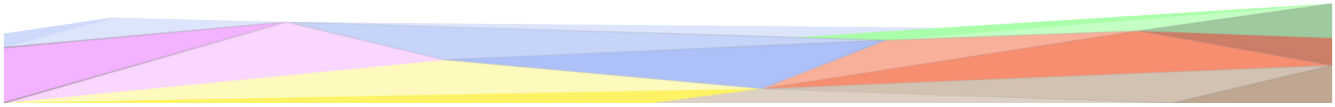
UNIVERSITY
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

KGE 2024 - Trentino Territory & Transportation

Document Data:	Reference Persons:
February 7, 2025	Mores Nicola, Roccon Marco



Index:

Revision History:

Revision	Date	Author	Description of Changes
0.1	October 21, 2024	Mores Nicola, Roccon Marco	Document created
0.2	October 30, 2024	Mores Nicola, Roccon Marco	Phase 1 - Purpose Definition
0.3	November 13, 2024	Mores Nicola, Roccon Marco	Phase 2 - Information Gathering
0.4	November 26, 2024	Mores Nicola, Roccon Marco	Phase 3 - Language Definition
0.5	December 04, 2024	Mores Nicola, Roccon Marco	Phase 4 - Knowledge Definition
0.6	December 19, 2024	Mores Nicola, Roccon Marco	Phase 5 - Entity Definition
0.7	February 07, 2025	Mores Nicola, Roccon Marco	Phase 6 - Evaluation, Metadata and Open Issues
1.0	February 08, 2025	Mores Nicola, Roccon Marco	Final version

1 Introduction

Reusability is one of the main principles in the Knowledge Graph Engineering (KGE) process defined by iTelos. The KGE project documentation plays an important role to enhance the reusability of the resources handled and produced during the process. A clear description of the resources as well as of the process (and single activities) developed, provides a clear understanding of the project, thus serving such an information to external readers for the future exploitation of the project's outcomes.

The current document aims to provide a detailed report of the project developed following the iTelos methodology. The report is structured as follows:

- Section 2: Definition of the project's purpose and its domain of interest.
- Section 3: Research and collection of data sources (at Data, Language and Knowledge level) that will provide us valuable resources for the creation of our final KG.
- Sections 4, 5, 6: The description of other iTelos process phases and their activities, divided by language, knowledge and data layer activities.
- Section 7: The description of the evaluation criteria and metrics applied to the project final outcome.
- Section 8: The description of the metadata produced for all (and all kind of) the resources handled and generated by the iTelos process, while executing the project.
- Section 9: Conclusions and open issues summary.

2 Purpose Definition

In this section we will cover the first phase defined by the iTelos methodology: The Purpose Definition. In this phase we aim to concretely define in a formal way the user's Purpose and what will be the information requirements that our Entity Graph will be able to satisfy. In order to do so, we will start from an Informal purpose, defining our Domain of Interest, and proceed with the creation of Personas, Scenarios. Using these we will define a set of Competency Questions (CQs), later used to identify the concepts (entities and properties) that we will work on and that are used to create an ER Model, the first purpose-specific version of the knowledge layer. Thus, at the end of this first step we will have a set of CQs, a set of identified concepts and an ER model that, all together, define our formal Purpose.

2.1 Informal Purpose

The first step to create an Entity Graph is the definition of a starting informal Purpose, stating, through a natural language sentence, the objective that drives us to the usage of the iTelos methodology.
In our case we want to create an Entity Graph containing information about transportation in the Trentino Region, focussed mainly on the city of Trento. In particular we want to extract not only data about Busses and Trains, but also regarding shared mobility alternatives (bike, scooter and car sharing), taxis, parking facilities and bike racks. In a more concise way, the informal purpose is:

> ”A person wants to move in an easy and efficient way through the Trentino region using public transports and other transport services available”

2.2 Domain of Interest

Having finalized our starting Purpose, we can also define the domain of interest in which our project will work and reason.
Our domain of interest will be the one of transportation services and, as stated in the informal purpose, from a spacial point of view, we will focus on the Italian region of Trentino, with special focus on it’s capital city: Trento.
Having delineated a first constraint on the space, we can also define one for the timespan we will consider: the project will focus on the currently available data about Trentino’s public transportation services, that covers a period of time around 10 months (from September 2024 to the end of June 2025).

2.3 Scenarios definition

In this section we define the set of Scenarios that will be taken into account during the project, showing the context in which our final users will act.
Every Scenario has been described in terms of a general description of the context and some possible needs that it may give rise to.

1. **Weekday:**
    - **Description:** It’s a weekday in Trento, with residents primarily commuting for work or study purposes. Buses and trains follow regular schedules, with commuters checking schedules to plan their movements.
    - **Needs emerged:**

  - Access to updated public transport schedules.
  - Travel planning to avoid peak hours.

2. **Weekend Excursion:**
   - **Description:** It's a weekend, and many residents take advantage of their free time to go on bike excursions around Trento. Public transport offers special services to carry bicycles.
   - **Needs emerged:**
     - Information on cycling racks available in Trento.
     - Schedules and regulations for bike transport on public transit.
3. **Holiday (Christmas):**
   - **Description:** During the Christmas season, celebrations lead to changes in public transport schedules. People use buses and trains to visit family and friends or participate in festive events in the city.
   - **Needs emerged:**
     - Information on special holiday public transport schedules.
4. **Rainy Day:**
   - **Description:** On a rainy day in Trento, residents prefer using taxis or car-sharing services to avoid walking in the rain. The demand for private transport increases significantly.
   - **Needs emerged:**
     - Access to information on the availability of taxis and car-sharing services.
5. **Cultural Event in the City Center:**
   - **Description:** A cultural event, such as a fair or festival, is taking place in Trento, attracting a large crowd to the city center. People seek to reach the event quickly and conveniently, so many decide to use electric scooters or bike sharing to avoid traffic and find bike parking easily.
   - **Needs emerged:**
     - Information on available bike-sharing racks and stations near the event.
     - Details on designated areas for rental scooters.
6. **Start of the Academic Year:**

- **Description:** At the start of the new academic year, new university students move to Trento and search for apartments. In their search, they consider not only price and availability but also how well-connected the area is to university departments and daily activities such as supermarkets and gyms.
- **Needs emerged:**
  - Access to information on the proximity of public transport stops in a certain location.
  - Details on public transport routes between two areas in the city.

7. **Graduation Day:**

   - **Description:** It's graduation day at the University of Trento. Family and friends of graduates come to the city to attend the ceremony, causing a significant increase in traffic and high demand for parking. Drivers are looking for information on available parking near the university and alternative parking options.
   - **Needs emerged:**
     - Access to information on available parking near the university.
     - Directions on how to reach the ceremony location by public transport from identified parking areas.

2.4 Personas definition

In this subsection we will define a set of Personas: Fictional actors involved in the project domain, characterizing user's needs and perception, and that will act in the previously defined Scenarios.
Below we describe our Personas, stating for each of them a brief description of their lives and what their needs and goals are:

1. **Sara:**

   - **Occupation:** University student
   - **Age:** 23 years
   - **Description:** Sara lives downtown near the station and regularly attends classes in the Department of Economics. Without a car, she uses buses and trains for her travels, and occasionally bike sharing.
   - **Needs/Objective:**
     - Find fast and direct routes to the campus.
     - Check for any unavailability of transportation.

2. **Marco**:
    - **Occupation:** University student and amateur cyclist
    - **Age:** 22 years
    - **Description:** Marco studies law in the city center and prefers to cycle when the weather is nice. He lives just a few minutes from the department, but on rainy days he prefers public transport.
    - **Needs/Objective:**
        - Discover the locations of public bike racks.
        - Find public transport alternatives in case of rain.
3. **Luisa**:
    - **Occupation:** Commuter
    - **Age:** 32 years
    - **Description:** Luisa works in the center of Trento but lives in a nearby town. Luisa suffers from motion sickness whenever she has to work on the computer during her travels. For this reason, and to have more space, she prefers the train for her daily commutes.
    - **Needs/Objective:**
        - Check train schedules and verify availability, even on holidays.
        - Plan trips that minimize wait times between trains.
4. **Giovanni**:
    - **Occupation:** Business executive
    - **Age:** 48 years
    - **Description:** Giovanni has frequent appointments in various parts of the city. He needs to move quickly and efficiently, often working while on the go, and for this reason, he prefers taxis.
    - **Needs/Objective:**
        - Know the nearest taxi parking in useful areas.
5. **Andrea**:
    - **Occupation:** Out-of-town student
    - **Age:** 24 years

- **Description:** Andrea and his fellow out-of-town students regularly organize weekend trips. They do not own a car, so they rent car-sharing vehicles for longer trips.
- **Needs/Objective:**
  - Find designated car-sharing zones and check the availability of vehicles for day trips.

6. **Helmut**:
   - **Occupation:** Tourist
   - **Age:** 36 years
   - **Description:** Coming from a nearby country, Helmut arrives by bike and wants to explore the city center without bringing it with him.
   - **Needs/Objective:**
     - Identify public bike racks and secure bike parking.
     - Check if the trains connecting his country to the city center have appropriate racks for transporting bikes.
7. **Francesca**:
   - **Occupation:** Employee
   - **Age:** 56 years
   - **Description:** To attend her son's graduation in the city center, Francesca drives there, but since she is not from the area, she doesn't know the locations of nearby parking.
   - **Needs/Objective:**
     - Find public and paid parking available for extended stays.
     - Know if the parking areas are accessible and close to the ceremony location.
8. **Davide**:
   - **Occupation:** University student
   - **Age:** 21 years
   - **Description:** Davide uses electric scooters to get around in the evening when public transport is less frequent. He is also a frequent user of bike sharing, having recently lost his own bike.
   - **Needs/Objective:**
     - Know the location of scooter and bike-sharing racks.
     - Find quick and flexible transport solutions during the evening hours.

9. **Anna**:
    - **Occupation:** University student
    - **Age:** 19 years
    - **Description:** Anna is a student with reduced mobility who moves in a wheelchair. She lives in the suburbs and attends university in the city center, so she relies on public transport for her daily travels. She often needs to check bus arrival and departure times and ensure they are accessible.
    - **Needs/Objective:**
        - Verify the accessibility of buses on urban routes and if the service is active during a specific time frame.
        - Know in advance if a stop is accessible in a wheelchair and if there are detours or route changes that could affect her mobility.

2.5 Competency Questions (CQs)

Now that we have defined Personas and Scenarios we can proceed extracting the KG functional requirements, defining ours Competency Questions. Each of them will refer to one of the Personas acting in one of the Scenarios previously enumerated, and will be used to identify the questions that our EG, once completed, will be able to answer.
Below are the CQs identified, grouped by the Scenario they are referring to:

1. **Weekday:**
    1.1 **Sara:** Which buses and trains can I take to go from Trento Station to the Department of Economics between 8:00 and 9:00?
    1.2 **Marco:** What is the arrival time of the next bus from the ”Povo Valoni” stop heading downtown?
    1.3 **Anna:** Which stops on bus line 5 are wheelchair accessible?
2. **Weekend Excursion:**
    2.1 **Marco:** Given a point on a map with its coordinates, what is the nearest stop to it?
    2.2 **Helmut:** Which train routes allow bicycle transportation during the weekend?
    2.3 **Helmut:** How many bikes can be parked in the rack closest to my current location?
    2.4 **Giovanni:** How many parking spots are available in the car-sharing station near Piazza di Fiera?
3. **Holiday (Christmas):**

3.1 **Luisa:** How do holiday schedules for busses and trains change on Christmas Day?

3.2 **Sara:** If I get on bus line 5 at “Povo Salé” stop at 12:03, when can I expect to arrive at the “Port’aquila” stop?

3.3 **Anna:** Are there any routes on the “P.Dante Rosmini S.Rocco Povo Polo Soc.” line that go directly to “Povo Polo Sociale”?

4. **Rainy Day:**

4.1 **Andrea:** Which car-sharing stations are closest to the San Bartolomeo area?

4.2 **Giovanni:** Where can I catch a taxi near Piazza Duomo?

4.3 **Marco:** On line 7, how many stops are there from “Gorizia Adamello” to “Gocciadoro Arcate”?

4.4 **Francesca:** Having just washed my car, where can I find an underground parking garage to avoid the rain?

5. **Cultural Event in City Center:**

5.1 **Davide:** Where can I find electric scooter stations near Piazza Fiera?

5.2 **Davide:** How many rental bikes can be parked in the station near the city center?

5.3 **Marco:** Where can I find a bike rack with frame locks near Piazza Duomo?

5.4 **Anna:** Which runs of bus line 5 heading downtown are wheelchair accessible?

6. **Start of the Academic Year:**

6.1 **Anna:** Which bus and train stops are available within a 500-meter radius of my apartment in the Santa Chiara area?

6.2 **Andrea:** How many stops separate the area of Piazza Dante from the Department of Economics on public transport lines?

6.3 **Luisa:** Which organization manages public transportation services in the city of Trento, and how can I contact it?

7. **Graduation Day:**

7.1 **Francesca:** What is the average maximum capacity of public parking spaces within a 1 km radius of the Department of Medicine at the University of Trento?

7.2 **Francesca:** What are the opening hours of the “Piazza di Fiera” parking lot?

7.3 **Andrea:** How many free public parking spots are there in Povo?

2.6 Concepts Identification

Having defined the CQs, we can proceed with the following step: Concept Identification. During this step we will extract the concepts identifying Entity Types (ETypes) and properties that will be modelled in our KG. To do so we will take into account the previously defined purpose and also the data layer, in terms of data sources availability. The final result of this step will be a Purpose Formalization Sheet (PFsheet), a dedicated spreadsheet combining Knowledge and Data Layer.

The following table shows the PFsheet we can generate from the Personas and Scenarios described in the previous sections. Each row contains one Entity and its corresponding properties, stating from which Personas, Scenarios and CQs these concepts have been extracted from.

In order to enhance the reusability, flexibility and quality of our future EG we will also consider well-known schema providers, such as SCHEMA.org, trying to find a proper mapping between their resources and our concept vocabulary whenever possible.

Finally we classified each entity with respect to its Focus, a parameter used to represent how much a concept is relevant to one's purpose, more concretely it can assume one of these values: Common, universal and commonly used concepts; Core, essential concepts for our domain; Contextual, highly specific concepts of our context and thus, usually, less reusable.

Scenarios	Personas	Competency Questions	Entities	Properties	Focus
6	3	6.3	City		Common
6	3	6.3	Organization	name, telephone	Common
1, 2, 4, 5, 6, 7	1, 2, 4, 5, 6, 7, 8, 9	1.1, 2.1, 2.3, 2.4, 4.1, 4.2, 5.1, 5.2, 5.3, 6.1, 6.2, 7.1, 7.3	Point	latitude, longitude	Common
1, 2, 3, 4, 5, 6	1, 2, 3, 6, 9	1.1, 1.2, 1.3, 2.2 3.1, 3.2, 3.3, 4.3, 5.3, 5.4, 6.2	Line (Bus/Train)	shortName, longName, type	Contextual
2, 3, 5, 6	1, 2, 6, 9	1.1, 1.2, 2.2, 3.1, 3.3, 5.4, 6.2	Trip	direction, headsign, bikeSlots, wheelchair	Contextual
1, 3, 4, 5, 6	1, 2, 8, 9	1.1, 1.2, 1.3, 3.2, 3.3, 4.3, 5.2, 6.1, 6.2	Stop (Bus/Train)	name, wheelchair, pos	Contextual
1, 3, 4, 6	1, 2, 5	1.1, 1.2, 3.2, 4.3, 6.2	Stop Event	arrivalTime, departureTime, stopSequence	Contextual
2, 5	2, 6	2.3, 5.3	Bike Rack	pos, capacity, type	Core
1, 2	1, 2, 6	1.1, 2.2	Schedule	byDay, validity(start end date)	Contextual
3	3	3.1	Special Schedule	date, type(variazione del servizio)	Contextual
2, 4	4, 5	2.4, 4.1	Car Sharing Station	pos, capacity	Core
4	4	4.2	Taxi Station	pos	Core
5	8	5.1	Scooter Sharing Station	pos	Core
5	8	5.2	Bike Sharing Stations	pos, capacity	Core
4, 7	5, 7	4.4, 7.1, 7.2, 7.3	Parking Lot	name, capacity, pos, type, openingHours isAccessibleForFree	Core

Figure 1: Purpose Formalization sheet

2.7 ER model definition

The last step of this initial phase, that will lead us to a complete formalization of the purpose, is to design an Entity Relation (ER) model using the concepts previously obtained. This ER model will be our first version of the final knowledge layer.
In order to create the model we will use the IDEF1X Notation ERD, a notation that will allow us to define entities and their attributes more precisely, compared to the traditional ERD, thus obtaining a clearer representation. To illustrate more clearly the focus level previously assigned to each entity, in the following diagram we will also use various colours to highlight the different levels: Red for Common ETypes; Green for Core ETypes; Blue for Contextual Etypes.

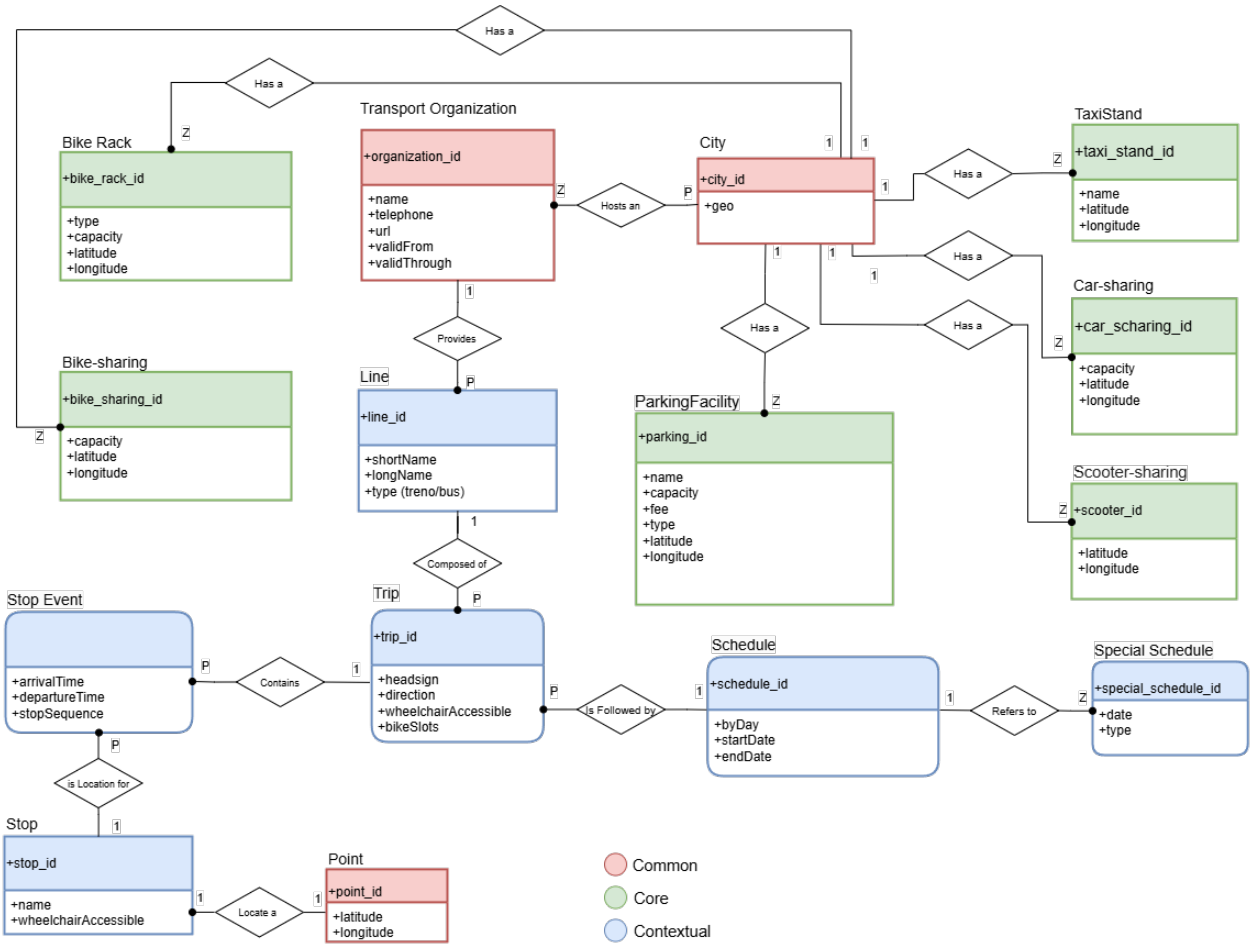


Figure 2: The ER model in IDEF1X Notation

3 Information Gathering

In this section we will cover the second phase of the iTelos methodology. Having formally defined the purpose of our project, we can focus on the collection on resources valuable for the creation of a EG able to satisfy the purpose. These resources includes dataset about every information's layers: Data, Knowledge and Language datasets. Moreover, during this chapter, we will also enhance their quality and reusability, by means of cleaning from unnecessary noise and standardizing them.

3.1 Source Identification

The first step of this phase covers the identification of the sources taken into account while gathering information. Among the many possible sources we can classify them in two main groups, based on their quality: High quality data sources contains distributed dataset characterized by high interoperability and reusability; On the other hand, Low quality sources' dataset are less standardized and with poor metadata, thus being less interoperable and understandable.

During our research we have founded these information resources:

- **Data value Datasets**
  - OPENdata Trentino: Open data Trentino is a platform, managed by the autonomous province of Trento, that offers an unique catalogue of reusable data and allows the search, access and download of open data about Trentino and services in its territory. We will use this platform to gather many of our datasets about the public transportation services and more. This source's dataset have really high variability in terms of quality: some of them are easily accessible (through APIs or direct download) and uses high quality, open license and non-proprietary data formats that support machine-readability, such as JSON and CSV; on the other hand many of their resources are occasionally accessible or not at all, while other are provided in formats such as PDF, making really hard to reuse them.
  - OpenStreetMap: OpenStreetMap is a collaborative project that provides free and editable maps of the world, also maintaining geographic data about roads, trails, railway stations, and so on. As an open-source and widely recognized resource, we will leverage OpenStreetMap to obtain additional information about parking lots, trainline and cycle path in the Trentino region, enriching our project and allowing us to cover also these part of the previously defined domain.
  - Trentino Trasporti Website: Trentino Trasporti is the main public transportation service provider in the city of Trento. In their website are provided information about the train

and bus lines available.

- **Knowledge Datasets:**
  - SCHEMA.org: ”Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond”. Being this a well-known and standardized resource, we will extract some reference schemas from it, in order to help us during the modelling of the needed ETypes. Proceeding in this way we can make our project more understandable and easily interoperable.
  - GTFS.org: ”The General Transit Feed Specification, also known as GTFS, is a standardized data format that provides a structure for public transit agencies to describe the details of their services such as schedules, stops, fares, etc”. As a widely adopted standard for transit data, GTFS will be integral in defining and modelling transportation-related entities, routes, stops, schedules, and calendars within our project.
- **Language Datasets**
  - Universal Knowledge Core (UKC): ”The Universal Knowledge Core (UKC) is a multilingual, high quality, large-scale, machine-readable, and diversity-aware lexical resource”. This will be our main resource at the language level, providing a formalization of many of the purpose-specific concepts founded during the project.
  - SCHEMA.org: In addition to the previously cited schemas and ontologies, Schema.org provides also informations about the language layer. As a matter of facts, each schema is provided also with a brief explanation of the meaning of the concepts introduced by itself, together with a description of each of its properties.
  - OpenStreetMap Wiki: OpenStreetMap, through their wiki page, provides also high quality dataset at language level, offering many formalized definitions of concepts about transportation services and streets.

3.2 Dataset Collection

In this subsection we will list the various dataset extracted from the previously cited data sources, showing for each of them their content and giving a brief note about their quality whenever needed.

- **OPENdata Trentino:**
  - Trasporti pubblici del Trentino: This dataset contains data about Trentino Trasporti’s urban and suburban lines. The data provided for the two types of lines follows

the GTFS standard and is similar, except for the missing ”Shapes.txt” file and the ”wheelchair_accessible” property in the ”Trips.txt” and ”Stops.txt” files for the suburban lines.. In particular, inside of this dataset we can find various files:

  * Agency.txt - This file contains data about Trentino’s Trasporti, the main public transportation provider in the city of Trento and in its region.
  * Calendar.txt - Contains data about the frequencies of a particular service, showing whether it is provided in each day of the week or not and also the timespan validity of each entry.
  * Calendar_dates.txt - Provides information about exceptional dates, showing how the service may change during these days.
  * Routes.txt - Contains information regarding the various line available. In particular, we can find their names (in a brief or more precise manner), their type (busses or trains), identifying colour and about which agency provides them.
  * Shapes.txt - Provide data showing the physical route followed by the vehicle during their trips, by means of sequences of points, characterized by their latitude and longitude.
  * Stops.txt - This file contains data about public transportation’s stops, including their names, descriptions, positions, zones and information regarding wheelchair accessibility.
  * Stop_times.txt - Contains information about the arrival/departure time of each trip at every stop station, also showing the sequence in which these stops will occur.
  * Transfers.txt - Provide data regarding the intersection point between different lines, showing if it is possible for the user to leave their current transportation vehicle to get on another one.
  * Trips.txt - Shows information about the various lines’ trips that occurs during a specific service’s timespan. For each of them data about the head-sign of the trip, its direction, its physical shape and wheelchair accessibility are provided

- Taxi: In this dataset are contained information regarding the various taxi stands located in the city of Trento, such as their position (both in WKT coordinate system standard, lat/long and street name) and the stand’s name.
- Car sharing: Through this dataset we can extract data about the parking lot used for the car sharing service in Trento. For each lot is provided its position (WKT and street name), its capacity and its decree number.
- Bike Sharing: This dataset provides information about bike racks used for bike-sharing services. For each rack we have its id, a brief description, location, capacity and type.

- C'entro in bici: Similarly to the previous dataset, this one provides details of bike racks designed for the "C'entro in bici" bike sharing public project. For each of them its provided the position (WKT), a brief description and the capacity.
- Rastrelliere per biciclette: In this dataset we can find details regarding public bike racks available in Trento, in particular we have their position (WKT, zone in the city, street name, street number and nearby palaces), type, capacity, year of installation, number of modules, photo, and category. However, this information aren't always available for every bike rack and many of them have one or more missing values.
- Parcheggio protetto per biciclette: This dataset focuses on bike racks that are protected, usually within a covered area or warehouse. For every rack we have its position (WKT, park and street name), a brief description and its capacity.
- Punti sosta monopattini condivisi a tariffa agevolata: In this dataset are stored data about scooter sharing stations. In particular there are information regarding location (WKT and street name), id, decree, reference code and additional notes.
- Itinerari ciclabili esistenti: This dataset provides information about the existing bike paths, showing for each of them their shape, type, year of construction, itinerary and decree.

- **OpenStreetMap:** In order to extract data from OpenStreetMap, we will use Overpass turbo, a tool that allow to easily query OpenStreetMap, through its APIs, and to show the obtained results directly on a interactive map. The query used to gather the following dataset can be found in our Github Repository.
  - Trento's Parking Lot: Using OSM, we can get details about parking lot in the city of Trento. These lots can be characterized by many different properties, such as: id, position (latitude, longitude), accessibility, capacity, information of eventual owners, fees, name, opening hours and information about the floor they are located on; However, most lots included only subsets of these properties, with smaller ones having often having just a few of them.
  - Bycicle path in Trentino Alto Adige - Sudtirol: From a specifically crafted query, we can gather information about the bicycle paths in the Trentino region. For each path, details on shape, walkability, surface type, and lighting are provided.
  - Trentino's Train lines' shapes: With a query we can gather information about the railways in Trentino, with details on shape, maximum allowed speed, name, lines and whether they are electrified.

This collection includes all the dataset we gathered during the current phase and can also be found on our Github Repository. To confirm that these resources are sufficient, we must ensure

they cover the list of the Competency Questions defined in the previous phase. Since they do, we can proceed with the next step.

3.3 Dataset Cleaning and Standardization

During the last step of this second phase, we will focus on removing any noise in the dataset collected so far. Specifically, we will remove those datasets, entities and properties that don't align to our formal purpose and modify the remaining ones to fit our goals. Note that the original datasets were primarily in Italian, so we will translate properties and values into English as needed. After that, we will convert every file to the same standardized format: CSV.

Follows a more in detail description of how we concretely modified the datasets:

- Trasporti pubblici del Trentino: For what concerns the Trentino Trasporti dataset, we decided to maintain all the files except for transfers.txt, stopslevel.txt, shapes.txt and feed_info.txt, this apply for both urban and suburban. The remaining files contain mostly useful information in an already well known standardized format (GTFS). For this reason only few small changes has been applied:
  - routes.txt: Removed the fields route_color and route_text_color.
  - stops.txt: Removed the fields stop_code, stop_desc, zone_id and renamed wheelchair_boarding as wheelchair_accessible, stop_lat in latitude, stop_lon in longitude.
  - trip.txt: Added the field bikeSlots, stating how many bicycle slots are available to the trip passengers. Busses will have a default value of 0; trains of 6. This information has been gathered from this web page.
  - calendar.txt: Instead of having a column for each day of the week, we decided to structure the information regarding the days of the week in which the line operates organized as a list (ex. [tuesday, friday]) as a single column, called byDay.
  - *.txt: Removed references to not used.
- Taxi: For what concern the Taxi Stands we decided to maintain only their names and positions. The initial file provided three different properties for the position: WKT, containing a UTM-32 Point, a particular coordinate system, and two others called x and y, expressing the position using latitude and longitude. Being these properties redundant, we decided to keep only the latter two. Follows the list of changes applied to this dataset:
  - id: Not present in the original file, so a hand-crafted one was introduced.
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento's city, was added.

  - x: Renamed to “latitude” in order to increase the readability.
  - y: Renamed to “longitude” in order to increase the readability.
  - nome: Renamed to “name” in order to increase the readability.

  The final CSV file is composed of: id, city_id, name, latitude and longitude.

- Car sharing: For what concern the Car sharing stations, we decided to maintain only some properties: via, auto and WKT, although some changes has been applied as follow:
  - id: Not present in the original file, so a hand-crafted one was introduced.
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento’s city, was added.
  - WKT: Converted from a UTM (32) Point to EPSG:4326 format, dividing it in two new properties “latitude” and “longitude”.
  - via: Renamed to description. Originally contained the name of the street the station is located and a brief description of the place. Being the street address redundant, we decided to maintain only the description.
  - auto: Renamed to capacity. Used to indicate the number of cars available in every car sharing station, so it has been renamed in order to increase the readability.

  The final CSV file is composed of: id, city_id, description, capacity, latitude and longitude.

- Bike Sharing: For what concern the Bike-sharing stations, we decided to maintain only some properties: id, desc, ciclopositeggi and WKT, although some changes has been applied as follow:
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento’s city, was added.
  - WKT: Converted from a UTM (32) Point to EPSG:4326 format, dividing it in two new properties “latitude” and “longitude”.
  - ciclopositeggi: Renamed to capacity. Used to indicate the capacity of every bike sharing station, so it has been renamed in order to increase the readability.
  - desc: Renamed to description.

  The final CSV file is composed of: id, city_id, description, capacity, latitude and longitude.

- C’entro in Bici: For what concern the second bike sharing dataset, we decided to maintain only some properties: desc, ciclopositeggi and WKT, although some changes has been applied as follow:

- id: Not present in the original file, so a hand-crafted one was introduced.
- city_id: Not present in the original file; a custom value, equal to the ID of Trento's city, was added.
- WKT: Converted from a UTM (32) Point to EPSG:4326 format, dividing it in two new properties "latitude" and "longitude".
- ciclopostegi: Renamed to capacity. Used to indicate the capacity of every bike sharing station, so it has been renamed in order to increase the readability.
- desc: Renamed to description.

The final CSV file is composed of: id, city_id, description, capacity, latitude and longitude.

- Rastrelliere per biciclette: For what concern the bike racks dataset, we decided to maintain only some properties: id, Tipo_generale, tot_bici and WKT, although some changes has been applied as follow:
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento's city, was added.
  - WKT: Converted from a Linestring, that is a list of UTM (32) Points, to EPSG:4326 format, dividing it in two new properties "latitude" and "longitude" and assigning them the average value of the initial one.
  - Tipo_generale: Renamed to type. Used to indicate the bike rack's type, whose possible values were "tradizionale" (traditional) and "bloccatelaio" (frame lock), so it has been renamed in order to increase the readability.
  - tot_bici: Renamed to capacity. Used to indicate the capacity of the bike rack, so it has been renamed in order to increase the readability.

  The final CSV file is composed of: id, city_id, type, capacity, latitude and longitude.
- Parcheggio protetto per biciclette: For what concern the second bike racks dataset, we decided to maintain only some properties: posti and WKT, although some changes has been applied as follow:
  - id: Not present in the original file, so a hand-crafted one was introduced.
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento's city, was added.
  - WKT: Converted from a UTM (32) Point to EPSG:4326 format, dividing it in two new properties "latitude" and "longitude".

  - type: Not present in the original file, so a hand-crafted one was introduced, with a fixed value of “guarded”. This field was added in order to allow us to merge the two dataset while having a common set of fields among them.
  - posti: Renamed to capacity. Used to indicate the capacity of every bike sharing station, so it has been renamed in order to increase the readability.

  The final CSV file is composed of: id, city_id, type, capacity, latitude and longitude.

- Punti sosta monopattini condivisi a tariffa agevolata: For what concern the scooter sharing service dataset, we decided to maintain only some properties: id, note and WKT, although some changes has been applied as follow:
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento’s city, was added.
  - WKT: Converted from a UTM (32) Point to EPSG:4326 format, dividing it in two new properties “latitude” and “longitude”.
  - note: Renamed to description. Used to provide additional information that could help locate the Scooter Sharing locations, so it has been renamed in order to increase the readability.

  The final CSV file is composed of: id, city_id, description, latitude and longitude.

- Trento’s Parking Lot: For what concerns the parking lot JSON dataset, fetched from OpenStreetMap, we decided to maintain only some properties: id, parking, access, capacity, fee, coordinates, although some changes has been applied as follow:
  - city_id: Not present in the original file; a custom value, equal to the ID of Trento’s city, was added.
  - access: This property defines the accessibility of the parking, for the purpose of this project we deleted all the private ones. Those were parking spots inside private property that are not of our interests.
  - coordinates: Renamed to latitude and longitude. The coordinates were in the right format but contained in a unique property, so we split it.
  - parking: Renamed to type. Used to indicate the type of every parking lot, so it has been renamed in order to increase the readability.
  - capacity: Used to indicate the capacity of every lot. Since this property wasn’t specified for every spot, a default value of -1 has been set whenever needed.
  - fee: Used to indicate the fees required to use a parking lot. Since this property wasn’t specified for every spot, a default value of “no” has been set whenever needed.

The final CSV file is composed of: id, city_id, access, fee, capacity, type, latitude and longitude

Note that any dataset mentioned in the Collection phase but missing from the list above has been removed due to discrepancies between the data provided and the data actually required for our purposes.

The results of this cleaning process can be found in our Github Repository, along with the Python script used to alter the various dataset.

4 Language Definition

This section will focus mainly on resources at Language layer: We will identify the purpose-specific concepts meaningful for our project and then formally state their definitions. Through these steps, at the end of this phase, we will produce a purpose-specific language file.

4.1 Concept Identification

The first activity of this Language phase focuses on defining the language resources for our project. To achieve this, we will identity every concept representing ETypes, properties and data properties values that will be used in the final Knowledge Graph to represent information. These concepts will be extracted from the results obtained in the previous iTelos phases, specifically from the ER model, PFSheet (produced during the first phase), and the resources identified at Language, Data and Knowledge layers (collected during the second phase).

Once identified, we will focus on determining the formal meaning of these concepts, either by finding an existing formal definition or defining one ourself whenever needed. To find these definition, we will leverage already existing resources, with a particular focus on the Universal Knowledge Core (UKC). Specifically, we will explore the UKC to search for the previously identified concepts and use the definitions provided (referred to as gloss) when they align with our objectives. If they do not meet our needs, we will consider other language resources among the ones identified in the Dataset Collection step. If no suitable definition can be found, we will create a formal definition ourselves. This will happen mainly for highly domain-specific concepts, such as the ones related to ETypes classified as Contextual in the PFSheet.

The final step regards the creation of the file representing the language resources meaningful for our purpose. This file will be structured as a table containing the following columns:

- **Concept ID:** Contains the ID identifying each concepts. When mapping to UKC Concepts is available, we will use the corresponding UKCIdentifier; For other resources, such as OpenStreetMaps, we will use the URL of the web page containing the concept and its term;

Lastly, when we need to create a definition ourselves, the corresponding concept will use an incremental id in the range assigned to our project, which is KGE24-1. In this way, the IDs will follow the format KGE24-1-<concept_id>, where concept_id is a numerical value obtained incrementing the previous value by 1, starting from 1.

- **Concept label:** Contains the word we are formally defining.
- **Concept description:** Contains the formal definition and meaning of each concept.

During the creation of this file, we considered including details on labels and description in multiple languages: English, as it is the de-facto world-wide standard language; Italian, since Trento is located in Italy and Italian is the main language used in the territory we are working on; German, as it is the third most spoken language in Trentino. German was also taken into account due to the significant number of German tourists who frequently visit Trento, as highlighted during the definition of personas in the first phase of the project. However, for the purpose of this course project, we decided to not include the information in German, mainly because neither of the two team members speak this language. Using automatic translations for such resources, which require exceptional accuracy to avoid misleading users does not seem like an optimal solution. Note that, in most cases, the translation of both words and glosses has been done manually.

Below is an image partially showing our language resource file, while the whole version is available in our GitHub Repository

ConcetID	Word-en	Gloss-en	Word-it
UKC-21898	Train stop, Train station (UKC)	Terminal where trains load or unload passengers or goods	Fermata del treno
UKC-45118	Bus stop	A place on a bus route where buses stop to discharge and take on passengers	Fermata dell'autobus
UKC-24387	Transit line	A line providing public transit	Linea di trasporto
KGE24-1-1	Train line	Part of a transportation system that provides, by means of a train, transportation from a fixed position to another following a predefined railroad	Linea ferroviaria
KGE24-1-2	Bus line	Part of a transportation system that provides, by means of a bus, transportation from a fixed position to another following a predefined path	Linea degli autobus

Figure 3: Section of the Language Resource file

4.2 Dataset Filtering

In this second activity, we will focus on the data layer of our final Knowledge Graph. Specifically, we will align the data-level resources previously collected with the concepts that we have just formalized, filtering out every data element not defined by any of these concepts.

In our case, every EType, attribute and property has a direct mapping to the language resource, so none of them will be removed.

5 Knowledge Definition

Having formalized the language resources, in this section we will focus on the Knowledge layer, aiming to develop a knowledge teleontology for our project. Additionally, we will also align the dataset collected in the phase 2 to ensure that they match the modelling choices defined by the teleontology. This alignment will result in a dataset structure consistent with the teleontology, unifying in this way the representation of the information collected so far.

5.1 Modeling a Knowledge Teleontology using kTelos

The first step of the Knowledge Definition phase aims at modelling the previously collected knowledge in a knowledge teleontology, following the kTelos process. To achieve this, we will leverage the language resources, defined during the third phase, which will help us in modelling both ETypes and properties. Specifically, we will start by creating a hierarchical structure of ETypes using an IS-A hierarchy, selecting terms from the language resource that denotes entity types. Once the ETypes are defined, we will focus on object properties, identifying terms that denotes relationships between ETypes and defining them as object properties, including their domain and range. A similar approach will be applied to the definition of data properties, ensuring a comprehensive and structured representation of the knowledge, completing in this way our knowledge teleontology file.

To concretely apply this process, we will use Protégé, ”A free, open-source ontology editor and framework for building intelligent systems”. Specifically, we created the hierarchical structure of ETypes within the Classes section, adding various Annotations for each class:

- **rdfs:label:** A standard annotation where we specify the EType name.
- **rdfs:comment:** A standard annotation where we inserted the Gloss-en defined in the Language Resource File.
- **conceptID:** A custom annotation created by us, representing the ID of the concept. This follows the values and formats specified in the Language Resource File. Since we do not have multiple classes with the same name referring to different concepts (and thus different ETypes), we decided not to include the conceptID directly in the concept names. Instead, we added it as an annotation, ensuring a clearer and more organized final result.

- **isEtype:** Another custom annotation created by us, denoting that the class is indeed an EType.

Note that for the annotations ”rdfs:label” and ”rdfs:comment”, since these were written in English, we also added the language tag, setting it to ”en”.

Then, as specified by kTelos, we proceeded with the definition of our Object properties, specifying for each of them their Domains and Ranges, which represent the ETypes involved in the relationship. During this process, one of our object property, ”has_a”, was used to link multiple pairs of ETypes. This was feasible because each of these relationships connected the City EType to another EType, and the semantic meanings of this link remained consistent across all the cases.

The final step to create our knowledge teleontology involved the Data properties. This process was similar to the one followed for the object properties, as we specified both the domain and range for each property. However, their meaning differed from the previous one: the Domain indicates the ETypes to which the property is relevant; while the Range defines the allowed value types for the property. Specifically, we selected from a limited subset of data types provided by Protégé: xsd:boolean, xsd:int, xsd:float, xsd:string and xsd:dateTime.

This process helped us defining our initial teleontology by formalizing the informal ER model developed during the first phase into an OWL file. This initial version can be found in our Repository.

The next step involved the integration of existing external ontologies, identified during the second phase, with the goal of aligning them with our teleontology. Specifically, we focused on the ontology provided by Schema.org, leveraging its well-established structure and concepts. During this process, we analyzed the entities and properties in our teleontology to identify:

- **Equivalent concepts:** When a concept in Schema.org fully matched one in our teleontology, we annotated the corresponding class in our teleontology with a custom annotation ”equivalentTo” to explicitly align it with the Schema.org concept.
- **Hierarchical relationships:** When we identified Schema.org concepts representing a more general version of one of our classes, we linked them using the ”SubClass Of” relationship.

Additionally, we reviewed our data properties to improve their alignment with Schema.org’s structure. Specifically, we migrated data properties initially associated with one of our classes to a Schema.org class introduced during this step. This adjustment was particularly relevant when the latter stood in an ”Is-A” relationship with our original class, ensuring better semantic alignment and interoperability.

One of the most significant result of this step was the introduction of a new class, ”Place”, derived from Schema.org, which will be the parent of many previously defined ETypes. Due to

this relationship, we also migrated the ”latitude” and ”longitude” data properties shared among these subclasses to their new parent. Another important change introduced with this new class was the removal of ”Point”, since it was equivalent to Place. Consequently, our old EType was no longer relevant and was eliminated. Furthermore, we updated our Language Resource file to indicate that the concept ”Place” from Schema.org and ”Point” from our Competency Questions will be treated as synonyms.

The final teleontology obtained through this process can be found in our Repository.

5.2 Schema Alignment

The following step of this Knowledge definition phase is Schema Alignment. During this phase we will produce another OWL file representing our final teleology, obtained by aligning the informal ER model to our teleontology. Schema Alignment is a critical step, as it ensures that the teleology reflects the purpose defined for the iTelos iteration. Typically, this phase require a complex process, often involving machine learning models, to ensure a precise alignment between schema components. However, due to the time constraint of the course, we will limit our workflow to the following steps:

1. **Identifying leaf ETypes:** We identified the most specific entity types in the hierarchy (the leaf nodes), for which we have data.
2. **Dropping general ETypes:** General ETypes higher in the hierarchy that are not directly linked to available data will be removed from the schema.
3. **Inheriting purpose-specific properties:** Whenever applicable, object and data properties defined for the removed ETypes will be transferred to the relevant leaf entity types.

Following these steps, we created our reference Teleology, which is accessible in our Repository, while below is an image showing its classes:

5.3 Teleology Validation

Before proceeding with the next phase, we performed a validation of the teleology developed so far, ensuring its alignment with the initial Competency Questions (CQs). This step resembles the validation carried out during the first phase when we verified that the informal ER model was capable of addressing the CQs. The results of that process were satisfying and provides a strong foundation for comparison, alongside the PFSheet created during the same phase. In our teleology, all 15 ETypes are present and consistent with the entities identified in the ER model and PFSheet. However, minor differences emerged when comparing with the ER model, particularly due to the redefinition of the ”Point” EType as Place (based on Schema.org’s ontology).

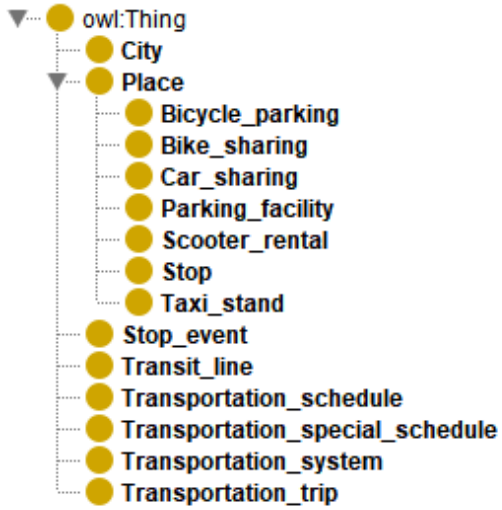


Figure 4: Teleology's classes

While this adjustment slightly diverges from the ER model, it does not compromise the ability to answer the CQs. Each EType in the teleology is equipped with the required object and data properties to address the CQs, and the relationships defined among ETypes fully align with the core requirements of the CQs. The validation confirmed that the teleology effectively addresses all CQs while incorporating external standards to enhance reusability.

6 Entity Definition

In this section, we address the final phase of the iTelos methodology: Entity Definition. This phase aims to merge the knowledge resources collected in the previous steps, specifically the Teleontology produced and the dataset gathered and cleaned during the Information gathering phase, into a unified structure: the final Knowledge Graph

Throughout the previous phases, we tackled various levels of heterogeneity:

- At the source level, selecting trusted data sources.
- At the format level, standardizing the resources collected into well-known, open formats, such as .csv.
- At the structure level, defining a Teleontology suitable for our purpose.

However, one type of heterogeneity remains unaddressed: Data value heterogeneity. This form of heterogeneity is crucial for identifying entities in the real world and for distinguishing one entity from another.

In this phase we focus on addressing this type of heterogeneity by defining activities to manage its various aspects: Entity Matching, Entity Identification, Entity Mapping.

6.1 Entity Matching

The first activity focuses on addressing the multiple ways of representing a real-world entity through different properties across various datasets. Specifically, this involves:

- **Schema layer:** Identifying the appropriate set of properties across the gathered datasets.
- **Data layer:** Resolving inconsistencies when multiple entity representations share the same properties but have differing values.

Following the iTelos middle-out approach, most misalignment between ETypes and entities have already been addressed by modelling the teleontology with the datasets in mind, and aligning the datasets with the Teleontology's modelling choices.

However, a significant issue persist: When working with dataset that describes the same category of real-world entities, such as bicycle racks and bicycle sharing parking lots, how can we ensure that no intersection exists between these datasets, i.e., that the same entity is not represented in both? We identified this issue in several of our datasets and addressed it as follows:

- **Urban and Extra-urban transportation datasets:** Our objective was to merge the urban and extra-urban transportation datasets into a single unified resource while ensuring that no duplicate entities were present. After thorough analysis, we confirmed that there were no duplicated stops between the two sources. However, an issue was identified regarding stop IDs: in both datasets, some stops shared the same ID despite being distinct entities, as indicated by their significantly different geographical positions. To resolve this, we introduced a prefix system, adding ”u_” and ”e_” to the stop IDs in the urban and extra-urban datasets, respectively. This modification was applied to the stops.csv and stop_times.csv files to prevent potential conflicts.
- **Bike_sharing.csv and Centro_in_bici.csv:** As both datasets referred to bike-sharing systems, it was crucial to verify whether bike_sharing.csv already contained data from centro_in_bici.csv. To identify duplicate entities, we compared their latitude and longitude values, applying a predefined tolerance to account for minor variations in positioning. Non-duplicated entities were then merged into a single consolidated file.
- **Rastrelliere.csv and Parcheggio_protetto_bike.csv:** A similar approach was applied to these datasets, which both contained information on bike parking facilities. By comparing

entity locations, we ensured that only unique entities were retained, and the final dataset was created by merging the non-redundant entries.

The script performing those operations is available in our GitHub Repository.

6.2 Entity Identification

The second step addresses the formal identification of entities within the dataset. Typically, each entity is identified by its properties. However, in high-quality dataset, one property is often explicitly designated as the identifier, uniquely referencing the entity it describes. When such a property is not available, entities can instead be identified by an identifying set: the union of two or more property values that collectively provide a unique reference within the set of considered entities.

In our case, many of the datasets used already included identifying properties. In instances where this was not the case, we created custom identifiers during the Dataset Cleaning phase to ensure consistent and unique identification. With all entities properly identified, we can now proceed with the next activity.

6.3 Entity Mapping

The last activity focuses on concretely mapping the information defined in the Teleontology to the corresponding values contained in the various datasets. This phase ensures that the entities and relationships modelled in the Teleontology are accurately linked to the actual data, providing a unified and semantically enriched representation.

To achieve this, we use Karmalinker, ”a tool for data linking which is enhanced with language understanding capabilities, which allow us to align data to reference ontology”. Through Karmalinker, we automate the process of associating the properties and values in the datasets with their corresponding ETypes, object properties, and data properties in the Teleontology.

Below is an image showing the result of one mapping operation performed on the Transit_line EType, which has been mapped with the transportation_lines.csv file:

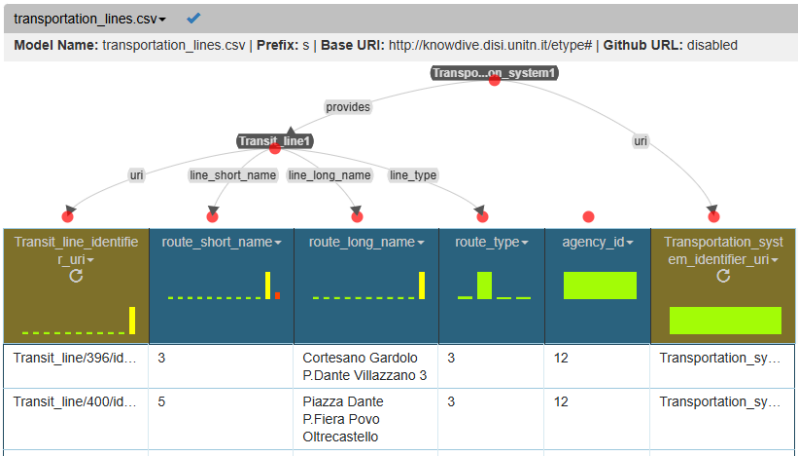


Figure 5: Entity Mapping using Karmalinker

As shown in the image, all properties from the transportation_lines.csv dataset have been successfully mapped to their corresponding data properties within the EType. Additionally, the identifiers have been correctly linked. As previously mentioned, each entity within the dataset is uniquely identified by an ID, which must be mapped as a URI, a unique sequence of characters that identifies a logical or physical resource used in web technologies. Instead of using the raw identifier, we applied the following Python transformation:

```
return "Transit_line/" + getValue("route_id") + "/identifier"
```

This transformation ensures that each entity receives a unique URI-based identifier. For example, the transit line with ID 396 is mapped to Transit_line/396/identifier.

The output of this phase consist of:

- **Mapping Model:** One or more RDF-Turtle files defining all Python transformations and mapping operation. These files are available in our GitHub Repository.
- **Final Knowledge Graph:** One or more RDF-Turtle files representing the KG generated through the iTelos methodology. These are accessible trough our GitHub Repository.

This step ensures that the resulting Knowledge Graph is both accurate and semantically consistent, making it ready for queries and further reasoning.

7 Evaluation

In this section, we evaluate the result obtained through the iTelos process, verifying the quality of our final Knowledge Graph. The evaluation follows the criteria defined by the iTelos methodology:

- **Purpose satisfaction:** Measures how well the final Knowledge Graph satisfies the Competency questions defined in the initial phase.
- **Reusability:** Defines the extent to which the final Knowledge Graph can be reused in future iTelos processes

7.1 Knowledge Layer Evaluation

We begin by evaluating the knowledge layer, using Coverage as the primary metric. Coverage is defined as the ratio between the intersection of α and β and the entire α set. This metric from 0 to 1, with higher values indicating better alignment of the KG with the domain, and lower values suggesting either inadequacies in the schema or that the target domain is mostly unexplored.

To ensure a comprehensive evaluation, we apply coverage-based metrics to two comparisons:

- **Teleontology vs Competency Questions (Primary Objective):** We evaluate how well the Teleontology covers the entities and properties extracted from the initial Competency questions.
  - **EType Coverage:** Given a set of Competency Questions CQ, to compute the EType coverage Cov_E, we consider the following: CQ_E, the number of ETypes extracted from CQ; T_E, the number of ETypes defined in our Teleontology T. The coverage is calculated as follow:
  $$Cov_E(CQ_E) = \frac{|CQ_E \cap T_E|}{CQ_E} = \frac{15}{15} = 1$$
  - **Property Coverage:** Given a set of Competency Questions CQ, to compute the Property coverage Cov_P, we consider the following: CQ_P, the number of properties extracted from CQ; T_P, the number of properties defined in our Teleontology T. The coverage is calculated as follow:
  $$Cov_P(CQ_P) = \frac{|CQ_P \cap T_P|}{CQ_P} = \frac{19}{20} = 0.95$$
  The missing property in the intersection is the one about parking lots' opening hours, that wasn't included in the following phases due to the scarcity of information regarding it founded in the datasets.
- **Teleontology vs Reference Ontologies (Secondary Objective):** We will evaluate how much our Teleontology covers the ETypes and properties extracted from the reference ontologies taken into account during the previous phases:

– **EType Coverage:** Given a set of Reference Ontologies RO, to compute the EType coverage Cov_E, we consider the following: RO_E, the number of ETypes extracted from the Reference Ontologies RO; T_E, the number of ETypes defined in our Teleontology T. The coverage is calculated as follow:

$$Cov_E(RO_E) = \frac{|RO_E \cap T_E|}{RO_E} = \frac{7}{28} = 0.25$$

In both evaluations, we used the SCHEMA.org ontology as a reference. Given its extensive coverage across multiple domains, we selectively extracted only the ETypes and properties relevant to our specific domain. This refined subset of the original ontology is available in our GitHub Repository

The results obtained indicate a relatively low coverage. However, this is primarily due to the nature of the refined ontology, which still includes many ETypes that are either too generic or too specific to be directly useful for our purposes. In fact, while SCHEMA.org provides a broad and well-structured knowledge base, not all its concepts align perfectly with the requirements of our Teleontology, as our focus is on a more domain-specific representation designed to answering the defined Competency Questions.

7.2 Data Layer Evaluation

Having analysed the final Knowledge Graph at the Knowledge Layer, in this section we will evaluate its Data Layer, aiming to measure how connected is the KG. In order to compute the connectivity of our KG, we leverage a Connectivity Matrix, a matrix whose (X,Y) cells assume these values:

- If $X = Y$, the cell (X,Y) is equal to the number of non-null data properties values for all the entities mapped on the EType X. We will call these values "# values".
- Otherwise, if $X \neq Y$, the cell (X,Y) is equal to the number of non-null object properties values for all the object properties having the EType X as domain and Y as range. We will call these values "* values".

Below is an image partially showing our Connectivity Matrix:

	City	Place	Bicycle_parking	Bike_sharing	Car_sharing	Parking_facility	Scooter_rental	Stop	Taxi_stand	Stop_event
City	8	0	451	47	8	572	39	0	9	0
Place	0	10188	0	0	0	0	0	0	0	0
Bicycle_parking	0	0	2706	0	0	0	0	0	0	0
Bike_sharing	0	0	0	282	0	0	0	0	0	0
Car_sharing	0	0	0	0	48	0	0	0	0	0
Parking_facility	0	0	0	0	0	4576	0	0	0	0
Scooter_rental	0	0	0	0	0	0	195	0	0	0
Stop	0	0	0	0	0	0	0	19830	0	162992
Taxi_stand	0	0	0	0	0	0	0	0	45	0
Stop_event	0	0	0	0	0	0	0	0	0	651904
Transit_line	0	0	0	0	0	0	0	0	0	0
Transportation_schedule	0	0	0	0	0	0	0	0	0	0
ransportation_special_schedul	0	0	0	0	0	0	0	0	0	0
Transportation_system	0	0	0	0	0	0	0	0	0	0
Transportation_trip	0	0	0	0	0	0	0	0	0	162992

Figure 6: A portion of our Connectivity Matrix

The whole Connectivity Matrix is available in our Github Repository

Specifically, we will use the Connectivity Matrix to analyse connectivity across two dimensions:

- **Entity connectivity:** This metric evaluates how much the different entities in the Knowledge Graph are interconnected. For a specific EType X, its Entity Connectivity $EC(X)$ is calculated by summing the "* values" in X's row of the connectivity matrix, and dividing this by $OP(X)$, the number of object properties defined for X:

$$EC(X) = \frac{\sum_{Y=1}^{N}(X,Y)}{OP(X)}$$

  Having computed $EC(X)$ for every EType X, we can compute $EC(KG)$, the Entity Connectivity for the overall Knowledge Graph:

$$EC(KG) = \sum_{X=1}^{N} EC(X) = 339042.14$$

- **Property connectivity:** This metric measures how well each entity in the Knowledge Graph is connected with its property values. For a specific EType X, its Property Connectivity $PC(X)$ is calculated by dividing the "# values" in the diagonal cell (X,X) of the connectivity matrix by $DP(X)$, the number of data properties defined for X:

$$PC(X) = \frac{(X,X)}{DP(X)}$$

  Having computed $PC(X)$ for every EType X, we can compute $PC(KG)$, the Property Con-

nectivity for the overall Knowledge Graph:

$$PC(KG) = \sum_{X=1}^{N} PC(X) = 185278$$

To compute $EC(X)$ and $PC(X)$ for each EType X, we developed a Python script, available in our GitHub Repository.

7.3 Knowledge Graph Exploitation

In the iTelos methodology, Knowledge Graph Exploitation focuses on leveraging the constructed Knowledge Graph to extract meaningful insights. This exploitation can take various forms, including: Interrogation and searches through query languages; Integration with Machine Learning models as structured input; Embedding within software applications, and more. For our project, we concentrated on graph interrogation, utilizing GraphDB, an enterprise-ready Semantic Graph Database developed by Ontotext. GraphDB is W3C-compliant and allows us to create a repository where we can upload all the .ttl files representing our Knowledge Graph. Once these files are loaded, the system reconstructs the graph and enables us to make queries through SPARQL, a powerful protocol and declarative query language for RDF data.

Below is an image showing a representation, obtained from GraphDB, of our Knowledge Graph:

Figure 7: Hierarchical View of our Knowledge Graph

Throughout the iTelos process, our primary goal was to develop a Knowledge Graph that effectively answers the Competency Questions (CQs) defined in the Purpose Definition phase (2.5). We successfully translated all CQs into functional SPARQL queries, with a single exception:

- **CQ 7.2:** As mentioned earlier in the Knowledge Layer Evaluation section, although we initially assumed that information on parking facility opening hours was available, we later found that this data was missing for most entities. Consequently, we decided to exclude

this query.

Below is an example of a SPARQL query, while All queries are available in our GitHub Repository.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX etype: <http://knowdive.disi.unitn.it/etype#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

# 4.3 (Marco): On line 7, how many stops are there from “"Gorizia Adamello” to “Gocciadoro Arcate”?

select distinct ?line_name ?departure_name ?arrival_name ((xsd:integer(?arrival_sequence) - xsd:integer(?departure_sequence)) as ?n_stops) where {

    # Define the departure and arrival event for every trip and filter by direction
    ?trip etype:Contains ?departure_event, ?arrival_event .
    ?trip etype:trip_direction ?trip_direction .
    FILTER(STR(?trip_direction) = "0")

    # Select the departure stop by name
    ?departure_stop etype:name ?departure_name ;
    FILTER(STR(?departure_name) = "Gorizia Adamello")

    # Select the arrival stop by name
    ?arrival_stop etype:name ?arrival_name .
    FILTER(STR(?arrival_name) = "Gocciadoro \"Arcate\"")

    # Connect the previously defined stops with the corresponding stop events
    ?departure_stop etype:is_location_for ?departure_event .
    ?departure_event etype:sequence_number ?departure_sequence ;
                     etype:arrival_time ?departure_time .

    ?arrival_stop etype:is_location_for ?arrival_event .
    ?arrival_event etype:sequence_number ?arrival_sequence ;
                   etype:arrival_time ?arrival_time .

    # Connect the lines with trips
    ?line etype:composed_of ?trip .
    ?line etype:line_short_name ?line_name .

    # We want only the arrival stops that appen after the departure event
    FILTER(str(?arrival_sequence) > str(?departure_sequence))

}
```

Figure 8: Complete query for the CQ 4.3

8 Metadata Definition

After verifying the quality of our final Knowledge Graph, this section focuses on its distribution. This is essential within the iTelos methodology, which defines each project as a ”cooperative composite project”. In such a context, various types of actors can be identified:

- **Producers:** Actors involved in the generation of resources through an iTelos project, who aim to share their results for future reuse.
- **Consumers:** Actors interested in reusing resources previously generated by other projects.
- **Intermediaries:** Actors focused on generating resources that reduce heterogeneity (in one or more layers) between producers and consumers.

To effectively share our results, we will utilize Catalogs, web-based access point for data repositories. These catalogs allow us to organize resources based on their layers (Language, Knowledge and Data). A key characteristic of these catalogs is that they do not store entire datasets but rather their metadata, structured information that is essential for ensuring the quality and reusability, as well as for facilitating the description, organization and discovery of resources.

For this project, metadata will be organized in three distinct layers:

- **People Metadata:** Attributes describing the actors involved in this iTelos project.
- **Project Metadata:** Attributes describing the iTelos project itself.
- **Dataset Metadata:** Attributes describing the datasets used during the iTelos project.

Our metadata are available in the GitHub Repository

Below is an image partially showing our Dataset Metadata:

DatURL	DatKeyword	DatPublisher	DatCreator	DatOwner
https://dati.trentino.it/dataset/trasporti-pubblici-del-trentino-formato-	Transportation System, Trento	OPENdata Trentino	Servizio Trasporti Pubblici	Provincia Autonoma di Trento
https://dati.trentino.it/dataset/taxi-open-data	Taxi, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://dati.trentino.it/dataset/car-sharing-open-data	Car, Sharing Service, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://dati.trentino.it/dataset/bike-sharing-open-data	Bike, Sharing Service, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://dati.trentino.it/dataset/c-entro-in-bici-open-data	Bike, Sharing Service, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://dati.trentino.it/dataset/rastrelliere-per-biciclette-open-data	Bike Rack, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://dati.trentino.it/dataset/parcheggio-protetto-per-biciclette-ope	Bike Rack, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://dati.trentino.it/dataset/punti-sosta-monopattini-condivisi-a-ta	Scooter, Sharing Service, Trento	OPENdata Trentino	Comune di Trento	Comune di Trento
https://github.com/NikoMrs/KGE-Trentino-Territory-Transportation/t	Parking, Trento	Open Street Map	Mores Nicola, Roccon Marco	Open Street Map

Figure 9: Dataset Metadata

Alongside the common metadata, we introduce the following additional attribute:

- DatLatestModificationTimestamp: This attribute encodes the timestamp of the most recent modification applied to the dataset.

9 Open Issues

Throughout this project, we followed the iTelos methodology, starting from an informal purpose, gradually structuring and formalizing it, retrieving and standardizing data from various sources, and defining key resources at multiple levels (knowledge, language and data) to build a Knowledge Graph. This structured approach allowed us to address many of the initial challenges; however, certain issues remain open.

The main unresolved issue concerns the inability to fully answer the Competency Questions related to parking lot capacity. Despite extensive research, we did not find datasets providing comprehensive information on the opening hours of parking facilities. In some cases, data was

available for specific parking lot, but it was insufficient for a generalizable solution. This gap prevents the Knowledge Graph from offering precise insights on parking availability, representing a key limitation of the current implementation.

Beyond this, there are several areas that could be improved in future developments. As previously mentioned, one of these is the integration of German into the Language Resource file. We decided not to include it due to our limited proficiency in the language, which would have made it difficult to ensure the level of precision required for this purpose. Given that this file must be highly detailed to avoid misinterpretations, a future improvement could involve collaboration with native speakers or language experts.

Another valuable enhancement would be the incorporation of real-world transport data, particularly actual departure and arrival times considering delays and anticipations. Rather than providing real-time information, this data could be used to generate historical averages, allowing for predictions on expected arrival times at specific stops based on past trends. Similarly, occupancy estimation data would enable the system to provide probabilistic insights on seat availability at different times of the day for specific routes. However, in both cases, we were unable to find datasets providing this information at a sufficiently detailed level for integration. These additions would significantly improve the usability of the Knowledge Graph, making it even more useful for end users.

Despite these limitations, the project successfully achieved its primary goals, creating a structured Knowledge Graph capable of supporting various urban mobility queries. Future efforts could focus on closing these gaps, enhancing both data completeness and system functionality.