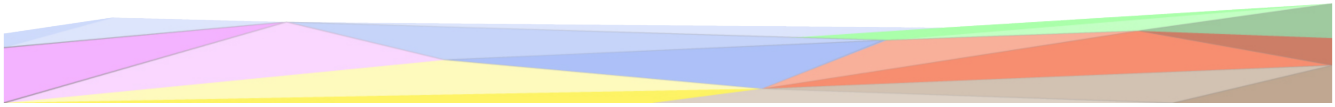
Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

Space Domain

Document Data:	Reference Persons:
16.12.2019	Alessandro Cacco, Sara Callaioli, Daniele Isoni, Michela Lorandi, Martina Battisti, Bertiana Balliu, Stefano Leonardi



Index:

Revision History:

Rev	Date	Author	Description of Changes
0	08.10.2019	Martina, Bertiana , Stefano	Initial personas definitions
1	18.10.2019	Martina, Bertiana, Stefano	Start adding the competency queries
2	30.10.2019	Martina, Bertiana, Stefano	Start formalizing competency queries
3	30.10.2019	Sara, Daniele, Michela	Related ontologies, start
4	7.11.2019	Martina, Bertiana, Stefano	Start creating EER
5	15.11.2019	All	Final EER completed
6	18.11.2019	All	Start creating ontology
7	5.12.2019	All	EER and ontology, final version of the model
8	9.12.2019	Alessandro	Introduction to Data integration
9	9.12.2019	Stefano, Michela	OOPS ontology evaluation
10	10.12.2019	Alessandro, Sara, Daniele, Michela	Dataset selection
11	11.12.2019	Alessandro, Sara, Daniele, Michela	Dataset mapping in karma
12	12.12.2019	Alessandro, Sara, Daniele, Michela	Dataset cleaning and merging
13	13.12.2019	Martina, Stefano, Alessandro	Conclusions and final considerations for both modeling and data integration
14	14.12.2019	Alessandro, Sara, Daniele, Michela	Dataset mapping in karma, final version
15	15.12.2019	Sara, Michela, Daniele	Related ontologies, final version
15	16.12.2019	Alessandro, Sara, Michela	Review and final corrections

1 Scenario Description

Nowadays people spent their free time travelling around. In our frantic world, time is running short and people want to improve the quality of their trips. Let us explore some possible personas.

Maria is a young woman; she is still studying at the university and next year she will graduate. She has free weekends because lessons end on Thursday morning. She travels with her friends who are students at the university like Maria. Maria plan one trip per month, she uses public transport, and she prefers to spend as little money as possible. As lessons start on Monday morning, she has only the weekend available, and she looks for accommodation for a maximum of 3 nights. She is really interested in visiting cities, the most popular attractions and museums.

Giovanni is a husband and a father of two young children. He usually travels with his family for two different weekly trips a year, and occasionally they take weekend trips during the year. Travelling with the family requires planning the whole trip to avoid problems. As Giovanni is a precise person, he takes care of the whole organization. He and his wife love to practice sports in the visiting areas, but at the same time, they look for attractions, like adventures parks, suitable for their 7 and 11 years old children. The budget is high because both Giovanni and his wife have a substantial salary: they are lawyers. Giovanni has a beautiful car, so he wants to travel in comfort using it.

Luca is a young, single, 30 years old boy. He always travels with his camper and his dog. He is an adventurer and loves nature. He is looking for wild places. Luca is a fan of trekking, which can also last several days with an overnight stay in a tent or in a lodge. When he goes on vacation, he seeks for peace and adventure. The camper needs a place to park and Luca looks for a place to camp. Having a humble job, he tries to spend as little as possible. He can go on vacation only on weekends because his work does not allow him more free time.

Monica, wife, mother of 3 children, and grandmother of 7 grandchildren. A very busy woman with household chores and children to look after. As after retiring some ailments appeared, Monica and her husband decided to take some breaks from their hectic life. Their favourite destinations are the SPAs where they can relax and benefit from the treatments. Monica prefers to travel by train as far as possible and then by taxi to reach the destination. She likes walks but due to her age, she can't do difficult trekking.

1.1 Storytelling definition

Trips organization is a complex problem, which people don't want to spend time on. The solution we propose is an integration of data regarding places, famous attractions, itineraries and any kind of points of interest. In order to obtain a complete data collection, we intend to integrate data about facilities like hospitals, shops, and public structures, and transports. It is important to adapt the trip to your needs in order to let you better enjoy the experience. Crossing data, it is possible to plan the whole vacation with no effort up to the smallest detail.

Maria is 25 years old, she's a student. She is studying law in Milan at university. During her studies, she met many friends who share her passion for travel. So they decided to take some vacation together. Maria loves travelling across Italy because she loves to discover all the cities and their distinctiveness. With the arrival of autumn, with its suggestive colours, the girl would like to travel to Northern Italy. As she has never been to Trentino, the next trip will be there. Due to university lectures, Maria has no much midweek free time. She can only travel during the weekend from Thursday to Sunday. As she has no car available, she travels by public transport in order to spend as little as possible. Sometimes, if necessary, she moves by foot or by bicycle. She takes into account the eco-sustainability of her journey. So, she does not mind giving up comfort. She also believes that by travelling on foot she can best enjoy the trip by paying more attention to the details of each place. As far as housing is concerned, Maria is satisfied with low-priced accommodations. She looks for a place to sleep for a maximum of 3

nights. She is very interested in visiting cities, pay attention to the culture, the peculiarities and the characteristic traits of each of them. She is passionate about art and architecture. Every time she visits a place, she doesn't want to miss the best attractions and the museums.

Giovanni is a 45 years old legal professional. He lives and works in Munich with his wife, a legal professional as well. They have two young children, 7 and 11 years old. Since the children still go to school, during the academic year they have free weekends and some national holidays where they have free time to go on vacation with their parents. Giovanni and his wife have no problems in taking days off during the whole year. The family of Giovanni travels for two different long trips a year: a two-week holiday in summer and another two-week holiday in winter. Occasionally, during the national holidays or weekends, the couple organizes short trips of 2 up to 5 days. Giovanni loves skiing, climbing and biking, but his wife prefers different sports like hiking and trekking. Often they choose to get to Trentino to pass their vacations, as they live not so far away from Italy. Owning a luxury car, they willingly travel using it to get to the destination. During their long summer trips, they love to spent time on Garda Lake to relax and occasionally practising sports: they bike, hike and go to the lake for some sun and lake bathing. Since Trentino offers many trekking routes, suitable for children as well, Giovanni and his wife organize part of the holiday dedicated to this type of activity. Their children love this time of the holiday. Their winter vacations usually are spent on snow area zones of Trentino where they all can ski, snowboard or ice-skate and take walks with snowshoes. During their long trips, the family of Giovanni prefers small, but very nice high-rated hotels. While during their short trips they prefer bed and breakfast option because they can have flexible timetables.

Luca is 30 years old man; he works in Verona as a provincial assistant officer, a humble job that forces him to look for a not too expensive vacation. Luckily, this is not a problem for him, as he can take a break from the work to live adventures. His favourite places are not exotic nor trivial ones, but in contact with nature, in wild area together with his dog, a setter puppy called Sky. To reach the locations he uses his camper that is well equipped with everything he needs. If something important is missing, Luca always looks for shops that can provide him with the missing element and other useful tools. He prefers finding these locations on the way and doesn't add useless travel time to the trip. Due to the work, he travels during the weekend, but his goal is to seek quite. That's why he prefers to travel during the off-spring like in autumn or in the early spring, far from summer, Easter or Christmas vacations. Depending on the period and the location, he goes outside for hiking, ski mountaineering, or canoeing. He is a person who accommodates himself smoothly to any condition. So, he can fulfil his desire to try new sports and activities like biking or cross country skiing, if suggested for the area.
The vacation of Luca always starts by driving the camper. After reaching the place and park it, its time for a reconnaissance tour to get in touch with the nearby area and begin to know local people. He loves travelling alone because its the perfect way to approach the local population or other unconventional tourists. The core of the holiday can be a set of one-day excursions, but also longer ones that require sleeping in a tent or a lodge for a night or more. He likes circular, but especially crossing trips, that often require public transport to get back to the original location.
At the moment, Luca is interested in some hiking trips in Trentino, like the trans-Lagorai, or climbing with a Ferrata the Marmolada, but also the circular hike of Cima d'Asta. He knows that all these paths are well labelled from the SAT so there is no danger but the adventure its guarantee.

Monica is a wife, a mother, and a grandmother. She is 70 years old. As a grandmother, she is busy watching after her grandchildren. She's very absorbed in with household too. Although she is very active and helpful, she feels the need to have a relaxing time with her husband. Trentino Alto-Adige has been chosen because, in addition to having numerous hotels with SPAs, it also has numerous specialized nursing homes. Monica and her husband love thermal baths and their treatments. The culture of Trentino Alto-Adige includes wellness as a key moment of

the day, with hay baths, mud baths and thermal waters, the place offers to pamper to visitors. Between one sauna and another, Monica and her husband can benefit from the care available in the nearby facilities. The elderly couple chooses to travel by public transport, which provides excellent and punctual services throughout the territory. The couple also loves taking walks in the countryside, nothing challenging to reconcile age. Among the fresh woods, they can recover from the scorching heat of the cities. Monica's husband, passionate about photography, can take advantage to take great photographs with breathtaking views. Therefore, Trentino lends itself as a perfect destination to recharge your batteries and regenerate back to the hectic life of all time.

1.2 Personas description

NAME	AGE	INTEREST	USAGE	DESCRIPTION
Maria (1)	25	Visit cities, popular attractions and museums.	Travel more and spent less. Trips focus on culture.	Maria is a student. She wants to use the system to organize her trips. She wants to spend as little as possible but at the same time, she wants to visit many places. Using the system could be an optimal solution, crossing the data and prices you can get a list of accommodation and facilities with the most advantageous price. Organizing a trip by public transport requires a huge amount of data to find the best connections, the proposed system will also provide such data.
Giovanni(2)	45	Enjoy holidays with his family, Skiing on winter and biking/hiking/trekking on summer.	Organize the trip, suitable sports with different levels of challenge. Parking lots (accommodation).	Giovanni works as a professional and goes on holidays with his family. He wants to use the system to organize their vacations in Trentino. Their holidays differ in length, type of sport and the level of challenge of the sports they practice. So, for each of their holidays, they have different needs about the type of staying structure. They travel with their own vehicle.
Luca (3)	30	Explore, travel and discover new emotions.	Adrenaline-filled holidays. Sport and entertainment in the first place.	Luca is a disorganized person, that will use the system for planning his list of points of interest at his best. He cares about a few spots where to stop in case of emergency, due to the weather condition or his disorganization. He is also interested in points of interest all around the path chosen that he will eventually cross. Besides, he wants to know if there are special activities, proposed in the area, to improve his adventure. To enjoy these activities he chooses the affordable ones, and those accessible to his dog Sky.
Monica (4)	70	Relax with her husband.	Best hotels with spas. Means of transports timetables.	Monica is a full-time grandmother. She wants to relax and take care of herself and her husband. Alto Adige is a perfect destination for numerous wellness centres and specialized nursing homes. Being older people, an organization is needed with a particular focus on public transport. The system can provide the timetables of all means of transports available. The treatments must be booked and for this is necessary planning of the schedules compatible with the organization of the transports.

2 Queries description

2.1 Competency queries

In order to create the competency queries, after the scenario and personas definition, we have tried to imagine what people could ask to the system. Using even the datasets selected by the data integration sub-team, we have formalized a complete list of competency queries. The actions described in the storytelling are the answer to the competency question we have modelled. So, the actions are mapped to the competency questions.
Given the application scenario, the set of queries aroused committing to our specific model. We have manually processed the competency questions to extract single lexical items, then we have grouped them and associated each entity type to a set of attribute terms. We have created a second table (Table 2) which reports how queries can be generalized from the actions in the plot above.

PERS	NUM	QUESTION	ACTION
Maria	1.1	Give the list of hotels near the station of Bolzano	The system search all the hotels within 5 km near the station of the city and returns all the fields.
Maria	1.2	Give all trains from the station of Bolzano to the station of Trento	Select the station of departure in Bolzano and all the trains available to reach the station of Trento will be provided with timetables.
Maria	1.3	Give the list of museums of Bolzano open on Sunday	A list with distances and other info will be provided (description).
Maria	1.4	Give the religious attraction timetable of Merano	In order not to lose the best monuments and churches of the place, the structure with the relative timetable will be provided.
Maria	1.5	Give all means of transport that pass in "Borino" stop In Trento	Select all the means of transport available to reach "Borino" stop in Trento.
Maria	1.6	Give the list of cities with more parks	Alto Adige is also known for its flower garden. To observe all the colorful flowers and the beautiful parks it is necessary to obtain, from the system, a list of parks and public gardens.
Maria	1.7	Give the list of cities with cheapest restaurant	The system will output the city with many restaurants with lower prices.
Maria	1.8	Select Hotels in Bolzano where you can check-in at 14.30 o'clock	Select all the hotels, in the city you want, when the check in hour is 14.30 o'clock.
Maria	1.9	Give the list of local traditional shops in Alto Adige	In order to buy something for the parents, she is looking for all traditional products.
Maria	1.10	Give the attractions at most 10km away from Cles	Search for the attractions in order to not do a trip of more than 10 km away from Cles.
Maria	1.11	Give the cities at most 1 km away from Villa Lagarina	Search for the cities in order to not do a trip of more than 1 km away from Villa Lagarina.

Maria	1.12	Give all skating rings open on Monday evening, ordered by proximity to the near Restaurant Anaunia in Romeno	Search for the nearest skating rings, open on Monday evening, near the Restaurant Anaunia in Romeno.
Maria	1.13	Give me the shopping centres with high rates	If the weather is not the best and therapeutic shopping is needed, the best shopping centres will be displayed.
Maria	1.14	Give the activity path near Waldheim Hotel in Ruffrè	If you want to make some sport activity you are looking for an activity path near a specific place, in this case near the Hotel Waldheim.
Maria	1.15	Give the cinema's of Bressanone or within 5 km from Hotel Gasser	Return the complete list of cinema of a specific place
Maria	1.16	Give the libraries and bookstores open at 14:00 on Saturday	Return the complete list of libraries and bookstores open at 14:00 on Saturday.
Maria	1.17	Give the position of a bike-shop	Moving without a car is very difficult but for small journeys you can use to rent bikes or e-bikes or buy specific equipment.
Maria	1.18	Give the list of visitable castles of South Tyrol reachable by means of transport	Return the list to public castles that can be reached by means of transport.
Maria	1.19	Give the emergency medical Service available for a first-aid emergency	In case of first-aid emergency the hospitals will be returned.
Maria	1.20	Get the most famous religious attractions in all South Tyrol with relative timetables (the most rated religious attractions with the best rating)	Return the list of unmissable religious attractions.
Maria	1.21	Get the easiest mountain path with average time between 1 and 2 hours	Return the easiest mountain path with average time between 1 and 2 hours.
Maria	1.22	Get the easiest mountain path without equipment required	Return all mountain path which can be practiced without equipment and which are easy.
Maria	1.23	Get the Pharmacy near the Bolzanos' cathedral in duty	Return the nearest Pharmacy in duty in case of emergency.
Maria	1.24	Get the contact of the 10 most well-voted hotel	Select the 10 hotels well rated and return all contacts of them.
Maria	1.25	Get all restaurant of Appiano where you can eat without a reservation	Select from the list of all restaurants of Appiano those where you can eat without a reservation.

Giovanni	2.1	Give the list of family accommodations near Garda Lake Trentino having parking	Extracts and returns all the accommodations within 15 km from municipality Riva del Garda that have four or more NumOfBeds and have parking.
Giovanni	2.2	Give the list of all cultural options in Trento, Rovereto, Arco, Riva d/Garda for the week 23/12/2019 - 29/12/2019	Extract and returns all attractions of type "culture", "NightlifeEntertainment" ocurring within 5 km from municipality of Trento, Rovereto, Arco and Riva del Garda; scheduled for the required period 23/12/2019 - 29/12/2019.
Giovanni	2.3	Give the list of smooth biking paths activity options close to Dro and upcoming for the next three days from 15/12/2019	From attractions of type "SportLeisure", select those with activityPath-s within 5km of distance from municipality of Dro, having the SuggestedType "bike" from activityPath, having difficulty Low (L), and scheduled in the next three days after 15/12/2019.
Giovanni	2.4	Give the list of natural climbing areas of Trentino	Extract and returns all points of interest in province of Trento being attractions of type SportLeisure with ActivityPath-s of Medium and High difficulty, PositiveGradient of more than 25%, and suggestedType: walk or other.
Giovanni	2.5	Give the list of Accommodation near climbing points in the area of Cadine	Extract and returns all Accommodations within 10km from municipality of Cadine, where exist attractions of type SportLeisure with ActivityPath-s of Medium and High difficulty, PositiveGradient of more than 25%, and suggestedType: walk or other.
Giovanni	2.6	Give the list of the sporting shops near the climbing areas in Arco	From the list of point of interest within 10 km from the municipality of Arco extracts and returns all attractions Shop of type SportsEquipment, SportsFacilities
Giovanni	2.7	Give the list of all important turistic infoPoint of provinces of Trento and Bolzano	Extracts all municipalities of province of Trento and Bolzano with more than 15'000 inhabitants and returns their list with the respective addresses of the municipalities, ordered by the number of inhabitants
Giovanni	2.8	Give the list of restaurants in Molveno near walking paths	From the list of PointOfInterest in Municipality of Molveno, with the SuggestedType "Walk" from the ActivityPath returns all restaurants within 2km from stops of the paths
Giovanni	2.9	Give the list of smooth trekking paths	Extracts and returns all ActivityPath of SuggestedType "Walk" where Difficulty is "Low" and the Length is up to 5km
Giovanni	2.10	Give the list of groups organizing medium-to-high difficulty trekking activities	Extracts from ActivityPath of Difficulty High and Medium all SuggestedType Walk and returns info about the Address, Description and contacts

Giovanni	2.11	Give the list of mountain biking paths in province of Bolzano	Extracts and returns all ActivityPath in province of Bolzano with SuggestedType Bike of Difficulty High and positiveGradient higher than 20/30%
Giovanni	2.12	Give all the skiing centers of Trentino	Extracts and returns all ActivityPath-s of Province of Trento, with SuggestedType Skiing, with respective Municipality
Giovanni	2.13	Give the skiing centers of Trentino with accommodation	Extracts all ActivityPath-s of Province of Trento, with SuggestedType Skiing, with respective Municipality, having PointOfInterest Accommodation and returns their addresses, description and contact.
Giovanni	2.14	Give the skiing centers in Alto Adige having accommodation with SPA and opened in February 2020	From Extracts all ActivityPath-s of Province Bolzano, with SuggestedType Skiing, with respective Municipality, having PointOfInterest Accommodation with Wellness, having Timetable with openings Scheduled from 01/02/2020 to 29/02/2020.
Giovanni	2.15	Give the skiing centers in val di Fiemme with in-day-for-children activities	From Extracts all ActivityPath-s of all municipalities within 50 km from municipality of Cavalese, with SuggestedType Skiing, having children-related activities.
Giovanni	2.16	Give the sailing centers of Riva Del Garda and Torbole	Extracts all the Attractions of municipality Of Riva del Garda and Torbole of type SportLeisure, having sailing activities and returns Address, Description and Contact.
Giovanni	2.17	Give the bike rental shops in Trentino	Extracts and returns all Shops of type Bikes, SportsFacilities, and SportsEquipement of all municipalities of Province of Trento.
Giovanni	2.18	Give options of jaunts of Trentino Garda Lake	Extracts and returns all ActivityPath-s, of SuggestedType Walk, starting within 5km from Riva del Garda or Torbole
Giovanni	2.19	Give the skiing centers with accommodation and SPA centers	From the list of skiing centers with accommodation extracts and returns those offering SPA specifying types of SPA
Giovanni	2.20	Give traditional restaurant near Vipiteno	Extracts and returns all restaurants within 15km from municipality of Vipiteno
Giovanni	2.21	Give challenging options of sports in Trentino Alto Adige	Select and return all ActivityPath-s in the province of Trento and province of Bolzano, of SuggestedType Walk, Bike or Other, with Difficulty High
Giovanni	2.22	Give skating centers in Alto Adige	Returns all activityPath-s of province of Bolzano of altitude higher than 1500m with suggestedType Skating and equipmentRequired

Giovanni	2.23	Give Snowshoes activities options in Trentino	Extracts from all municipalities of province of Trento the ones with altitude higher than 1500m and returns all ActivityPath-s with suggestedType Skiing or Walk and equipmentRequired present.
Giovanni	2.24	Give all swimming pools in TAA opened from 01/11/2019 to 29/02/2020	Extracts all Attractions of Type SportLeisure, being/having swimming pool with Timetable openings scheduled for the period 01/11/2019 -29/02/2020, and returns their Addresses and Contacts.
Giovanni	2.25	Emergency medical of municipality of Comano Terme	From the point of municipality of Comano Terme, selects and returns all emergency medical points within 10km from municipality of Comano Terme
Luca	3.1	Give me a list of the main cities to cross from the central station of Verona to Pozza di Fassa	Connect the roads from the station of Verona to Pozza di Fassa and then return a list of the main cities on this path.
Luca	3.2	Give me the expected time to travel up to Pozza di Fassa, passing through the main cities computed	Compute the travel time through all the points, previously computed.
Luca	3.3	Give me the list of tools stores on the planned route from Cembra to Predazzo	For each point of route, check the presence of tools stores near it. Return a list of all the shops found.
Luca	3.4	Give me the fuel stations along the planned road from Cembra to Predazzo	For each point of route, check the presence of fuel stations near it. Return a list of all the stations found.
Luca	3.5	Check the price of the camping Daino (Pietramurata)	Provide the accommodations according to the city Pietramurata provided in the address, the type Camping and the name Daino, and leave the user free to choose if there are more choices. Then provide the price
Luca	3.6	Give me the number of spots existing in the camping Daino (Pietramurata)	Ones selected the camping, retrieve the number of spots. If this data does not exist provide all the rules (description) of the camping
Luca	3.7	Give me the 5 closest cross country ski stations to the Predazzo municipality	Compute the distance as the crow flies from Predazzo. Then, sort and filter according to this parameter.
Luca	3.8	Give me a list of the cycle paths start from the Predazzo municipality	Select all the cycle paths than has as starting point the municipality of Predazzo.
Luca	3.9	Give me the best ranked bike shop on the cycle path from Predazzo to Moena (the two municipalities location)	Filter the cycle path followed by Luca, then for each intermediate point check the presence of bike shops near it. Choose the closer one and provide it.

Luca	3.10	Give me the camping close to Predazzo municipality, that are open in November	Select all the camping according to the distance from Predazzo (less than 30km), and provide the only ones that are open in November.
Luca	3.11	Give me the camping near Pozza di Fassa (Sèn Jan municipality) open in November with a price lower than 80€ per night	Select according to the location of the camping (force them to be near Sèn Jan municipality), and select the ones open in November with a price lower than 80€ in November.
Luca	3.12	Give me the opening days of the camping Daino (Pietramurata)	Select the camping and provide the schedule of the opening days along the year.
Luca	3.13	Give me the list of bars open evening of the 12 November, close to the camping Daino (Pietramurata)	Select the restaurants with type "bar" and in the city of "Pietramurata", then for each of them provide the ones that are open on the given day.
Luca	3.14	Give me the list of hikes starting from the municipality of Moena	Select the walking routes (not only mountain paths) with the intermediate point of Moena.
Luca	3.15	Give me the list of hikes that start from a location, that I can reach in less than an hour driving, from Moena's municipality	Select the walking routes (not only mountain paths) with at least a point with an euclidean distance lower than k kilometers.
Luca	3.16	Give me the points of interest 10km away from the cable way of Passo Pordoi. Distance computed as the crow flies	Select the points of interest with an euclidean distance less than 10 km, respect to the selected position.
Luca	3.17	Give me 5 points of naturalistic interest, near the cable way of Passo Pordoi, excluding the 10 most popular ones	Select the points of naturalistic interest (identified by using the description tags) with an euclidean distance respect to the cable way, then sort them according to that distance. Finally, remove the most popular ones.
Luca	3.18	Give me a list of activities, in the Sèn Jan municipality, that allow me to try seasonal sports	For each facility, near the Sèn Jan municipality, check if it has tags similar to seasonal sports.
Luca	3.19	Check if there is at least one structure that can rent me the equipment for practice ski	Select in all the area the shops labelled as rental_shops, and choose the ones that has the tag: equipment for ski.
Luca	3.20	Give me the price of the nearest ski station respect to the ski rent shop in Pozza di Fassa called Gross Sport	Select all the facilities labelled as ski station, and choose the one with the minimum distance from the given rent shop.
Luca	3.21	Give me the list of lodges open the weekend of the 12 November in Trentino	Select the lodges in Trentino and check which ones are open during the given date.

Luca	3.22	Give me the mountain paths that start from the lodge Ciampedìe	Select all the routes that have Ciampedìe lodge, as one of the points through which it passes.
Luca	3.23	Give me the mountain paths that go from the lodge Roda di Vael to the lodge 'Aleardo Fonza' alle coronelle	Select all the paths that has as starting point the lodge Roda di Vael and the lodge 'Aleardo Fonza' alle coronelle as ending point.
Luca	3.24	Give me the difficulty of the mountain path number 546 going uphill, and also the expected travelling time	Extract from the onward path called E546 the expected travel time and the difficulty.
Luca	3.25	Show me on a the planimetry of the mountain path number 546	Select all the points of the path E546 and retrieve the relative distance and positive/negative gradient between them.
Monica	4.1	Give me the open nursing homes in Bolzano	Select the complete list of nursing home available at the moment, in the area of Bozano.
Monica	4.2	Give me the doctor near Twenty Shopping Center in Bolzano	In case of emergency it is important to know the doctor for tourists near Twenty Shopping Center in Bolzano.
Monica	4.3	Give me the list of the cheapest and most rated spas in Alto Adige	Select the best cheapest spas in Alto Adige with a suggestive view.
Monica	4.4	Give me the bus station near the restaurants of Trento	Select the stop near restaurants of Trento to facilitate travel by public transport.
Monica	4.5	Give me the address of the thermal centres	For a day of thermal return the thermal centers' addresses.
Monica	4.6	Give me the list of trekking paths ordered by increased difficulty and time	Returns the list of practicable marked routes with relative difficulty included.
Monica	4.7	Give me all info points	Select all the info points to get information quickly on the territory.
Monica	4.8	Give me all structure that have wellness treatments near Merano available	Select all structures with treatments near Merano.
Monica	4.9	Give me all paths in Merano with no equipment required, ordered by difficulty	Select all paths in Merano, ordered by difficulty.
Monica	4.10	Select a list of hotels filtered out with respect to the user evaluations	Select hotels, then sort based on user ranking and retrieve the top 20.
Monica	4.11	Construct a route from Trento up to Bolzano, with public transports without walk path in the middle	Build a path from Trento up to Bolzano using only transportation and without connecting path by walking.

Monica	4.12	Give me the location of an info point that organize a guided tour through Bolzano	Select the info points in Bolzano, and retrieve the ones that has the tag: guided_tour.
Monica	4.13	Give me the attractions open in Bolzano in the early morning	Select the attractions in Bolzano that are open before 10.
Monica	4.14	Give me a list of restaurants that offers, in the menu, local specialities	Select the restaurants with the serves cousine: Lokale and retrieve their full description.
Monica	4.15	Give me simple trekking walk ending to a restaurant, in no more than 3 hours	Select all the paths (with a partial length of 3 hours) starting from Bolzano and check which of them pass near a restaurant.
Monica	4.16	Give me a simple trekking path that connects the Parco Petrarca in Bolzano and a bus station, that can bring me back to the starting point	Select the trekking paths with endpoints the Parco Petrarca and a bus stop, then check if it is possible to close the loop with a bus connection.
Monica	4.17	Check if there is a guided tour to the museum of Firmiano castle, in Bolzano	Select the attractions related to the "Firmiano castle", and retrieve the ones with the tag: guided_tour.
Monica	4.18	Give me the timetable of the Funivia del Colle in Bolzano	Select the Funivia del Colle in Bolzano and retrieve its timetable.
Monica	4.19	Give five easy-to-medium-difficulty alternatives to have a walk near Rovereto, reachable by public transportation	Get the list of all ActivityPath within 15km from municipality of Rovereto, having SuggestedType "Walk" with positiveGradient/negativeGradient 0 - 15%, select those with starting/ending within 500-1000m from a public transportation point, return the five most popular ones according to reviews
Monica	4.20	Give easy paths passing by shops with local products in Alto Adige	Selects all ActivePath-s of Alto Adige including ShopType of "LocalProducts" with negativeGradient/positiveGradient 0- 15%, difficulty "Low", with length max 5km, returns the list ordered by best reviews
Monica	4.21	Give all the mountain points easy-to-walk of Alto Adige reachable by public transport	Select all ActivityPath-s of province "Alto Adige", with SuggestedType "Walk", Difficulty "Low", within 500m of distance from a point "Stop", having altitude higher than 1500m.
Monica	4.22	Give ten crafting options of Alto Adige, reachable by public transport	From the PointOfInterest "Shop" type "Handicrafts" of Province "BZ", reachable by public transport within 500m of distance from a point "Stop" of transportation service, select top 10 as per user reviews.

Monica	4.23	Give accommodations in Alto Adige with SPAs being near a photo-related shop	From all accommodations with Wellness, in province "BZ", select the ones within 5 km from a ShopType "OpticianPhotography".
Monica	4.24	Give accommodations in Trentino near children playground reachable by public transportation	Extracts and returns accommodations in the province of Trento having playground within 1000m of distance, and within 1000m from a point or Stop of public transportation
Monica	4.25	Give all medium-priced restaurant in Brixen that can be reserved	Extracts and returns all PointOfInterest of Municipality "Brixen", being "Restaurant"-s with Price "$$", having AcceptReservations present.

This table is used to link the competency queries to the classes of the EER schema. The last two columns respectively state for each query the classes used and its attributes. Instead, the first one contains the set of representative numbers: the number of the persona in bold, separated by a comma, and then the number of the single queries, separated by a dash.

NUM	TYPES	PROPERTIES
2:1-13-19, **4:**1-23-24	Generic (Accommodation)	type
1:1-24, **2:**1, **4:**3-10-23-24	Hotel (Accommodation)	price, stars, parking, number of beds, wellness
2:8, **3:**21-22-23	Lodge (Accommodation)	-
2:6, **3:**5-6-10-11-12	Camping (Accommodation)	price, parking, number of spots
1:3-4-6-8-9-10-12-15-18-20, **2:**2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-18-19-21-22-23-24, **3:**7-18-20, **4:**8-9-13-16-17-18-19-20-21-22-24	Attraction	type, theme type
1:7-25, **2:**20, **3:**13, **4:**14-15-21-22	Restaurant	price, reservation
1:1-3-4-5-6-7-10-11-15-16-17-19-20-23-25, **2:**1-2-3-5-6-7-8-10-12-13-14-15-16-17-19-20-22-23-24-25, **3:**3-4-5-6-7-8-9-11-13-14-18-20-21, **4:**1-5-19-21-22-23-24-25	Address	province, city, village, street, number, CAP
1:13-16-17-23, **2:**17, **3:**3-4-9-19, **4:**7-12-25	Shop	price, type, offer
1:2-5-14-18-21-22, **2:**3-4-6-8-9-10-11-18-21-23, **3:**1-2-3-4-8-9-14-15-22-23-24-25, **4:**6-11-15-16-19-20-21-22-24	Activity Path	difficulty, positive gradient, negative gradient, equipment required, suggested type, avg travel time, length, start point(from), end point(to)
1:19, **2:**25, **4:**2	Emergency Service	type, grade of emergency

1:2-5-8-10-11-14-19-23, **2:**3-4-6-8-9-11-18-21-23, **3:**1-7-15-16-17, **4:**2-4-5-7-11-15-19-20	Point	altitude, latitude, longitude
1:2-8-12-19-20-23, **2:**2-3, **3:**10-11-12-13, **4:**9-13-18	Timetable	schedule 1/2, main vacation[ ]
1:3-4-6-9-11-15-18, **2:**1-7-12-13-14-15-16-17-19-20-22-24-25, **3:**16-17, **4:**21-22-23-24-25	Point Of Interest	name
1:3-13-20-24, **3:**6-9-17, **4:**3-8-10-12-14-19-20-21-22	Description	tags, text, vote average, num of votes
1:18, **4:**4-16-19-20-21-22-23-24	Stop	direction, mean of transport
1:24, **2:**1-2-7-8-10-12-13-14-15-16-17-19-20-22-24-25, **4:**21-22-23-24-25	Contact	website, phone, e-mail
2:1-2-7-8-10-12-13-14-15-16-17-19-20-22-24-25, 3:7-8-9-11-14-15-18, **4:**11	Municipality	name, ISTAT_municipality

3 Model design

3.1 Introduction

Starting from the generalized queries represented in the table above, this section provides the diagram that represents the EER model. We have used the tool **yEd** to represent the model. The column of the types represents the entities. The column of the properties represents the attributes.
The EER model is very similar to the ER model, the main distinction consists in the subdivision of the entity in three different types. The colour of the model can help the user to distinguish between these types. The types, ordered by decreasing importance, are:

1. core entity types (plus relations and attributes), the blue ones
2. auxiliary entity types (plus relations and attributes), the red ones
3. common entity types (plus relations and attributes), the yellow ones

In order to better represent the entities in our model, we have decided to add another type that is the violet one. It represents the union of two pre-existent types that are common entity types and structured attributes. We have used the new type to define the Address entity. This representation is an adaptation to our model of the original one.
Furthermore, there are two types of attributes, the simple ones and the structured ones. Attributes are defined as structured, if a possible value is a tuple of values, in other words, there are attributes that includes a series of attributes.

3.2 Core entity types

We model as core entities type all the most important entities for our project, since it is on the space domain, we labelled the space entity as core. The most characterizing ones are *Point* and *LineString*. *Point of interest* is the

most used entity, it contains most of the data. The other entities are connected with an is-a relation to the main important ones listed. The core entity types are the cornerstones of the model. The EER, and consequently our ontology, was built on them.

3.3 Auxiliary entity types

The auxiliary entity types, represented with the red colour, are the entities that are not indispensable to the realization of the model but they complete it. We classified as auxiliary the descriptions and the contacts of the points of interest. These two entities can be classified as a structured attribute, for their aim to add more information to the existing entity, in our model core entity.
All the enum type defined in our model have been classified as auxiliary entities type. We have used the enum of the standards to avoid forgetting important classifications. In almost all enum types, however, the "other" option is present in order to not exclude particular cases.
In entity TransportEnum, for example, all the enum types are taken from the standard GTFS ontology.

3.4 Common entity types

The entities related to the time are represented using two different standards, **iCal** and **GTFS**. This choice was made in order to exploit the pre-existing ontology to represent the data with the best structure. The datasets obtained were better suited to the standard reality chosen. This was possible as they are common entity types related to time, in fact, these two standards have been used to map the timetables of means of transport and point of interest like shops, restaurant, accommodation and so on. The I-Calendar standard result simpler than the other standard. As you can see in GTFS there are involved more entity but the choice to use it, was a conscious choice in order to handle even the Stops, StopTimes, Path, Route and Agency for the means of transport. The data structures presented in the EER model is the result of careful analysis and pruning compared to the original standard with the aim of obtaining a structure suitable for the project but containing only the essential entity and attributes, structured or not. Time is an indispensable feature that has to be included in our model to manage shops and transports but can be classified as common because it is not the focus of our project, which is space domain.

3.5 Arrows

As you can see in the model there are different types of arrows. We choose to change the cursor of the arrow to better represent the different relationships. The arrow with the full triangular head represents all the is-a relations. This kind of relationship is the one between a class and a subclass, it is an inheritance. The arrows with a black dot as the cursor are representing the structural attribute. While the other arrows describe other relationships, represented as labels on the arrow itself. As already stated, this differentiation is not part of any standard but was chosen by the group to discriminate, also visually, the different relationships presented in the model.

3.6 Evolution

The EER model has undergone many variations during the project, especially during the ontology creation phase, but even in the Karma modelling phase. Working coordinating the model and the data integration sub-team has led to difficulties in understanding, that have led us to change the EER several times, but this was a regular way to proceed. The final refinement of the EER model has been completed by modifying only small things, mainly

by adding arrows or modifying attributes. The latter has been modified taking inspiration from existing standards models.

3.7 EER model

The final EER model can be observed in Figure 1

4 Model formalization

4.1 Introduction

In this section is explained how the ontology (Figure 2) was created with the use of the Protege, and what are the main differences respect to the EER model designed and described previously.
The goal of the model formalization is to elaborate the EER model in order to create a representation of the entire relational database, described in the EER, that can manage the union with other EER models. Mainly we want to represent the same concepts, with another structure, called ontology.
To build the ontology, owl extension, we have used the Protege tool. As said the ontology and the EER must manage the same data, but using two completely different approaches, for this reason, do not exist an automatic procedure of conversion from one schema to the other.
The main differences that we have encountered, in the conversion from one schema to the other, are:

- In the ontology, everything is an entity (**Class**), not the same as tables. An entity has a meaning even without any relation, instead, a table alone means nothing.
- The relations in an EER model connect two tables. In an ontology the relations (**Object properties**) connect two parties the *Domain* and the *Range*. Both the *Domain* and the *Range* can be a connection to more than one entity.
- All the relations in an EER model are possible thanks to the foreign keys that link a table to the other one. In the ontology the relations exist by themselves, the specification of the foreign key it is useless because the entities are linked by themselves and not based on a specific attribute.
- In the EER model exists the relation *is-a*, to create a sub-table of another one. In the ontology, all the entities have one super-entity, except the root of the entire ontology.

We describe now our real case scenario.

4.2 Classes (Entities)

The inheritance tree structure of the entities is generated starting from the *OntologyRoot* called by default *Thing*. The second level is composed of 4 topics (not classes) that cover the main aspects that we have to manage in the space domain:

- **Point**: Is a class, all the physical objects that we are dealing with are a point. The point provides the localization of something in the space thanks to the three geographical measures: Latitude, Longitude and Altitude.
  The Points are mainly a *PointOfInterest* but there are two exceptions:

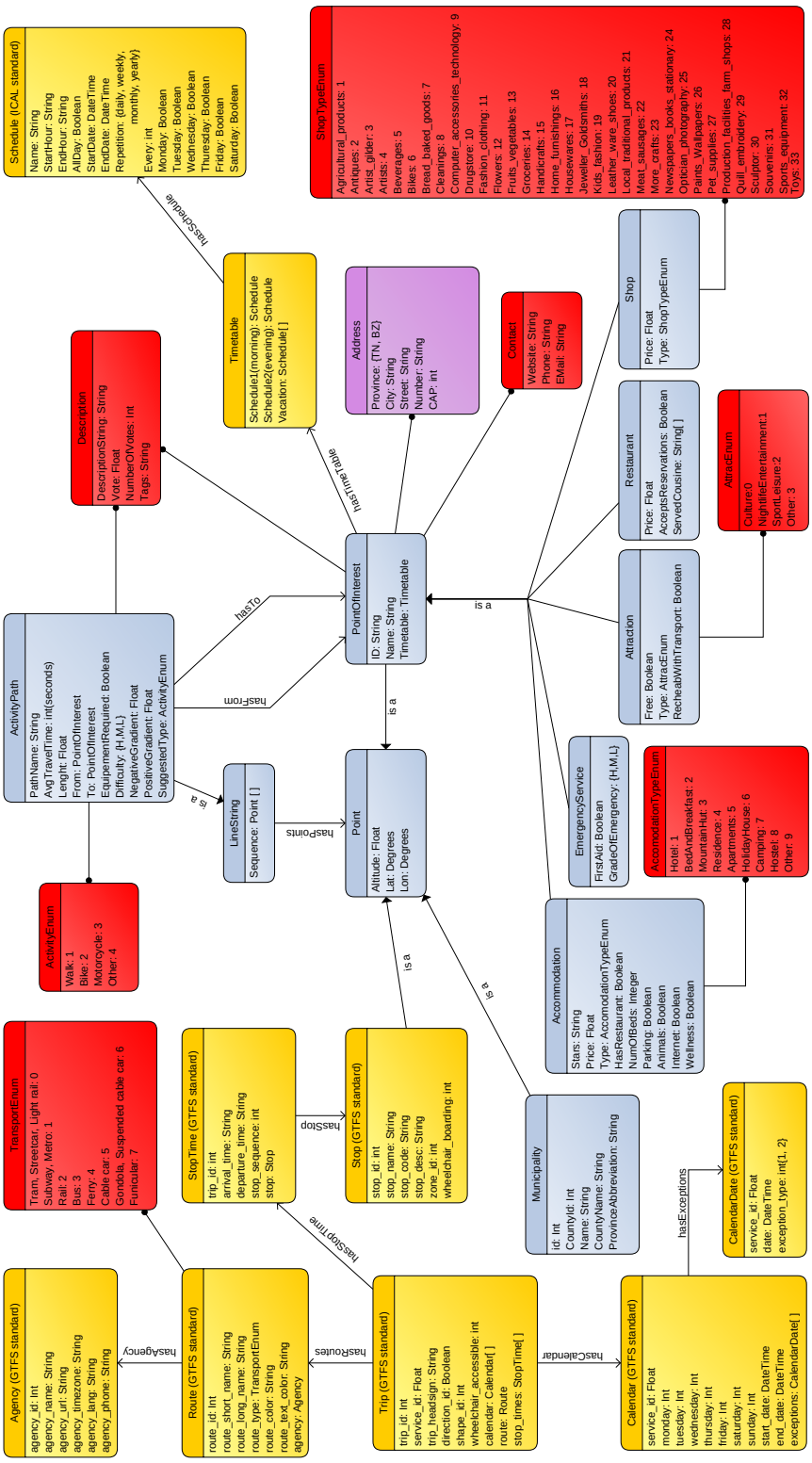


Figure 1: Final Extended ER Model

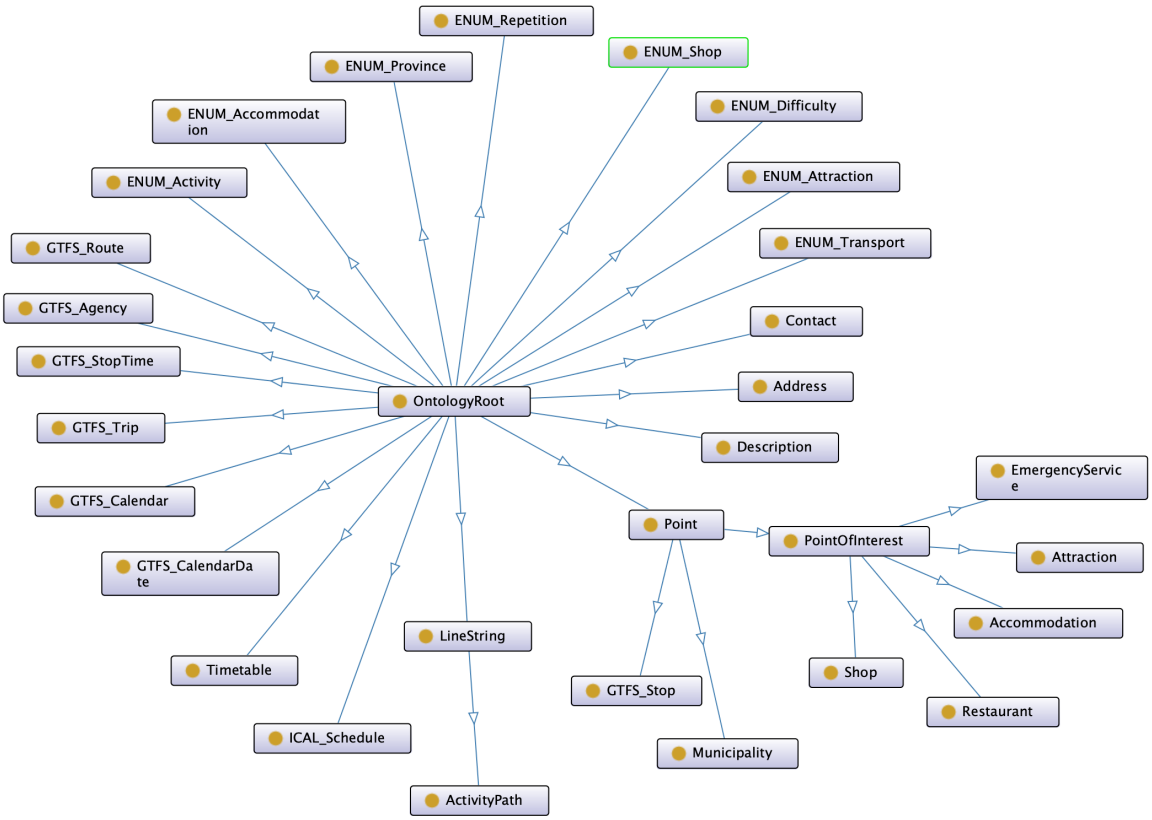


Figure 2: The final version of the ontology that we have elaborated

- *Stop* used from the GTFS standard
- *Municipality* that represent the main point of the area of a municipality

It is important to notice that often in the geospatial domain the coordinates of a point and its address are stored together. This is not the case, we have chosen to split address and coordinates because we want to manage also points not located in urban areas, so not always the address of that point exists, but also because we give different effort to the point of interest that has something to offer, and simple points that are used as intermediate steps in a path.

- **LineString**: Is a class, the connection of physical objects (points) is done with the LineString. This data structure is inspired by the Geography Markup Language (GML).
- **GTFS**: is a topic, it is the standard of the transportation system, it is an alternative connection method respect to LineString. This is the example of the concatenation of two similar concepts in the same ontology. We have chosen to use the standard to manage the stops of public transportation, in order to agree with the providers of the services, but we have also implemented our method to make our application characterized and to manage additional features like defining the trajectory of a path, not only the connections based on this one.
- **Utilities**: is a topic, it contains a set of entities that complete the representability of the other entities of the ontology. The elements are:

- Contact & Description: Are two structured attributes used for adding information respect to a given PointOfInterest
- Address: again a structured attribute, but strictly important for the space domain, for this reason, is coloured as purple in the EER model
- Schedule & Timetable: Are the entities used for representing the ICAL standard. Schedule represent the standard itself, while Timetable is a wrapper for it.
- Enum: is a set of entities used for representing the enumerated type. For each enumerated exist a class that has as instances all the possible values of the enumeration.

It is important to note that we have described GTFS, Utilities and Enum as topics because an entity can be used if it is characterized, this means that contain at least some data, not just like a container. For this reason, our second level ontology contains so many entities, but only 4 topics.
All the names used in this and the future sub-sections can be understood in relation with their domain, the geo-spatial one, but even in this case, the definitions are not always clear. To solve that exist a unique definition provided by WordNet, this formalization can be found at subsection4.5.

4.3 Relations (Object Properties)

The conversion of the EER schema to the ontology has respected the structure of the classes, called now entities, for this reason also the set of relations is mainly respected, and the same will be for the attributes. Some few difference are presented.
In the EER we have put a lot of effort in the type of connection between the classes defined by the type of the cursor of the arrows, in Protégé this effort is lost. All the types of connection are merged into a single type. There is no exception apart for the *is-a* relation that is intrinsically managed by the inheritance structure of the ontology, and so, these relations do not need to be managed as object properties.
Another lack of Protégé is the cardinality of the object properties. A relation is always considered as a 1-to-1 link, but this is not always the case. An example can be the connection between a *LineString* and a *Point*, we have used a semantic notation to describe the relations with the right cardinality, so it is called *hasPoints* to express the fact that a *LineString* can have more than one *Point*, and not simply *hasPoint* (without the s).
In the end, Protégé offers also some advantages: multiple domains and ranges for a relation. In an EER schema, a link is from a class to another one, differently in the ontology the relation has a domain and a range both can contain more than one entity. This is the case of the relation labeled as *hasDescription* the range is the entity *Description*, but the domain is a combination of two entities *PointOfInterest* and *ActivityPath*, both of them can have a description and so the relation exists for both.

4.4 Attributes (Data Properties)

The data properties contain all the attributes of the different classes described by the EER schema, but as for the relations there are few differences.
First of all, the important role of *primary key* of a table is lost. From this point onward all the data properties will have the same importance, the previous attributes labelled as the primary key does not disappear, but lose the reference. Another role that is missing is the *foreign key*, the join of two tables will no longer be done based on the foreign key because in the ontologies every value can be used to compute the integration with another entity.
Another issue to manage is that, by collapsing all the attributes into one single set, some collision can be found,

this can be handled in two methods. With multiple domain and range as for the relations, or by uniquely identify an attribute. We have mainly chosen this second option, an example is the attribute *name* rename with new labels as: *AgencyName*, *CountyName*, *PathName* and so on and so forth.

Finally, to be compatible with the future necessary tools, in particular, *Karma*, each data properties, but also all the object properties and the entities need to be labelled, if not everything will appear flat from the Karma point of view.

4.5 Lexical information description

In order to address heterogeneity and to disambiguate the meaning of all entity types, taking particular care of the core ones, we used as a linguistic resource, WordNet, a lexical database for English. As a first step, we identified the lexical items and phrases we needed for our specific purposes. Next, we found the respective item took as reference for WordNet. While deciding on the most appropriate words for our entities, we focused on picking out the most precise and informative lexical items from WordNet. Considering that some of our core entities are phrases, we put some effort into pinpointing the right mapping. After the identification of the most informative term to be mapped in our model, we annotated the respective classes in the Protégé file. This annotation consists of the id of the synset we selected for the mapping. Although there are some small differences in the term or phrase we used in modelling, we choose not to change the name of the object or the attributes. Hereafter is reported a list of our specific lexical information. The list is organized in four components separated from semicolon: the first element consists of the item or phrase we used for our object/attribute; the second element contains the id (from WordNet) we used for the mapping in Protégé, the third is the WordNet precise item for this id along with the respective category; the fourth element is the synset of the term.

- ***OntologyRoot***; {08524579}; ⟨*noun.location*⟩ [15] S: (n); beginning, origin, root, rootage, source (the place where something begins, where it springs into being)
- ***Address***; {08508037}; ⟨*noun.location*⟩ [15] S: (n); address (the place where a person or organization can be found or communicated with)
- ***Contact***; {07293990} ;⟨*noun.communication*⟩ [10] S: (n) ;contact, touch (a communicative interaction)
- ***Description***; {06737512}; ⟨*noun.communication*⟩ [10] S: (n) ;description, verbal description (a statement that represents something in words)
- ***GTFS_Agency***; {08073958}; ⟨*noun.group*⟩ [14] S: (n); agency (a business or organization that provides a particular service, especially the mediation of transactions between two parties)
- ***GTFS_Calendar***; {15198526}; ⟨*noun.time*⟩ [28] S: (n); calendar (a system of timekeeping that defines the beginning and length and divisions of the year)
- ***GTFS_Route***; {04103160}; ⟨*noun.artifact*⟩ [06] S: (n); road, route (an open way (generally public) for travel or transportation)
- ***GTFS_StopTime***; {06507319}; ⟨*noun.communication*⟩[10] S: (n); schedule (an ordered list of times at which things are planned to occur)

- ***GTFS_Trip*** {09410115} ⟨*noun.object*⟩[17] S: (n) path, track, course (a line or route along which something travels or moves)
- ***LineString***; {08633886}; ⟨*noun.location*⟩[15] S: (n); path, route, itinerary (an established line of travel or access)
- ***ActivityPath***; {09410115}; ⟨*noun.object*⟩ [17] S: (n); path, track, course (a line or route along which something travels or moves)
- ***Point***; {08637636}; ⟨*noun.location*⟩ [15] S: (n); point (the precise location of something; a spatially limited location)
- ***GTFS_Stop***; {08674524}; ⟨*noun.location*⟩[15] S: (n); stop (a spot where something halts or pauses)
- ***Municipality***; {08643858}; ⟨*noun.location*⟩ [15] S: (n); municipality (an urban district having corporate status and powers of self-government)
- ***PointOfInterest***; {00156307}; ⟨*noun.act*⟩[04] S: (n); localization, localisation, location, locating, fix (a determination of the place where something is)
- ***Accommodation***; {03547513}; ⟨*noun.artifact*⟩ [06] S: (n); hotel (a building where travelers can pay for lodging and meals and other services)
- ***Attraction***; {06627914}; ⟨*noun.communication*⟩[10] S: (n); attraction (an entertainment that is offered to the public)
- ***Shop***; {04209460}; ⟨*noun.artifact*⟩[06] S: (n); shop, store (a mercantile establishment for the retail sale of goods or services)
- ***Restaurant***; {04088393}; ⟨*noun.artifact*⟩[06] S: (n); restaurant, eating house, eating place, eatery (a building where people go to eat)
- ***EmergencyService***; {03545775}; ⟨*noun.artifact*⟩ [06] S: (n); hospital, infirmary (a health facility where patients receive treatment)
- ***Timetable***; {06508490}; ⟨*noun.communication*⟩[10] S: (n); timetable (a schedule listing events and the times at which they will take place)

5 Top-level grounding

In this section, we are going to show a possible integration of the ontology we have created, handling the space domain. The main reference has been taken from **schema.org**. The ontologies used are called **GeoCoordinates**[1] and **GeoVocab**[2]. There are plenty of existing ontologies that can be taken to be integrated with other domains. We want to manage a well known and widely used schema.
The idea is to integrate our ontology with an existing one that can be more solid and permit the reusability of the entire ontology. A pre-existent ontology can be more safety because most of the bugs are worked out. Others positive notes to integrate the ontology could be complexity reduction, usability and clarity improvement. The

[1] `http://schema.org/GeoCoordinates`
[2] `http://geovocab.org/geometry#LineString`

main options are: *Schema.org* and *Dbpedia*.

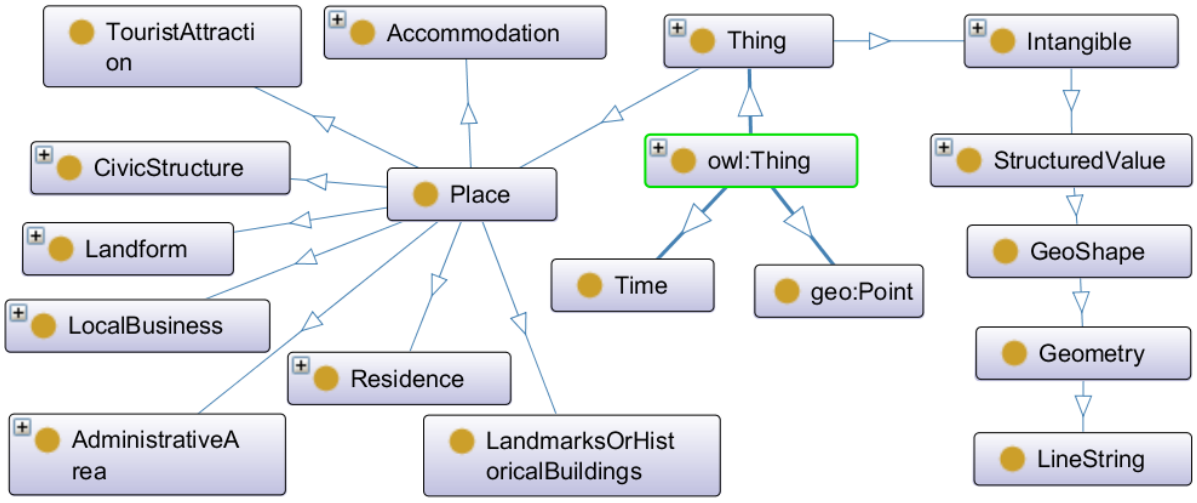


Figure 3: The original ontology of GeoCoordinates and GeoVocab that we want to integrate

Exploring the two schemes we have chosen to use *Schema.org* because the integration can be easily managed thanks to the better adaptation of our ontology to the latter.
As the first step of the integration, we have decided to integrate **GeoCoordinates** and **GeoVocab**. The integration turned out to be easy because the two ontologies are both from Schema.org and are compatible with the vocabulary. Coming to the integration itself by omitting the entities not related to our ontology and the consequent relative subtrees, the main entity for our ontology that have been linked to existing ontologies are:

- **Point**: The Point entity originally was called *geo:Point*, we have decided to set an equivalent relation to the original entity. This decision was made because of the geo:Point entity, as you can see in Figure 3, is exactly the entity Point of our ontology. We have taken the *geo:Point* entity from GeoVocab[3]. Point is a subclass of Thing.

- **PointOfInterest**: This fundamental class for our scenario is extremely similar to *Place*, for this reason, we have added an equivalence between these two classes. The respective sub-classes overlap on few parts but are also an extension one of each other.

- **Intangible**: This entity is used to manage all the entities that are related to the ontology but aren't the focus of the project. We include *Address*, *Description*, *Contact*, all the Enums types and the GTFS ones. Intangible is a sub-entity of *Thing*.

- **LineString**: This entity was not managed in the GeoCoordinates ontology, so our choice is to manage it setting an equivalent class with another ontology. We have taken the *LineString* entity from GeoVocab. LineString path is *Thing/Intangible/StructuredValue/GeoShape/Geometry*.

- **Schedule**: This class represent the ICAL (ICalendar) standard, we have set it as a subclass of the already existing *Time* class.

[3] `http://geovocab.org/geometry#LineString`

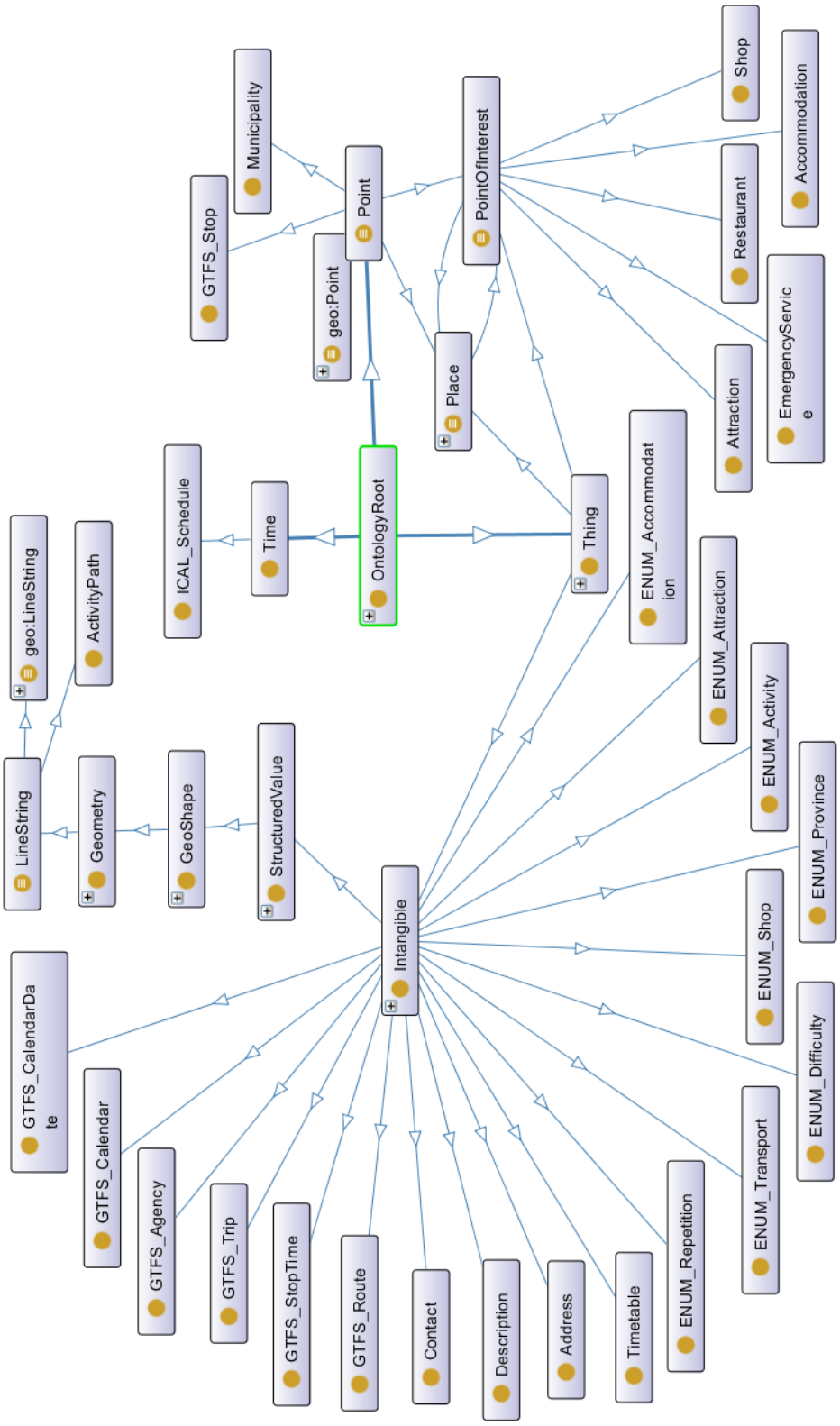


Figure 4: A complete integration of our ontology with GeoCoordinates

In the file included in the git repository, there is the top-level grounding ontology in the complete version, with all the entities. In this chapter, instead, only the part relating to the connection of the two ontologies has been explained.

6 Final considerations and open issue

To evaluate our model and to sum up the work we made during this semester, we used the **OOPS!** tool (OntOlogy Pitfall Scanner!). It detects pitfalls in ontologies, extending the list of pitfalls detected by most recent and available approaches and allowing the selection of subsets of pitfalls to be analysed according to different evaluation dimensions. In addition, the system also provides an indicator (critical, important, minor) for each pitfall according to their possible negative consequences. The system, which is free and online available, operates independently with any ontology development platform.
In this sense, OOPS! helps detecting some of the most common pitfalls occurring in ontology developments. For example, OOPS! warns you when:

- The domain or range of a relationship is defined as the intersection of two or more classes. This warning could avoid reasoning problems in case those classes could not share instances.
- No naming convention is used in the identifiers of the ontology elements. In this case, the maintainability, the accessibility and the clarity of the ontology could be improved.
- A cycle between two classes in the hierarchy is included in the ontology. Detecting this situation could avoid modelling and reasoning problems.

Results for P08: Missing annotations.	135 cases | Minor
Results for P11: Missing domain or range in properties.	1 case | Important
Results for P13: Inverse relationships not explicitly declared.	24 cases | Minor
Results for P19: Defining multiple domains or ranges in properties.	8 cases | Critical
Results for P20: Misusing ontology annotations.	2 cases | Minor
Results for P22: Using different naming conventions in the ontology.	ontology* | Minor
Results for P29: Defining wrong transitive relationships.	1 case | Critical
Results for P30: Equivalent classes not explicitly declared.	1 case | Important
Results for P41: No license declared.	ontology* | Important

Figure 5: Evaluation of the ontology with OOPS! tool

The following points are the explanations of the errors taken from the tool.

- **Missing annotations** This pitfall consists of creating an ontology element and failing to provide human-readable annotations attached to it. Consequently, ontology elements lack annotation properties that label them (e.g. rdfs:label, lemon:LexicalEntry, skos:prefLabel or skos:altLabel) or that define them (e.g. rdfs:comment or dc:description).
  This error is not very important, we have to complete the explanation of the ontology in Protege. We are aware that we have not completely completed the ontology in Protege, this has been done for reasons of time.

- **Missing domain or range in properties** (Solved) Object and/or datatype properties without domain or range (or none of them) are included in the ontology.

- **Inverse relationships not explicitly declared** This pitfall appears when any relationship (except for those that are defined as symmetric properties using owl:SymmetricProperty) does not have an inverse relationship (owl:inverseOf) defined within the ontology.
  This is not a real problem, of course, this specification could improve the quality and accuracy of ontology.

- **Defining multiple domains or ranges in properties** The domain or range, or both, of a property (relationships and attributes) is defined by stating more than one rdfs:domain or rdfs:range statements. In OWL multiple rdfs:domain or rdfs:range axioms are allowed, but they are interpreted as a conjunction, being, therefore, equivalent to the construct owl:intersectionOf. This pitfall is often present inside the representation of the GTFS standard. We consider the multiple domains as a point in favour of Protege, as explained in Section 4.3.

- **Misusing ontology annotations** (Solved) The contents of some annotation properties are swapped or misused. This pitfall might affect annotation properties related to natural language information (for example, annotations for naming such as rdfs:label or for providing descriptions such as rdfs:comment). Other types of annotation could also be affected as temporal, versioning information, among others.
  This is not a real problem, of course, this specification could improve the quality and accuracy of ontology.

- **Using different naming conventions in the ontology** The ontology elements are not named following the same convention (for example CamelCase or use of delimiters as "" or "_").
  This issue could be an important point to work on.
  As already mentioned in the previous chapters, the whole project has undergone iterative developments to allow the modification of already complete sections in order to correct the whole and make it consistent. Surely there would be a further iteration to be done to solve the problem that the tool has identified, to improve ontology.

- **Defining wrong transitive relationships** (Solved) A relationship is defined as transitive, using owl:TransitiveProperty, when the relationship is not necessarily transitive.

- **Equivalent classes not explicitly declared** This pitfall consists of missing the definition of equivalent classes (owl:equivalentClass) in case of duplicated concepts. When an ontology reuses terms from other ontologies, classes that have the same meaning should be defined as equivalent in order to benefit the interoperability between both ontologies. The following classes might be equivalent: Point, Stop
  The classes that the tool considers to be equivalent, aren't. Point indicates a set of coordinates, while Stop indicates a bus stop with other attributes like stop_id, stop_code, zone, code etc., following the GTFS standard.

- **No license declared** The ontology metadata omits information about the license that applies to the ontology.
  This error is common because we can't have a license for our ontology, it is not important so it is not considered.

Future improvement of the schema will be based on the enumeration type, at the moment pointOfInterest has only one level of sub-classes and for each of these ones, there is an enumeration type that considers plenty of options. An improvement will consist of the creation for each instance of these enumeration types, of a sub-class, resulting in the second level of sub-classes for pointOfInterest.

The latest overview of the project needs to be done not only based on the ontology created but also accordingly to the data that needs to fit it. Further conclusions are presented in Section 13.

7 Scenario - Data Integration

The scenario used for the data integration is the same as described in Section 1 for the from-informal-to-formal model phase. There have been many iterations over the original design, aligned towards the datasets taken into consideration, with focus on:

- Limiting the depth of the model, as the broadness of the topic didn't allow us to define deeply in detail all the minor entities of the scenario.
- Data coverage, adapting the model to the data sources for the system to fully exploit the information
- Non-triviality, that is maintaining a certain level of complexity that can be fully exploited by detailed competency questions and queries.

For what concerns the **scenario definition**, the Data Integration subgroup was involved mainly in the revision, while working on the dataset selection and gathering. Continuous inputs have been provided to the other subgroup in order to guide the definition of the personas also according to what seemed to be covered by the data sources (still hypothetical, in this phase).

The **personas** has also been revised several times to diversify them and exploit model and data and, on the other hand, to limit their depth, therefore avoiding to lose the focus of the work in specific details beyond our area of interest.

8 Datasets selection and extraction

Many data sources have been considered, in particular for what concerns the numerous types of points of interest(POI) that had to be included. This also brought us in front of the choice of whether to consider province of Trento, Bolzano or both. In the following sections we are going to separately analyse the available data sources, highlighting their characteristics and issues, which brought us to make specific design choices.

8.1 Topic division and considerations

In order to efficiently parallelize the datasets search we split the workload into four main topic, according to Table 3.

Table 3: Workload division for the data integration tasks

Michela	Accommodations and hotels
Sara	Transportation and municipality
Daniele	General points of interests, spanning from private shops to public structures.
Alessandro	Mountain and hiking, including mountain huts and paths.

8.1.1 Accommodations and hotels

Accommodations are not an easy information that can be found in big public datasets. In fact, all the data of this type we considered comes from web scraping. Initially we considered as data sources medium-size hotel search engines, in order to gain a good amount of information and avoiding the anti-scraping countermeasures that big websites adopts. We initially considered several small websites which specialized on some specific interesting information, for instance the possibility of hosting animals and providing wellness services, however the data was insufficient and qualitatively poor. Consequently, after several attempts, we settled on working with `Booking.com`[4], which ended up being easier to scrape than other minor websites while still providing very specific information that was needed for the queries.

Specifically, the scraping has been performed with ad-hoc python scripts using the *scrapy*[5] library, as the dynamic pages couldn't be analyzed with plain HTML scraping. The scripts automatically query the website for accommodations in Trentino and Alto-Adige. In order not to get the client blocked for the number of requests, we implemented a bash script to launch the python scraper periodically, which successfully retrieved more than enough results.

We also used the *Esercizi alberghieri*[6] dataset in order to retrieve the contacts and the prices for the available hotels in Trentino.

8.1.2 Transportation

The initial idea for the transportation data gathering was to fetch public services and APIs, however the quality of data was not satisfying for our purposes. We tried using Google Maps APIs, but we encountered some difficulties due to the limitations for free users, therefore we finally decide to download from *dati.trentino* [7] for what concern the province of Trento, and from *sasabus*[8] for Bolzano. Since we want to create an ontology that is compatible with as many data as possible, we decide to take the Google transport standard called GTFS[9].

For what concern the data of Bolzano, the only datasets available are in VDV 452 standard. However, using a public library we were able to transform them in GTFS standard in an efficient way (explain in the next chapter). Because of this intermediate step, the files that has being taken in consideration in our model are less than the standard provided. However, all the required files are integrated in the ontology, and in the future it is possible to expand it with the optional ones.

8.1.3 Municipalities

Municipality dataset has been included because of the queries about POI around a city. These data has been downloaded from ISTAT page[10] for all the city in Italy and then only the municipality in Trento and Bolzano province has been considered. Since these data do not contain coordinate values, we integrate them by calling the OpenStreetMap API[11].

[4] `https://www.booking.com`
[5] `https://scrapy.org/`
[6] `https://dati.trentino.it/dataset/esercizi-alberghieri`
[7] `https://dati.trentino.it/dataset/trasporti-pubblici-del-trentino-formato-gtfs`
[8] `http://sasabus.org/it/opendata`
[9] `https://developers.google.com/transit/gtfs`
[10] `https://www.istat.it/it/archivio/6789`
[11] `https://www.openstreetmap.org/`

8.1.4 General points of interest

Point of Interest is a very broad category to search datasets for. We started searching for them from the *Open Data Trentino* catalog. After some researches on the platform, we found out that there exists some APIs available that let you retrieve all sources that contains a dataset with the tag "luoghi e punti di interesse" ("places and point of interests" in english)[12]. This API endpoint returns a list of sources with links to download the datasets.

Then, the other datasets we have retrieved for Point of Interest, that are the one from province of Bolzano, have been retrieved from the *OpenDataHub APIs*. In particular we have used the *Poi* endpoint[13] which provided us more than 8000 records. But this dataset didn't come with restaurants so, we also used the *Gastronomy* endpoint[14].

8.1.5 Mountain hiking and huts

The main characteristic of the information on mountain places and activities is that it is peculiar to the specific geographic area. In fact, maps and places are more often than not collected in more traditional means of communication, like paper maps and books. In all our research, the only practical and rich enough data source for what concerns Trentino is the SAT website[15] and its services, in particular:

- Mountain paths data sheets, including complete lists of stages and places along the path, altitude and gradient information, but no specific coordinates positions. This had to be scraped directly from the website, which we accomplished by means of python scripts (using Requests[16] and ElementTree[17] modules) designed to repeat the scraping whenever needed in order to keep the information as up to date as needed.
- Mountain paths *.shp* shapefiles, directly obtained from the SAT website dedicated page[18]. These contains the latitude/longitude information about the path points, which are not present in the data sheets, but no further information.
- Mountain huts SAT OpenData[19], linked to the web pages[20] containing data sheets. The Open Data contains information about the huts including names, positions and links to the aforementioned pages, which add descriptions and specific accommodation details (e.g. number of beds, contacts, openings).

Additionally, we seeked for options to include Alto Adige information, however the lack of publicly available Open Data left scraping as the only choice. The Sentres[21] search engine appears to be the only available source, but considering the difficulty of this scraping task and the time constraints, we finally decided to discard this option and limit the mountain data coverage to Trentino.

[12] `https://dati.trentino.it/api/3/action/package_search?q=luoghi+e+punti+di+interesse`
[13] `http://tourism.opendatahub.bz.it/api/Poi`
[14] `http://tourism.opendatahub.bz.it/api/Gastronomy`
[15] `https://www.sat.tn.it`
[16] `https://2.python-requests.org`
[17] `https://docs.python.org/3/library/xml.etree.elementtree.html`
[18] `https://sentieri.sat.tn.it/static/download-sentieri.html`
[19] https://api.webmapp.it/trentino
[20] `https://www.sat.tn.it/rifugi/elenco-rifugi`
[21] https://www.sentres.com

9 Datasets cleaning, merging and analysis

9.1 Accommodations and hotels

For the cleaning and merging phase, we used RapidMiner tool. We have two different datasets that have to be merged in order to gather all the information we need to correctly represent accommodations. We created a RapidMiner process composed by different steps, as shown in Figure 6. First of all, we have two datasets: a dataset coming from open Data in xml format; a scraped dataset coming from booking in json format. The process steps are:

- Read the xml, extract the attributes we want to extract, rename the columns in order to have proper names, split the price column in order to obtain all the prices available and set the missing values;
- Run a python script (`cleanEncoping.py`) that generate a csv file starting from the json dataset, read the csv and generate a column that will be used to merge the two datasets;
- Use the upper case name of the accommodations to merge the datasets;
- Generate a column containing the enumeration uri for the accommodation type, replace the boolean values from upper case to lower case, select the correct attributes and write the resulting data in a csv file.

The python script, that generate a csv file starting from a json file used in the RapidMiner process, calculates the address fields of each hotel starting from an address string scraped from the web page and contained in the json dataset. In order to obtain the address fields, we used the *pypostal*[22] library that is a library for parsing street addresses using statistical NLP and open data.

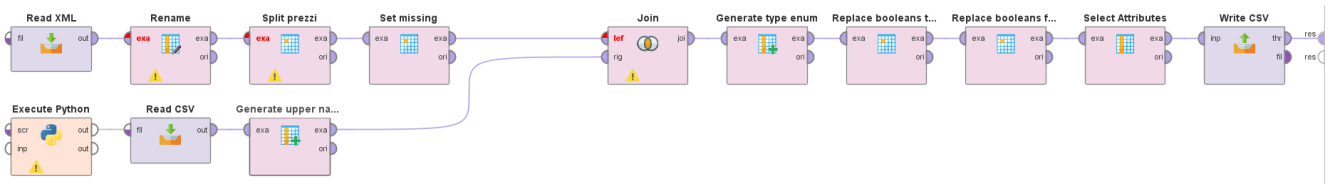


Figure 6: The hotel process in RapidMiner.

9.2 Transportation

For transportation, we decide to take GTFS standard. The transportation dataset of Trento is divided in *urbano* and *extraurbano* in the chosen standard. However, Bolzano data are available only in VDV 452 format, we use a public library called *onebusaway-vdv-modules*[23] that converts the dataset in files compatible with GTFS. The files created with this process are the required ones (agency.txt, stops.txt, route.txt, trips.txt, stop_times.txt, calendar.txt), where you can find the basic information on the bus and their timetable, plus an optional one (calendar_date.txt).

After doing that, we unify the three datasets using RapidMiner (Figure 7) between corresponding files. First, we did the union between the Trentino datasets, and then with the Bolzano data. Eventually, a new csv file is exported.

[22] `https://github.com/openvenues/pypostal`
[23] `https://github.com/OneBusAway/onebusaway-vdv-modules`

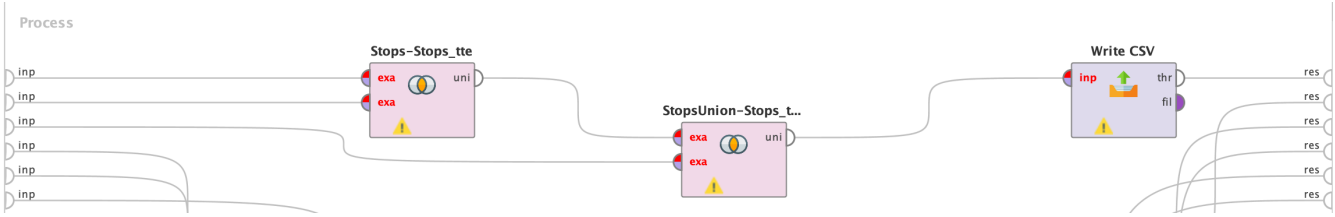


Figure 7: The transportation process in RapidMiner

9.3 Municipality

First of all, looking at the dataset it is possible to notice a couple of "new line" command in the header of the table. In order to prevent an incorrect division of the columns in RapidMiner we remove manually this two commands. Then, the data retrieved from the the ISTAT site needs to be clean and integrated with coordinates. For doing that, we create a process with RapidMiner (Figure 8).

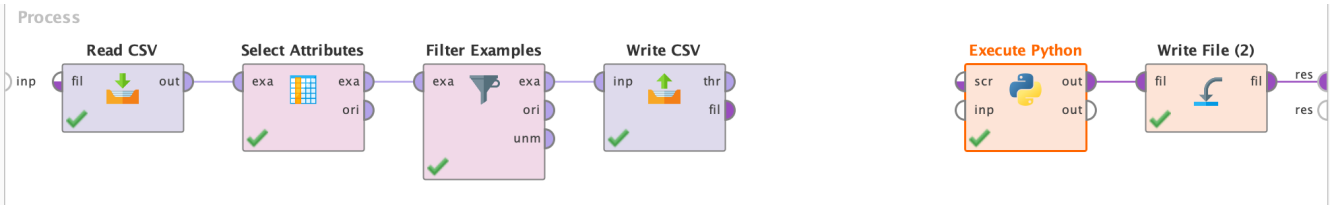


Figure 8: The municipality process in RapidMiner

Initially, we select only the columns we are interesting in, which are: *codice comune in formato numerico, codice regione, denominazione in italiano, denomincazione regione* and *sigla automobilistica*. Since in the file there are the municipality of the whole nation, we select the city within the province of Trento and Bolzano. This has been done looking at the province abbreviation on the licence with a *Filter* command. After that, a Python script is executed on the file created with the selected data. In this script, we use Nominatim, which is a search engine for OpenStreetMap data, and passing to it each name of the municipalities we add latitude and longitute coordinates at the existing table.

9.4 General point of interest

For what concerns Point of Interests, the data preparation process needs to be separated in two parts. One for the datasets from the province of Trento and the other from the ones coming from the province of Bolzano.

9.4.1 Province of Bolzano

The first we addressed is the one from the province of Bolzano. Since the final EER model has more than one entity inheriting from the PointOfInterest entity, we decided to split the big dataset retrieved from the OpenDataHub APIs in as much datasets as the number of the child entities of PointOfInterset except the Gastronomy and Accomodation ones which already had their specific datasets. This process of separation has been not that trivial because of the complex structure of this json dataset. We started this process by retrieving from the APIs all the types of the POIs that were present in the dataset, this task was easily accomplished thanks to the *PoiTypes*[24] endpoint. Then,

[24] `http://tourism.opendatahub.bz.it/api/PoiTypes`

we mapped by hand each of the main types to the various entities, and for some of them, like the Shop one, we have mapped the subtypes with the enums from the EER model. The result of this mapping can be seen in Table 4.

EER Entity	**Type(-Subtype) list**
PointOfInterest	"Service providers", "Traffic and Transport (without Bus stops, Taxis, and Railways subtypes)", "Public institutions (without Hospitals subtype)"
Attraction	"Culture and sights", "Sports and leisure", "Nightlife and entertainment"
Shop	"Shops", "Craft"
EmergencyService	"Public institutions-Hospitals", "Doctors, Pharmacies"

Table 4: Entitiy → Type-SubType mapping for BZ

After this process done by hand we split the dataset by using the filters available in the OpenDataHub APIs.

9.4.2 Province of Trento

When the datasets from Bolzano were ready, we started cleaning the one from Trento. The datasets from Trentino were separated by Municipality. So, the first thing was to merge them and create only one dataset for all Municipalities. Once downloaded, we started analyzing this and we found out that a fraction of this dataset have latitude and/or longitude equal to zero or it was out of the bounding box for the province of Trento. The values for this bounding box were found in the website of the province of Trento[25]. To avoid this errors we cleaned the dataset by removing all POIs that were out of this bounding box. Another thing we noticed was that addresses weren't in a standardized format and, moreover they were all in one long string instead of being separated in multiple columns (like street, house number, postal code, etc.). So, we used the *libpostal*[26] library with the *pypostal* python wrapper to split them.

Once this cleaning on the full dataset has been done, we have analyzed the types for this dataset and we mapped by hand each of this types to the 5 entities in the EER model, just like we have done for the Bolzano datasets. Moreover we have translated the types from Italian to English so that we could use those values for the Tags field inside the Description structured type we have in the EER model. Once we have done this, we split the dataset in 4 different datasets, one for each entity in the EER model. In this process, some errors may occur during the execution of the python script beacause some new type is added, this is a wanted error so that we can add that type to the mapping.

The mapping for the dataset from Trentino can be seen in the 5

9.5 Mountain hiking and huts

9.5.1 Mountain paths

The mountain path dataset is obtained by combining the results of the SAT website scraping and the parsed shapefile geometry information. For what concerns the scraping, a dedicated python script has been implemented (`get_sentieri.py`), downloading the html data sheets and returning a csv file. The script works in two modes, it can either download the data or parse already downloaded data: in this way there is no need to download again all the information in case of a change in the structure of the output csv.

[25] `http://www.territorio.provincia.tn.it/geoportlet/srv/eng//metadata.show?uuid=p_tn:Ceduo`
[26] `https://github.com/openvenues/libpostal`

EER Entity	Type list (in Italian)
PointOfInterest	“Chiesa”, “Biblioteca”, “Ufficio postale”, “Scuola infanzia”, “Scuola elementare”, “Trasporti”, “Carabinieri”, “Vigili del fuoco”, “Nido infanzia”, “Scuola (media o secondaria)”, “Polizia”, “Luogo religioso”, “Polizia locale”, “Varie”, “Malga”, “Patrocinato sindacale”, “Centro diurno”, “Luogo Religioso”, “Parcheggio”, “Alloggio protetto”, “Nido d’infanzia”, “Scuola d’infanzia”, “Centro infanzia”, “Polo sociale”, “CRM”, “Info point”, “Punto allattamento / cambio”, “Spazio di aggregazione ”, “Etnografia”, “Punto di interesse”, “Associazione”, “Infrastrutturazione turistica”
Attraction	“Palazzo”, “Forte”, “Parco naturale”, “Piscina”, “Centro sportivo”, “Museo”, “Sito storico”, “Castello”, “Sci aplinismo / Sci Escursionismo”, “Scuola di sci”, “Località turistica”, “Via o piazza”, “Centro di associazione (musica, teatro...)”, “Frazione”, “Biotopo”, “Monte”, “Stadi - campi da calcio”, “Bocciodromo”, “Parco giochi”, “Arrampicate”, “Mountain Bike”, “Luogo incantevole”, “Rovine”, “Monumento”, “Sito Archeologico”, “Teatro”, “Lago”, “Ciclismo”, “Parco avventura”, “Alpinismo”, “Vela/Windsurf”, “Ferrata”, “Pista ciclabile”, “Snow park”, “Trekking”, “Ecomuseo”, “Parchi e orti botanici”, “Percorso naturalistico”, “Musei”, “Area archeologica”, “Altri siti di interesse storico artistico”, “Natura”, “Edificio storico”, “Parchi e giardini”, “Impianto sportivo multidisciplinare”, “Poligono di tiro”, “Campo da tennis”, “Campo da calcio”, “Centro di equitazione”, “Altri siti di interesse turistico”, “Dolomiti”, “Cascate”, “Monumento naturale”, “Tennis”, “Canyon/Grotte”, “Area faunistica”, “Canoa/Kayak”, “Canyoning”, “Ciclabili”, “Downhill Bike Park”, “Equitazione”, “Esposizione/Galleria”, “Fiumi e torrenti”, “Ghiacciaio”, “Golf”, “Ice Climbing”, “Noleggio attrezzature sportive”, “Nordic Walking”, “Parapendio/Deltaplano”, “Rafting”, “Sci alpinismo / Sci Escursionismo”, “Tiro con l’arco”, “Via Claudia Augusta”, “Piste da sci”, “Ciaspolata”, “Pattinaggio sul ghiaccio”
Shop	“Mercato”, “Bottega storica”
EmergencyService	“Pronto soccorso”, “Ambulatorio”, “RSA (Residenza Sanitaria Assistenziale)”, “Farmacia”, “Casa di riposo”, “Sanità”

Table 5: Entitiy → Type mapping for TN

The geometry information is encoded in the *.shp* format, which is not easy to integrate with the scraped data. In order to make it easily usable we processed it through the opensource tool Mapshaper[27] in order to obtain a topojson file, which we parsed with an ad-hoc python script(`parse_topojson_sentieri.py` into a csv format.
The two resulting datasets has been merged by means of a simple RapidMiner process, executing the scraping stript and loading the parsed mapshaper csv file, as shown in Figure 9. The initial plan consisted in splitting the complete SAT paths in many smaller ones, going from an intermediate stop to the next. However, there was no way for us to find the position of the intermediate stops in the data we found: the topojson seemed to have this information, as the paths are made of many arcs for which the starting and ending point do have coordinates, but these didn’t match with the intermediate stops listed in the html path data sheets. This lead us to keep the path in one piece (duplicated, for the return path), with all the intermediate points being taken from the json to compose a LINESTRING[28] value.

[27] `https://mapshaper.org/`, `https://github.com/mbloch/mapshaper`
[28] `https://docs.microsoft.com/it-it/sql/relational-databases/spatial/linestring`

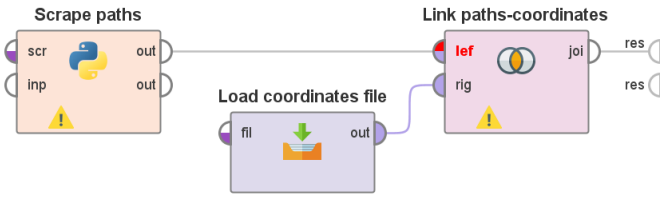


Figure 9: RapidMiner process for merging mountain paths datasets

9.5.2 Mountain huts

As previously mentioned, mountain huts data is loaded starting from a geojson file[29], available through the SAT public APIs. This file is loaded by a python script (`get_rifugi.py`), which parses it and extract position data and a link to the html data sheet, also hosted on the SAT website. The script automatically proceed parsing the web pages and extracting the rest of the information (see Section 8.1.5 for more details). The information is therefore merged within the script and saved in a csv file.

10 Related ontologies

Our ontology has the characteristics of being general and expandable because of the hugeness of the argument, but at the same time specific enough to make some interesting queries. Because of this, it is possible to integrate other ontologies already existing, like the standard ones. The three formal models that we decided to take into consideration as proposal are the following: GTFS and iCal standards, and Geometry.

10.1 GTFS Ontology

For the transportation part, we include only the files that are required from the GTFS standard, but there are a lot more that can be added into the formal model. As you can see in Figure 10, our ontology is perfectly compatible with the proposed one[30], because all the entities we created are mappable. Also, all the new classes can be added to our formal model following our modeling, including the types present in the new one; indeed, the entity called ENUM_Transport is equal to TrasferType.

10.2 Geometry Ontology

We propose the Geometry Ontology[31] to expand and map our Space domain using a standard that allow us to have sequences of points that can compose paths. In fact, we used the class Linestring, as shown in Figure 11, to represent the points contained in the ActivityPath class. We also used the Point class of the Geometry Ontology to create our Point class.

10.3 ICal Ontology

The proposal for the iCal ontology has been retrieved from the W3 organization[32]. The structure of this model can be seen in Figure 12 This formal model has been proposed to expand the simplistic version of the iCal standard

[29] `https://api.webmapp.it/trentino/geojson/rifugi.geojson`
[30] `http://vocab.gtfs.org/terms`
[31] `http://geovocab.org/geometry`
[32] `http://www.w3.org/2002/12/cal/ical`

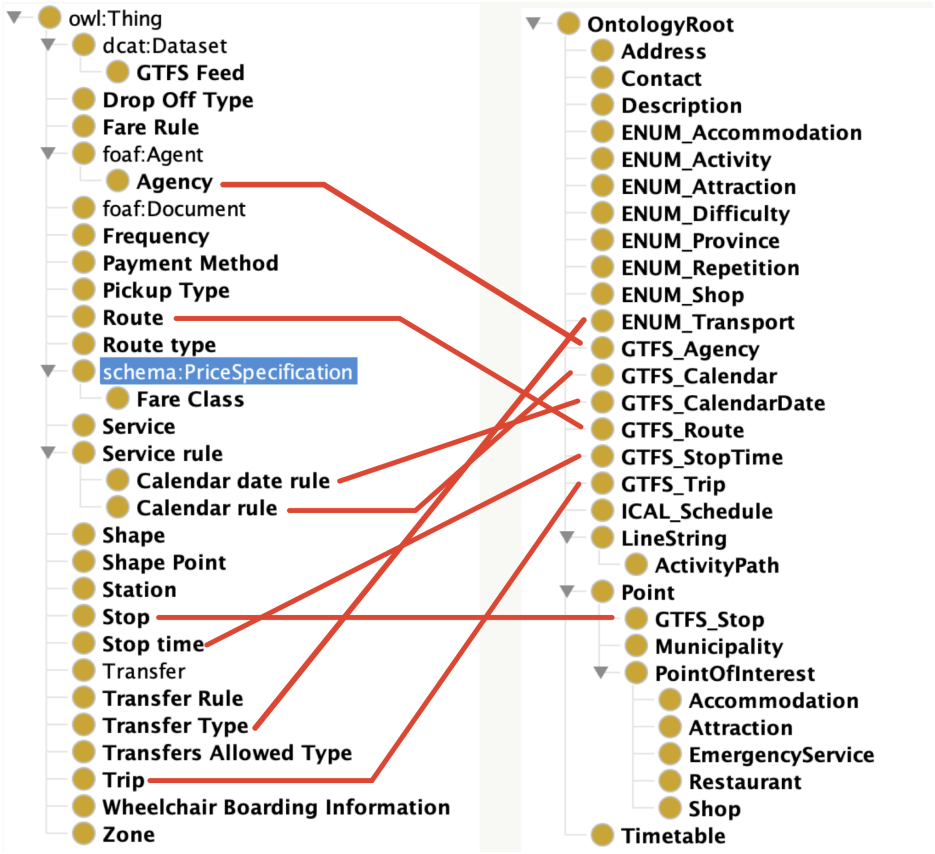


Figure 10: Mapping between GTFS Proposal ontology and GeoSpace ontology

that is available in our ontology. Since our domain is GeoSpace and iCal is a standard useful to manage time, we have finally decided to keep the structure in our formal model as is but to leave the possibility to expand it.

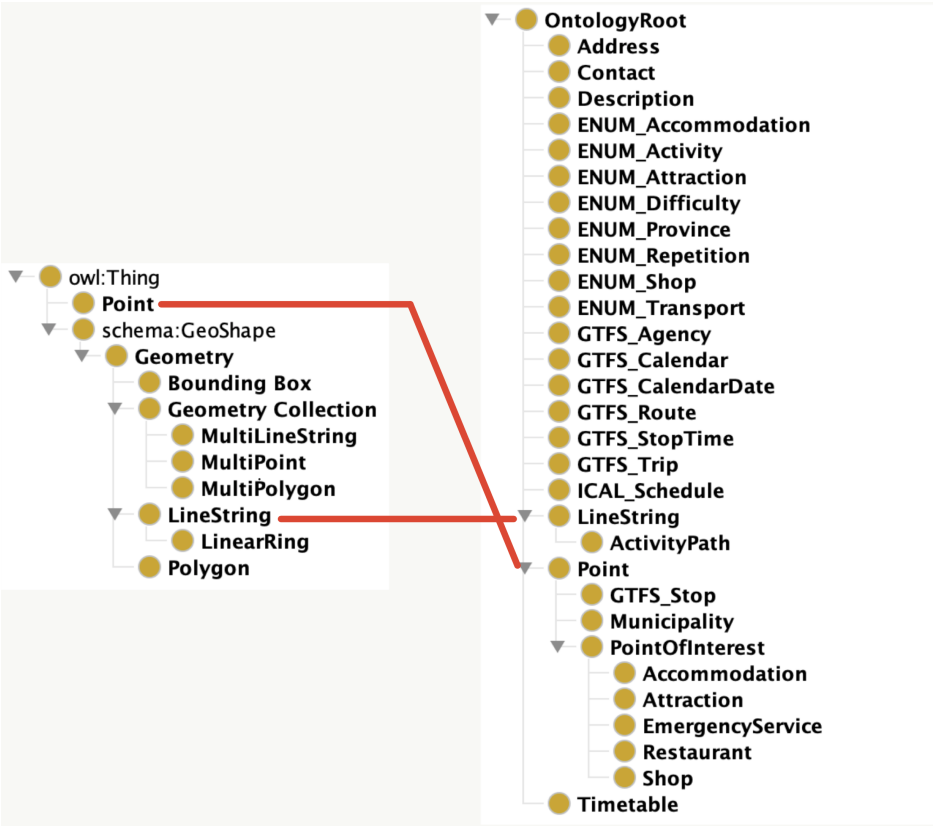


Figure 11: Mapping between Geometry and GeoSpace ontology

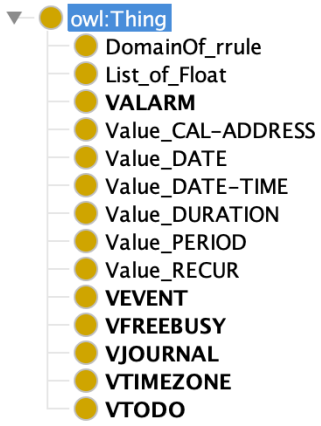


Figure 12: ICal ontology

11 Integration process

The main entities of the model are Point and PointOfInterest, with all the subentities of the latter and the Enum auxiliary entities. The main URIs of these are computed based on the information contained in the entity itself, according to Table 6, with some exception which are going to be listed in the following sections. Note that all the entities use as base URI the ontology one, which is *http://www.semanticweb.org/aleca/ontologies/ 2019/10/untitled-*

ontology-10. Moreover, all the values containing strings have been elaborated replacing the special characters (such as the tab command, quotation marks, backslash, etc..) with underscore or with nothing.

Table 6: Karma URIs for common entities

Entity	**URI computation**
Point	[base_uri]/point_[latitude]_[longitude]
Point Of Interest	[base_uri]/pointofinterest_[latitude]_[longitude]_[poi_name]
Description	[base_uri]/description_[hash(description)]
Province Enum	if province is 'BZ' return [base_uri]/#BZ, if province is 'TN' return [base_uri]/#TN
Address	[base_uri]/address_[hash(city + province + street + number + cap)]
Timetable	[base_uri]/timetable_[poi_name]_[city]
Schedule	[base_uri]/schedule_[schedule_name]_[poi_name]_[city]
Contact	[base_uri]/contact_[hash(website+email+phone)]

11.1 Accommodations and hotels

For the accommodations' dataset, we used the PyTransform available in KARMA in order to generate some new columns. The first generated column is *min_price* that calculates the minimum price between the available prices for each accommodation. Then, we generated a column for each uri in order to link each uri with its relative class. We had some problem in the uri generation caused by the presence of special characters in the accommodations' names used as identifier of one accommodation. For this reason, we had to replace the special characters with the _ character in order to generate correct uri.

Table 7: Karma PyTransform for Accommodations

New column name	**PyTransform**
min_price	Float value, computed taking the minimum value between prezzo_1 and prezzo_2
accommodation_uri	[base_uri]/accomodation_[accommodation.poiName]_[accommodation.city]
province	String value, computed based on value of cap. If cap starts with 39 return `BZ`, if cap starts with 39 return `TN`
checkin	String value, returns "checkin" and it used to identify the checkin schedule name
checkin_uri	[base_uri]/schedule_[checkin]_[accommodation.name]_[accommodation.city]
checkout	String value, returns "checkout" and it used to identify the checkout schedule name
checkout_uri	[base_uri]/schedule_[checkout]_[accommodation.name]_[accommodation.city]

11.2 Transportation

Transportation data are divided in 7 files, and each one need to be mapped in order to create the links between themselves and with Point. In Table 8, you can find all the PyTransform created in each file, and a little explanation on what type of connection they are made for.

Table 8: Karma PyTransform for Transportation

New Col. Name	**PyTransform**	**Connections**
	Stops.txt	
stops_uri	[base_uri]/stop_[stop_id]	With a *SubclassOf* outgoing link to point_uri
	StopsTimes.txt	
stops_uri	[base_uri]/stop_[stop_id]	*uri* of Stops
stopTimes_uri	[base_uri]/stopTimes_[stop_id]_[trip_id]	*hasStop* with Stop
trip_uri	[base_uri]/trip_[trip_id]	*hasStopTimes* to StopTimes. It has been done in this file because in *StopTimes.txt* there is not stop_id
	Trip.txt	
trip_uri	[base_uri]/trip_[trip_id]	*HasCalendar* to Calendar and *hasRoute* to Route. These connections has been done here because in *Calendar.txt* does not exist trip_id
calendar_uri	[base_uri]/calendar_[service_id]	*uri* of Calendar
route_uri	[base_uri]/route_[route_id]	*uri* of Route
	Calendar.txt	
calendar_uri	[base_uri]/calendar_[service_id]	*uri* of Calendar
	CalendarDates.txt	
calendar_uri	[base_uri]/calendar_[service_id]	*uri* of Calendar
calendarDates_uri	[base_uri]/calendarDates_ + [hash(service_id +date + exception_type)	*uri* of CalendarDates
	Routes.txt	
route_uri	[base_uri]/route_[route_id]	*uri* of Route. *hasTransportEnum* to TransportEnum e *hasAgency* to Agency
Transport_enum	if str(getValue('route_type')) == '0': '[base_uri]#T_Tram_Streetcar_Light_rail' *(...and continue with the other types of the enum)*	*uri* of TransportEnum
agency_uri	[base_uri]/agency_[agency_id]	*uri* of Agency
	Agency.txt	
agency_uri	[base_uri]/agency_[agency_id]	*uri* of Agency

11.3 Municipality

For what concern municipality, the modeling phase with Karma is pretty fast. The only PyTransform needed are for the uris, so *point_uri* and *municipality_uri*. For the last one, we decide to append only the numeric code of the city because it is a unique value for all the municipality of Italy. In Figure 13 you can find the data integration process.

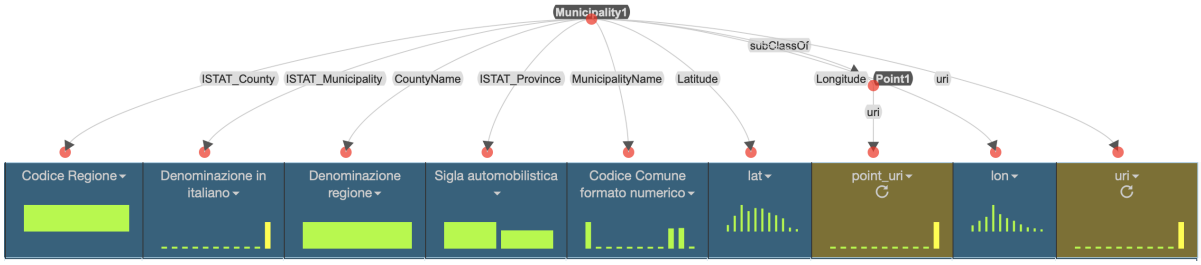


Figure 13: The data integration process in KARMA for the municipality dataset.

11.4 General point of interest

For the Point of Interest even in this section we have to divide concerns between province of Trento and Bolzano.

11.4.1 Province of Trento

For each of the datasets from Trentino we have applied the same transformations using the pyTransform feature of Karma. This can be seen in Table 9

Table 9: Karma PyTransform for POI TN

New Column Name	PyTransform
Province Enum	[base_uri]#TN if len(getValue('Province')) > 0 else ''

The only difference between this and the other TN datasets is that there is another uri that is analogue to poi_uri named <entity_name>_uri and has <entity_name>_ in the uri instead of poi_ after the base uri.

Unfortunately datasets from Trentino lacks of some attributes, which are all of the attributes that compose Timetable and Schedule.

11.4.2 Province of Bolzano

As well as province of Trento even the datasets from the province of Bolzano are mapped very similarly to the Ontology. The biggest difference has been that the datasets from Bolzano were in JSON format instead of CSV. This has created some problem when data was nested in the JSON model. Fortunately, Karma has some python APIs that helped us in overcoming this problem. We will provide the same example (POI dataset) for PyTransforms as the Trentino dataset in table 10.

Table 10: Karma PyTransform for POI BZ

New Column Name	PyTransform
enum_province_uris	[base_uri]#TN if len(getValue('Province')) > 0 else ''
house_num	house_number = re.compile('[\d]*[/]?[a-zA-Z]*[ ]*$') nums = house_number.findall(getValue('Address')) return nums[0] if len(nums) > 0 else ''
address_street	street = re.compile('^((?![\d]+[/]0,3[a-zA-Z]*[ ]*$).)+') s = street.match(getValue('Address')).group() return s if len(s) > 0 else ''
CityNoZipCode	re.sub(r'[ ]?\d5[ ]?', ' ', getValue('City')).strip()
Province	'BZ' if len(getValue('City')) > 0 else ''
schedule_uri	[base_uri] + getValueFromNestedColumnByIndex('OperationscheduleName', 'de', 0).replace('|', '').replace('"', '').replace('\t','') + '_' + getValueFromNestedColumnByIndex('Detail', 'en/Title', 0).replace('|', '').replace('"', '').replace('\t','') + '_' + getValueFromNestedColumnByIndex('GpsInfo', 'Latitude', 0) + '_' + getValueFromNestedColumnByIndex('GpsInfo', 'Longitude', 0)
timetable_uri	[base_uri] + getValueFromNestedColumnByIndex('Detail', 'en/Title', 0).replace('|', '').replace('"', '').replace('\t', '') + '_' + getValueFromNestedColumnByIndex('GpsInfo', 'Latitude', 0) + '_' + getValueFromNestedColumnByIndex('GpsInfo', 'Longitude', 0)

11.5 Mountain hiking and huts

11.5.1 Mountain paths

The mountain path datasets is mapped to the ActivityPath entity, which is designed to model this type of path. We used the PyTransform function to compute the ActivityPath *difficulty* and *EquipmentRequired* fields, and to generate URIs for the various entities involved in the model according to Table 11.

Table 11: Karma PyTransform for mountain paths

New Column Name	PyTransform
Difficulty	URI of the DifficultyEnum corresponding instance. Computed based on value of DifficoltaSentiero, which is a value in T (*Turistico*), E (*Escursionisti*), EE (*Escursionisti Esperti*), EEA-F (*Escursionisti Esperti con Attrezzatura - Facile*), EEA-PD (*Escursionisti Esperti con Attrezzatura - Poco Difficile*), EEA-D (*Escursionisti Esperti con Attrezzatura - Difficile*), EEA-MD (*Escursionisti Esperti con Attrezzatura - Molto Difficile*). The first is mapped to the `[base_uri]/#Low` difficulty, the second to `[base_uri]/#Medium`, all the other to `[base_uri]/#Hard`

Continued on next page

Table 11 – *Continued from previous page*

New Column Name	PyTransform
Equipment Required	Boolean value, computed based on value of DifficoltaSentiero. `True` if DifficoltaSentiero contains `A` (*con Attrezzatura)*
ActivityEnum URI	[base_uri]/#Walk
Linestring URI	[base_uri]/linestring_[hash(linestring.sequence)]
ActivityPath URI	[base_uri]/activitypath_[activitypath.name]

11.5.2 Mountain huts

The mountain huts data is mapped in Karma to the Accommodation entity, connected to the AccommodationEnum instance `[base uri]/#MountainHut` specifying the mountain hut accommodation type.

Table 12: Karma PyTransform for mountain huts

New Column Name	PyTransform
Accommodation URI	[base_uri]/accomodation_[accomodation.name]_mountainhut
Timetable URI	[base_uri]/timetable__[accommodation.name]_mountainhut
Schedule URI	[base_uri]/schedule_[schedule.name]_[accommdation.name]_mountainhut
Address URI	[base_uri]/[hash("mountainhut_TN"+accommodation.name))]
Address province	[base_uri]/#TN
Schedule name	opening

12 Knowledge Graph

12.1 Knowledge Graph exploration

The final knowledge graph covers well enough the domain on which the project focuses. From a dataset coverage point of view, most of the graph is populated with information from the data we have found and processed. In Figure 14 we can observe how the information about shops is distributed and linked in the knowledge graph, which is very similar to the one of the generic point of interest. The dataset is able to populate the graph fairly well, and queries successfully retrieve all the information needed.

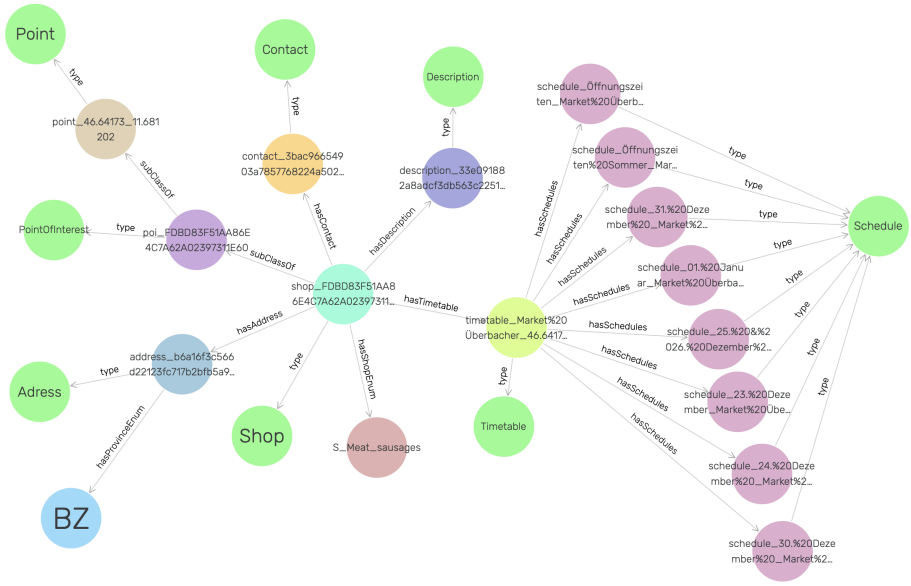


Figure 14: Knowledge graph, example of shop

In Figure 15 we can observe how the GTFS standard is implemented, considering that *Stop* is a subtype of *Point*. This is populated by GTFS data as explained before, making the graph very rich, returning lots of interesting results.

Accommodation is one of the subentities of *PointOfInterest* on which we focus most. The corresponding portion of the knowledge graph is very dense, and it is well populated of information from the datasets we found and processed. The mountain huts are also mapped onto this entity, with all the auxiliary ones involved and shown. Figure 16 shows the aforementioned portion of KG.

The mountain paths are mapped on the *ActivityPath* portion of the graph (Figure 17), with the *linestring* parent entity and the starting and ending points of interest. The opening time information was not available, so it was not populated, but the model can be extended as the ontology includes this possibility.

12.2 SPARQL

We used SPARQL in order to perform queries and retrieve data contained in our knowledge graph. We wrote some of the queries we described in Section 2 in order to verify the correctness of our result.

Now, we will show some queries that prove the integration process has been successful. Note that these are only some samples from the set of queries we prepared based on the competency question and formal queries, all of

Figure 15: Knowledge graph, example of stop

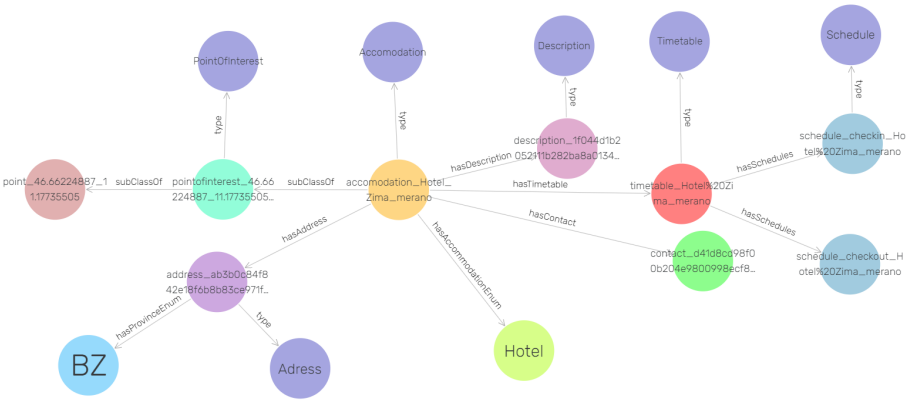


Figure 16: Knowledge graph, example of accommodation

which can be found in the project repository

12.2.1 Query 1.1

The query in the Listing 1 and described in Section 2 is interesting because we can see the use of the space domain and we retrieve the information from different datasets. In this query we want to find all the hotels near the station

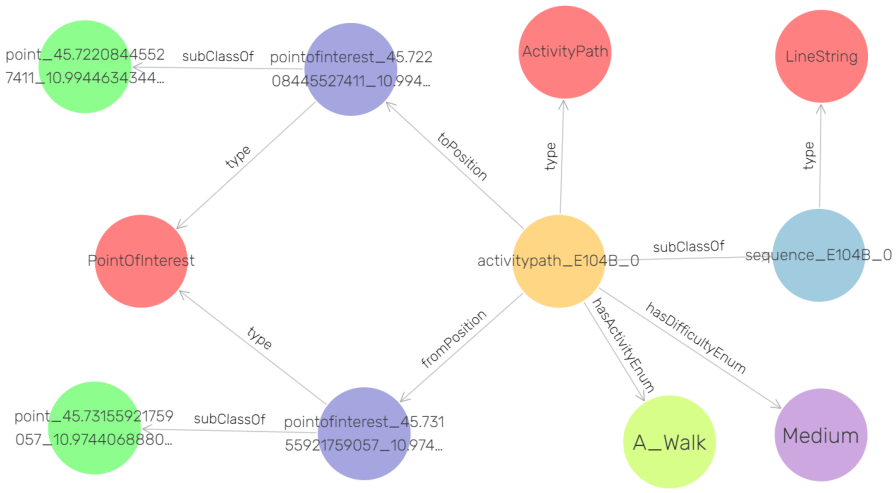


Figure 17: Knowledge graph, example of activity paths

of Bolzano. First of all, we need to find the station of Bolzano by searching in the Stop entity the station that is near the municipality of Bolzano, only after have found this station we can find all the hotels near the station. In order to calculate the proximity of two points, we use the function `omgeo:distance` that requires the latitude and the longitude of two points and returns the distance in kilometers. Using the distance returned by the function we can filter the results.

12.2.2 Query 4.6

The query shown in Listing 2 and described in Section 2 is interesting because we can see the use of Enum. In this query we want to find all the list of trekking paths ordered by increased difficulty and time. We want to order the results with respect to the difficulty, that is represented by an Enum, so we need to link each value of the Enum with a number. Then we can use the new values associated to the Enum to order the results using the `ORDER BY` clause.

Listing 1: SPARQL code of the Query 1.1

```
# Give the list of hotels near the station of Bolzano
PREFIX ontology:<http://www.semanticweb.org/aleca/ontologies/2019/10/untitled-ontology-10#>
PREFIX omgeo:<http://www.ontotext.com/owlim/geo#>

SELECT ?name ?latacc ?lonacc
WHERE {
    ?p a ontology:Stop .
    ?p ontology:StopName ?value .
    FILTER regex(str(?value), "stazione") .
    ?p ontology:Latitude ?lat .
    ?p ontology:Longitude ?lon .
    ?com a ontology:Municipality .
    ?com ontology:ISTAT_Municipality 'Bolzano' .
    ?com ontology:Latitude ?latc .
    ?com ontology:Longitude ?lonc .
    FILTER( omgeo:distance(?lat, ?lon, ?latc, ?lonc) < 2)
    ?acc a ontology:Accomodation .
    ?acc ontology:PoiName ?name .
    ?acc ontology:Latitude ?latacc .
    ?acc ontology:Longitude ?lonacc .
    ?acc ontology:hasAccommodationEnum ontology:Hotel .
    FILTER(omgeo:distance(?latacc, ?lonacc, ?lat, ?lon)<5)
}
```

Listing 2: SPARQL code of the Query 4.6

```
# Give me the list of trekking paths ordered by increased difficulty and time
PREFIX ontology:<http://www.semanticweb.org/aleca/ontologies/2019/10/untitled-ontology-10#>

SELECT ?name ?time ?diff
WHERE {
    ?path a ontology:ActivityPath .
    ?path ontology:PathName ?name .
    ?path ontology:hasActivityEnum ontology:A_Walk .
    ?path ontology:hasDifficultyEnum ?diff .
    values (?diff ?diff_) { (ontology:Low 1) (ontology:Medium 2) (ontology:High 3) }
    ?path ontology:AvgTravelTime_seconds ?time .
} order by ?diff_ ?time
```

13 Final consideration and open issues - Data Integration

The project we developed can be considered overall successful, as the result knowledge graph is well populated with data and can answer the competency question we initially planned. Furthermore, our ontology can be easily integrated with ICal and GTFS standards, allowing for further expansion with ease. On the other hand, the ontology is designed to be integrated with the upper ontology, with the upper one we proposed. Finally, the ontology can be extended from the enumeration types we used, as their role is to enclose the scope of our project while leaving a point of possible expansion by means of more specialized entities.