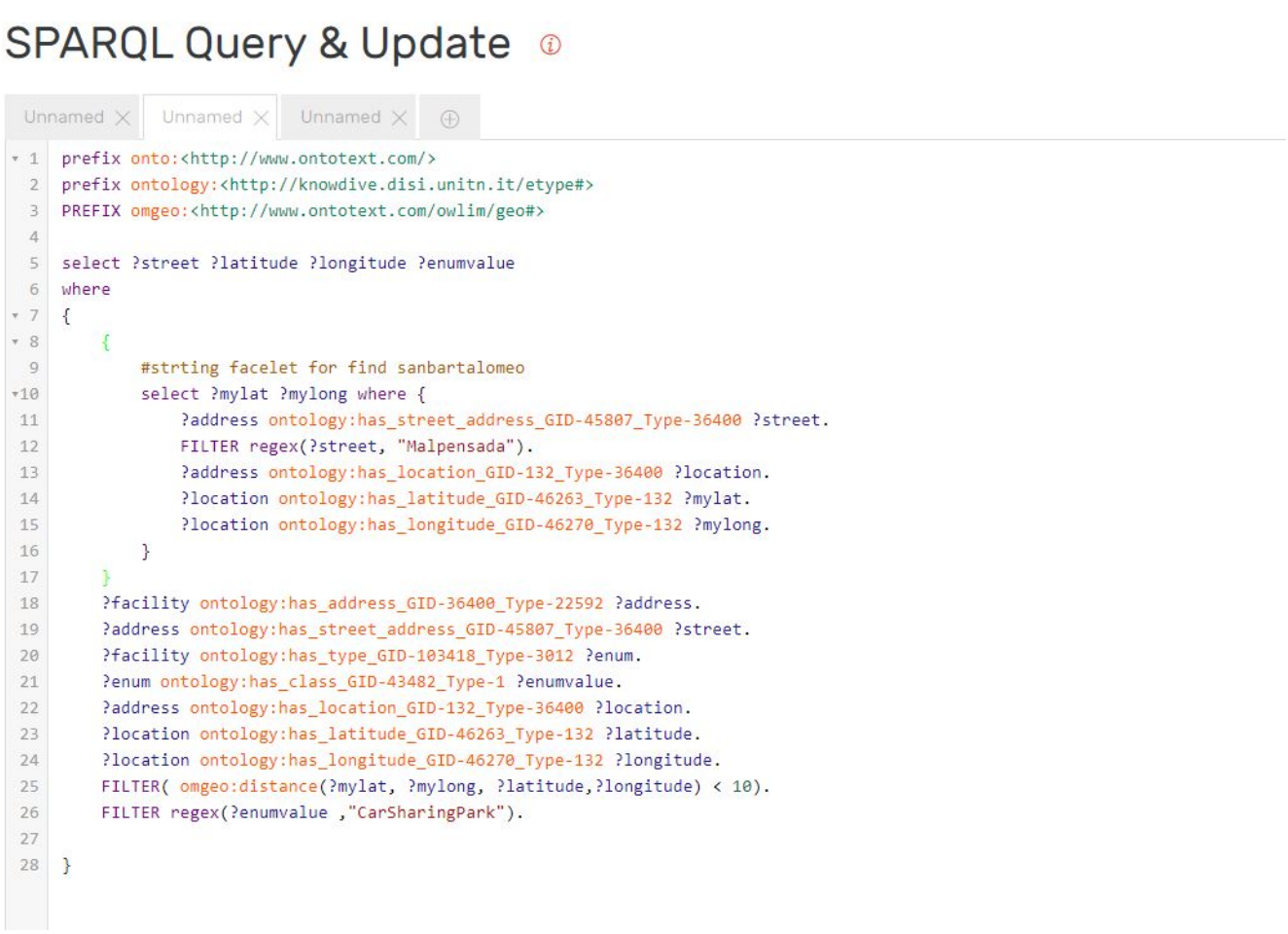


Figure 14: Query 2 code

- **query 3:2.6** Give the longest path in Trentino.
  In this query we returned all the roads, converted the data into a float type and sort the results in descending order by limiting to 1. We have done this approach because we weren't able to use a function like *max*.

```
# Returns the longest path in Trentino

prefix onto:<http://www.ontotext.com/>
prefix ontology:<http://knowdive.disi.unitn.it/etype#>
PREFIX omgeo:<http://www.ontotext.com/owlim/geo#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?road ?lengthF ?length
where
{
    ?road ontology:has_length_GID-28259_Type-22592 ?length .
    BIND(REPLACE(STR(?length), ",", "") AS ?lengthF) FILTER (xsd:float(?lengthF) > 1).
}
ORDER BY DESC(xsd:float(?lengthF)) LIMIT 1
```

Run

Table | Raw Response | Pivot Table | Google Chart | Download as

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, yesterday at 22:03.

	road	lengthF	length
1	http://localhost:8080/api/route_150691	"67300"	"67,300"

Figure 15: Query 3 result.

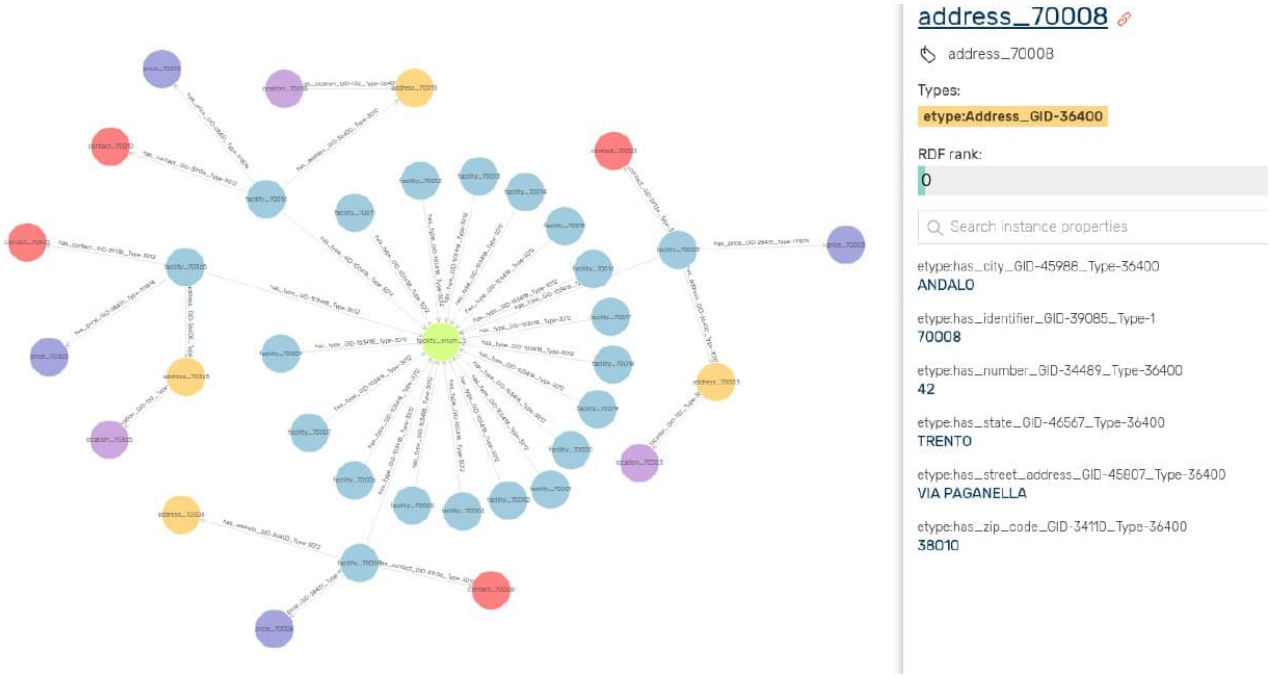


Figure 16: Graph Visualization. It is possible to see the relations between the facility and other entities. And Address is highlighted so that is possible to see the instance properties.

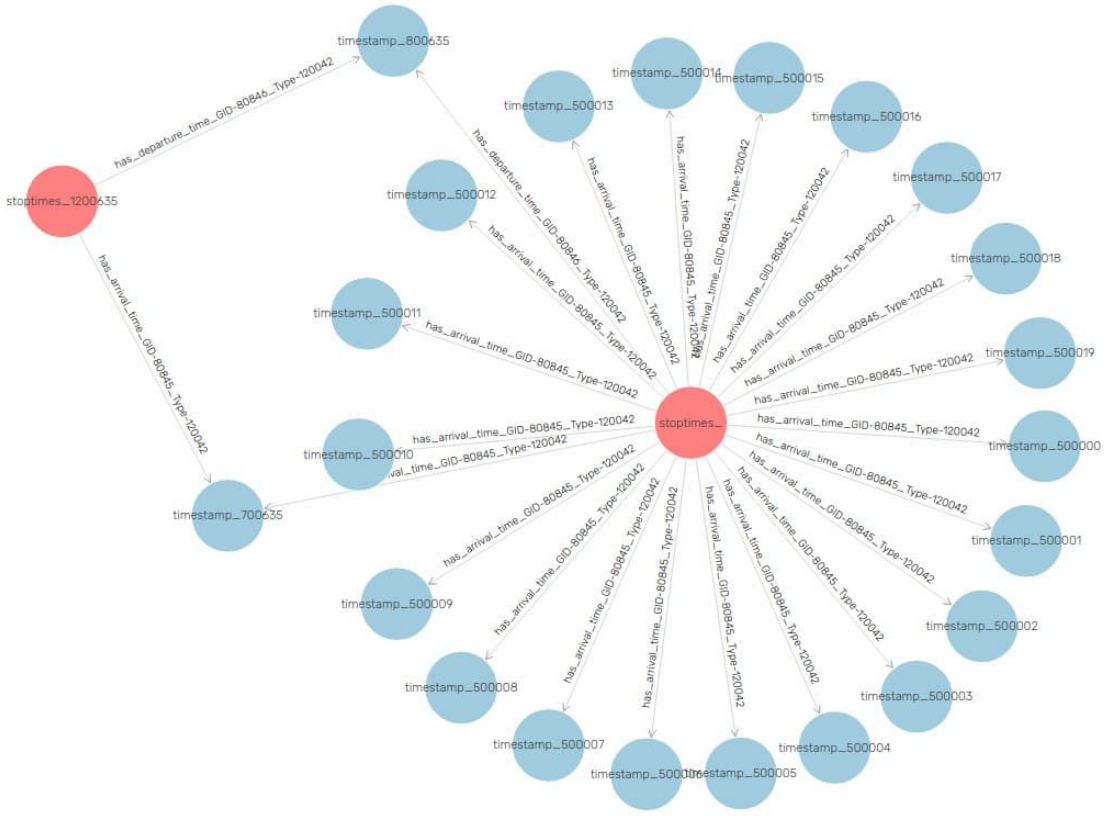


Figure 17: Graph Visualization of Stop.

It is possible to find the **KarmaLinker** printed models in the *Data Integration* dataset knwoledge subfolder. Moreover we exported also the RDF in turtle format and the EML.

2.5.3 Variance respect Formal Modeling datasets

In this section we are going to talk about the difference between data sets in data integration phase and Formal modeling phase. It is important to say that even if we searched a lot of datasets the majority of them had missing values or improper formats. We were able to mitigate the problem in the previous phases, but the datasets still need to be polished a bit more. For instance, if we say that an estimated values is fine, we could infer the missing information like we have done with prices in the accomodations datasets. Another thing it is that we were not able to translate every string location to actual longitude and latitude. As reminder, In Formal modeling after filtering data we aligned data attributes to formal schema one by one and we put all data in json format. To achieve this we have been defined models for each EType which contained the attributes of that. In this case we kept the hierarchy of the classes and also relations. At the end for each instance of the class we have data aligned properly with attributes.

In the data integration phase we notices karma linker has serious issue with json and doesn't load data in the current mode so we have to change the structure of the models and relations. To achieve this we inherited relational database idea by storing key-value as indexes and seperation data in diffrent CSV files. This is a more proper approach cause it is easier for karma linker to load file and apply model transformation to that. For instance

stazioni dataset will have these files seperated from each other **Address, Calendar, Contact, Facility, Enum, Location, Price**. In this case Address has Location, to seperate them we put uniqe key (row index of Location) in address, we dont need to keep index of the address an attribute in Location cause the relation is one way. We will do this approach to all datasets. Later this indexes will help us fix issue of importing data in the graphDB just with indexes.