Dipartimento di Ingegneria e Scienza dell'Informazione

– KnowDive Group –

KGE 2023 - Project Report Trentino Transportation & Education Facilities

Document Data:	Reference Persons:
January 21, 2024	Thomas Pasquali, Riccardo Peiretti

Index:

1 Introduction **1**

2 Purpose and Domain of Interest (DoI) **1**

2.1 Purpose . . . 1

2.2 Domain of Interest . . . 2

3 Project Development **2**

3.1 Data Production . . . 2

3.2 Data Composition . . . 3

4 Purpose Formalization **4**

4.1 Scenarios . . . 4

4.2 Personas . . . 4

4.3 Competency Questions (CQs) . . . 4

4.4 Concepts Identification . . . 5

4.5 ER model definition . . . 7

4.5.1 Design process . . . 7

4.5.2 Discussion on the ER model . . . 7

4.5.3 ER model . . . 8

5 Information Gathering **9**

5.1 Consumer activities . . . 9

5.1.1 Knowledge layer . . . 9

5.1.2 Data layer . . . 9

5.2 Producer activities . . . 10

5.2.1 Knowledge layer . . . 10

5.2.2 Data layer . . . 11

5.3 Schema generation . . . 13

5.4 Formal resources generation . . . 14

5.5 Summary of the final datasets . . . 15

6 Language Definition **16**

6.1 Dataset filtering . . . 20

7 Knowledge Definition **21**

7.1 Ontologies . . . 21

7.1.1 Datascientia LiveKnowledge . . . 21

7.1.2 Schema.org . . . 21

7.1.3 Custom . . . 22

7.1.4 Final result . . . 22

7.2 Teleology . . . 22

7.3 Teleontology . . . 24

8 Data Definition **26**

8.1 Entity Matching . . . 26

8.2 Entity Identification . . . 27

8.3 Entity Mapping . . . 27

8.3.1 Producer activities . . . 28

8.3.2 Consumer activities . . . 28

9 Evaluation **29**

9.1 Knowledge Layer Evaluations . . . 29

9.1.1 Teleontology . . . 29

9.1.2 Teleontology vs CQs . . . 29

9.1.3 Teleontology vs Reference Ontologies . . . 30

9.2 Data Layer Evaluations . . . 31

9.2.1 Knowledge Graph information statistics . . . 32

9.3 Query execution . . . 32

10 Metadata Definition **48**

10.1 Language resources . . . 48

10.2 Knowledge resources . . . 48

10.3 Data resources . . . 49

11 Open Issues **51**

12 Conclusion **52**

Revision History:

Revision	Date	Author	Description of Changes
0.1	2023-10-10	Thomas Pasquali	Document created
0.2	2023-10-15	Thomas/Riccardo	Purpose Formalization
0.2.1	2023-10-23	Thomas/Riccardo	Revision of Purpose Formalization
0.3	2023-10-25	Thomas/Riccardo	Information Gathering
0.4	2023-11-08	Thomas/Riccardo	Information Gathering finalization + Language definition
0.5	2023-11-21	Thomas/Riccardo	Knowledge definition
0.6	2023-12-18	Thomas/Riccardo	Data definition + Evaluation + Matadata (draft)
1.0	2024-01-20	Thomas/Riccardo	Evaluation queries + Open Issues + Conclusion

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 reusabiltiy of the resources handled and produced during the process. A clear description of the resources as well as of the process (and sub processes) developed, provides a clear understanding of the project, thus serving such an information to external readers for the future exploitations of the project's outcomes.

The current document aims to provide a detailed report about *"Trentino Transportation and Education Facilities"* KGE project developed following the iTelos methodology. The report is structured, to describe:

- Section 2: Definition of the project's purpose and its domain of interest.
- Section 3: High level description of the project development, based on the two main sub process considered by iTelos, producer and consumer, respectively.
- Sections 4, 5, 6, 7 and 8: The description of the iTelos process phases and their activities, divided by knowledge and data layer activities, as well as considered from the point of view of the producer first, and the consumer later.
- Section 9: The description of the evaluation criteria and metrics applied to the project final outcome.
- Section 10: 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 11: Conclusions and open issues summary.

Here is a link to the GitHub repository that contains all the material used during the development of this project: https://github.com/ThomasPasquali/TransportationAndEducationFacilities_KGE2023.

2 Purpose and Domain of Interest (DoI)

2.1 Purpose

The goal of this project is to build a Knowledge Graph (KG) which satisfies the many different needs of students, professors and educational staff to reach their middle/high school and university by public transportation within the region of Trentino or from the main Italian cities, depending on the specific person engagements, domicile and residence.

This is obtained by integrating static transportation data from different sources with educational facilities and synthetic user data, in order to produce a KG and provide reusable tools to parse raw data.

This KG purpose is to provide data to a web/smartphone application (e.g. UniTrentoApp) for, given the user weekly educational schedule, the current date and time: easily and conveniently gathering information about public transportation to organize the most convenient trips. This purpose includes, both out-of-door and Trentino's students and teachers that rarely have to reach Trentino.

2.2 Domain of Interest

The DoI of this project consists of two boundaries: space and time. This KGE project considers the public transportation of the region of Trentino (Italy) and main Italian cities (e.g. Milano, Torino, Bologna) together with Trentino education facilities over a period of time of approximately nine months between September 2023 and June 2024.
Notes:

- The temporal domain boundary is given by when Trentino Trasporti changes from winter to summer timetables;
- The final knowledge graph considers only the main itineraries that should never change;
- Some datasets may only cover a portion of the time between the boundaries.

3 Project Development

This section describes, at top level, how the project's purpose is satisfied.

3.1 Data Production

The goal of the data producer is to provide to the consumer the data needed to fulfill its purpose, in particular, data is preferred to be in a high-quality format. If, for each specific dataset, only low-quality data (e.g. complex PDF) is available, the producer is in charge of providing a high-quality version of data.
Data about the transportation is gathered from sources that offer datasets similar to those provided by the following sources:

1. Trentino Transporti extra-urban
2. Trentino Transporti trains
3. Trentino Transporti urban
4. Trentino Transporti Open Data
5. Flixbus
6. Trenitalia
7. E656

More precise details on the data sources are provided in Section 5.1.2.
From this kind of sources, it is possible to find many information like stops, timetables, routes etc.

about buses and trains. This data usually covers information for one year. Many of them follow the guidelines of the General Transit Feed Specification (GTFS).
Data about the education facilities are gathered from the following sources:

1. OPENdata Trentino
2. Tuttitalia Trentino - middle and high school
3. UniTrento Digital University

From these sources, it is possible to find datasets contain the number of facilities of middle-schools, high-schools, faculties, education-staff, and administration of the University of Trento, Povo, and Rovereto

3.2 Data Composition

Once the consumer has retrieved the data, its goal is to transform it into a well-defined standard format so to be able to dynamically provide it to the final user and possibly reuse in similar contexts. Considering the above data sources, regarding transportation the consumer has to:

1. Download high-quality i.e. GTFS data from available sources (Trentino Trasporti, Flixbus and Trenitalia);
2. Integrate (for example FTM trains timetables) data, which is only available as PDF, or other non high-quality formats, by web scraping and "manual" parsing;

About educational facilities, the consumer has to:

1. Get as much useful data from *OPENdata Trentino* (middle and high-schools information);
2. Get as much useful data from *Tuttitalia Trentino* (middle and high-schools information);
3. Use *UniTrento Digital University* to get Universities information.

Data about users is manually crafted to demonstrate how the purpose is fulfilled. Below an intuitive schema of the workflow of this project.

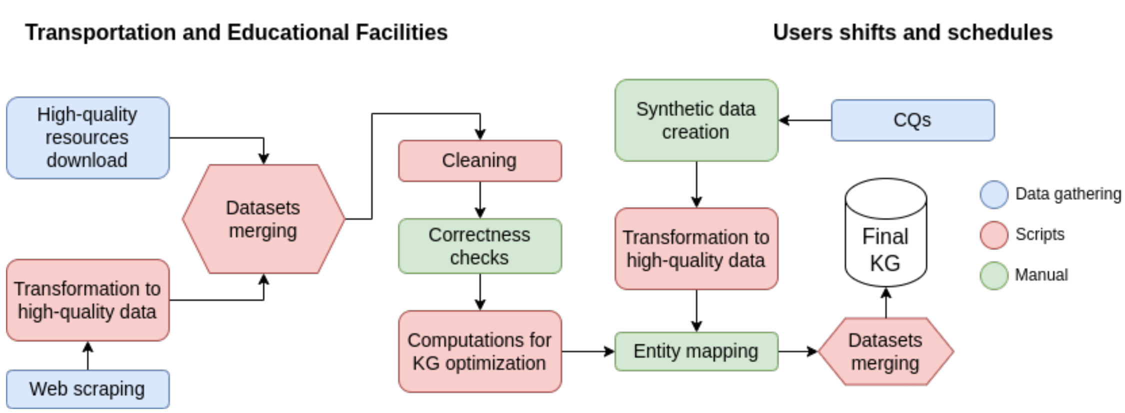


Figure 1: Process flow intuition

4 Purpose Formalization

The following scenarios and personas try to describe all the possible combinations of circumstances, some of them depend on the Persona while the others describe when something happens or which kind of public transportation the trip refers to.

4.1 Scenarios

1. Week day (Monday to Saturday):
   a. Morning (7-12AM);
   b. Afternoon/evening (2-7PM);
   c. Night (after 7PM).
2. Holiday:
   a. Morning (7-12AM);
   b. Holiday afternoon/evening/night (after 1PM);
3. Train / Bus trip;
4. Urban trip;
5. Extra-urban trip.

4.2 Personas

ID	Name	Address	Home Location	Studies/Works	Special needs
1	Mario	Mollaro	Mollaro	MS (Mezzolombardo)	None
2	Jessica	Povo	Povo	HS (Trento)	None
3	Luca	Trento (Clarina)	Trento (Clarina)	HS (Trento)	Wheelchair
4	Gian	Trento	Trento	U (Rovereto)	None
5	Gaia	Trento (SanBa)	Bologna	U (Povo)	None
6	Carla	Trento (Lungadige)	Torino	U (Povo)	None
7	Fausto	Milano	Milano	U (Trento)	None

Table 1: Personas

Legend: MS = Middle school, HS = High school, U = University

4.3 Competency Questions (CQs)

1. Mario lives in Mollaro and from Monday to Friday needs to go to the middle-school in Mezzolombardo from 8AM to 1PM, except on Thursday he needs to stay in school until 4PM. He only has a train subscription and he must be independent because his parents work all day;

2. Jessica lives in Povo and from Monday to Saturday she has to go to the high-school in the center of Trento. Her hours are from 8AM to 1PM;
3. Luca lives in Trento and he goes to high-school from Monday to Saturday in the center of Trento from 8AM to 1PM, but he is in a wheelchair, and he must be independent using suitable and accessible buses;
4. Gian is a Rovereto university student who lives in Trento and he needs to reach the university from Monday to Wednesday from 10:30AM, and he needs to arrive at home before 4:30PM;
5. Gaia lives at Sanbapolis. From Monday to Friday she needs to reach the University in Povo from 7:15AM to 1PM, but on Wednesday she goes to uni library in University of Sociology in Trento from 9:30AM to 10:30AM. However, on 2024-03-07 she will have a special lecture in University of Sociology in Trento at 12:30AM;
6. Carla is an out-door-students and she lives in Lungadige in Trento. She return at home in Turin from Trento every Friday to stay at home in the weekend;
7. Fausto is a Professor at University of Milan, and he needs to book a ticket train from Milano Centrale to reach the University of Sociology on the 13st, Nov 2023 (Monday) before 10:30AM to have an important seminary.

The above situations aim at catching the diversity expressed by the project purpose, by defining clearly the different aspects, about the involved contexts and actors.
These sentences are used in Section 9.3 to produce a *schedule of shifts* for each person. By using this schedule and gathered data, it is possible to answer questions like:

- "On day d at time t, which bus or train shall the person p take in order to reach its school or university in time?"
- "On day d at time t, which bus or train shall the person p take once finished his/hers duties to get back home?"
- "On day d at time t, which bus or train shall the person p take in order get to his/her real home?" (out-of-doors students)

The answers to these questions is a list of possible public transportation solutions, for instance, "from the bus stop s_0, at time t_0, take the bus with number (route) r, and get off the bus at stop s_1 OR from the train station s_1 take the train with destination d at time t_1" and get off it at stop s_1.

4.4 Concepts Identification

The following table describes the terms representing the entities, and their properties, to be consider in the project, classified using the popularity categories.

In table 2, *Focus* defines how much an entity is "important" w.r.t. to the main purpose while *Popularity*, defines how much an entity is reused in already existing data (considering the input information sources).
Both Focus and Popularity, for each entity, can assume one of the following three values:

- *Common*: (Focus) general entities for the purpose considered. (Popularity) The entity is largely available in existing resources.
- *Core*: (Focus) specific entities for the purpose considered. (Popularity) The entity is available in existing resources, but not so common.
- *Contextual*: (Focus) very specific entities for the purpose considered. (Popularity) The entity is not available in existing resources.

Scenarios	Personas	CQs	Entities	Properties	Focus	Popularity
1,2,3,5	1-7	1-7	EndUser	id, name, specialNeeds, domiciled, reside, work	Contextual	Core
1-5	1-7	1-7	Provider	id, name	Contextual	Contextual
1-5	1-7	1-7	Route	id, provider, type, shortName, longName	Core	Core
1-5	1-7	1-7	Trip	id, headsign, direction, accessibility, characterized, avaibilitySchEx, avaiabilitySchedule	Core	Core
1-5	1-7	1-7	Stop	id, name, type, localized	Core	Core
1-5	1-7	1-7	TripStop	id, arrivalTime, departureTime, stopSequence, of, at	Core	Core
1-5	1-7	1-7	WeeklySchedule	id, monday, tuesday, wednesday, thursday, friday, saturday, sunday, startDate, endDate	Core	Core
1-5	1-7	1-7	ScheduleException	id, date, type	Core	Core
1-5	1-7	1-7	Position	id, address, latitude, longitude	Contextual	Core
1-5	1-7	1-7	EducationalFacility	id, legalName, address, type, nearestStops	Core	Core
1-4	1-7	1-7	Shift	id, from, to, arriveBefore, leaveAfter, occurenceSchEx, occurenceSchedule, involvement	Contextual	Contextual
1-4	1-7	1-7	Domicile	id, domicile, nearestStopsDom	Contextual	Contextual
1-4	1-7	1-7	Residence	id, residence, nearestStopsRes	Contextual	Contextual

Table 2: Purpose Formalization sheet

The table shows that each CQs uses every entity, with some exceptions e.g. Mario that can only travel by bus, thus highlighting that questions contain similar answers. The difference between them consists in different combinations of: positions, date, time, and type of transportation.

4.5 ER model definition

4.5.1 Design process

Considering the situations and personas involved in the competency questions, the following steps have been followed:

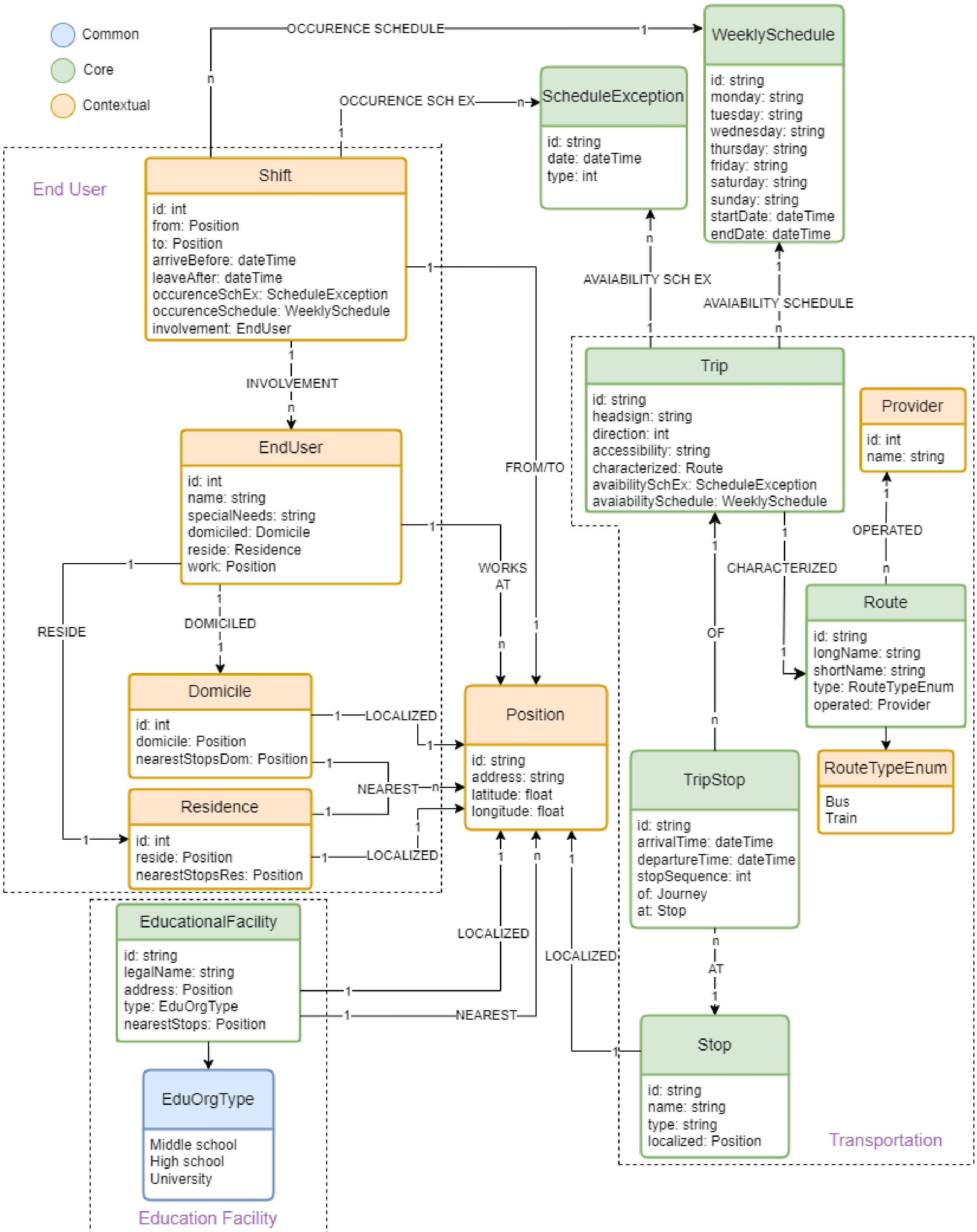
1. Entities identification:
   - *Trip (Route, Stop, TripStop)*: express how a person can move from one place to another;
   - *EndUser (EndUser, Shift, Domicile, Residence)*: describe who the user is, where the person lives, and the shifts she/he needs;
   - *EducationalFacility*: describes a school or university;
   - *WeeklySchedule* and *ScheduleException*: describe when trips or shifts are scheduled.
2. Attributes definition: for each entity, associated attributes are defined. For example, a *Route* entity has attributes such as *id, longName, shortName, type (bus/train), operated (provider)*. This allows the storage of information useful to fulfill CQs and the purpose;
3. Relationships identification: determine how entities relate to each other. For example, relations such as *OF, CHARACTERIZED, AT, OPERATED, AVAIABILITY SCHEDULE EXCEPTION*, and *AVAIABILITY SCHEDULE* have been identified for the contest of transportation. Relations are fundamental to "link" entities, for example a Trip is characterized by a Route. For more details, in section 4.5.3 there is the entire ER model;
4. Relationship cardinality: each relation should specify how many instances of the two entities are involved. For example, the *INVOLVEMENT* relationship connects *Shift* to *EndUser* with a cardinality of "1 to n", meaning that one shift can involvement many users;
5. Create the ER Diagram: based on all previous steps the final ER diagram is created with draw.io. This diagram is a visual representation of entities, attributes, and relationships.

4.5.2 Discussion on the ER model

The ER model - in section 4.5.3 - gives an idea of the three main components of the knowledge graph:

- *EndUser*: entities *EndUser*, *Shift*, *Domicile*, and *Residence* are used to store person data together with the shifts it needs and informations about where a person lives;
- *Education*: the entity *EducationalFacility* main purpose is to gather geographical information about schools and universities;
- *Transportation*: entities *Trip, Provider, Route, Stop*, and *TripStop* are used to identify which places can be reached with public tranportation;
- *Common*: entities *Position, WeeklySchedule*, and *ScheduleException* are used by multiple of the just presented "domains".

4.5.3 ER model



Common
Core
Contextual
OCCURENCE SCHEDULE
OCCURENCE SCH EX
WeeklySchedule
id: string
monday: string
tuesday: string
wednesday: string
thursday: string
friday: string
saturday: string
sunday: string
startDate: dateTime
endDate: dateTime
ScheduleException
id: string
date: dateTime
type: int
End User
Shift
id: int
from: Position
to: Position
arriveBefore: dateTime
leaveAfter: dateTime
occurenceSchEx: ScheduleException
occurenceSchedule: WeeklySchedule
involvement: EndUser
AVAIABILITY SCH EX
AVAIABILITY SCHEDULE
INVOLVEMENT
FROM/TO
Trip
id: string
headsign: string
direction: int
accessibility: string
characterized: Route
avaibilitySchEx: ScheduleException
avaiabilitySchedule: WeeklySchedule
Provider
id: int
name: string
EndUser
id: int
name: string
specialNeeds: string
domiciled: Domicile
reside: Residence
work: Position
OPERATED
WORKS AT
CHARACTERIZED
RESIDE
DOMICILED
OF
Route
id: string
longName: string
shortName: string
type: RouteTypeEnum
operated: Provider
Domicile
id: int
domicile: Position
nearestStopsDom: Position
LOCALIZED
Position
id: string
address: string
latitude: float
longitude: float
NEAREST
Residence
id: int
reside: Position
nearestStopsRes: Position
TripStop
id: string
arrivalTime: dateTime
departureTime: dateTime
stopSequence: int
of: Journey
at: Stop
RouteTypeEnum
Bus
Train
EducationalFacility
id: string
legalName: string
address: Position
type: EduOrgType
nearestStops: Position
AT
Stop
id: string
name: string
type: string
localized: Position
EduOrgType
Middle school
High school
University
Transportation
Education Facility

5 Information Gathering

This section aims at reporting the execution of the activities involved in the Information Gathering iTelos phase. The report, starting from the current section, is organized along two main dimensions. The first one considers the parallel execution of the producer and consumer processes, while the second dimension takes into account the activities operating over data and knowledge layers.

5.1 Consumer activities

These activities are directed towards collecting resources that are ”quality and formal” and have deemed suitable for the project.

5.1.1 Knowledge layer

In the domains of education and transportation, the prior year's projects had established a dedicated knowledge layer tailored to their specific purposes, aligned with available data. However, these pre-existing knowledge layers cannot be directly applied to the current project due to its broader or too strict scope. Consequently, it is treated as a producer activity.

With regard to transportation, an initial briefing is made to acquire data and knowledge oriented toward the APIs of FlixBus and Trenitalia. The primary objective is to obtain up-to-date and diverse information and subsequently adapt it to adhere to a standardized schema (GTFS). Unfortunately, this initial attempt has proven unsuccessful due to a lack of adequate documentation and excessively protracted API key acquisition procedures.
As a result of these challenges, the pursuit shifted towards the acquisition of static datasets. Trenitalia, FlixBus, and Trentino Trasporti have provided open data compliant with the General Transit Feed Specification (GTFS), which are elaborated upon in subsequent sections. Given that GTFS is a well-structured and widely accepted format, it has been chosen as the foundational inspiration for the transportation and some of the end user segment of the final Knowledge Graph schema.
The final schema has (slightly) modified GTFS to better fit the purpose on this project: for more detail refer to the related producer activity in section 5.2.1.

5.1.2 Data layer

A list of high-quality data resources considered for the project is presented:

- As just stated, the main sources of information about public transportation for both Trentino and main Italian cities are GTFS datasets. These are the websites that provide such files:
  - https://www.trentinotrasporti.it/open-data
  - https://www.dati.lombardia.it/Mobilit-e-trasporti/Orario-Ferroviario...
  - https://www.transit.land/feeds?search=trenitalia
  - https://transitfeeds.com/p/actv/630
  - https://www.transit.land/feeds/f-u-flixbus

  These files are then parsed, cleaned and merged as a producer activity (section 5.2.2).

- *Schools in Trentino* and *Universities of Trento, Povo, and Rovereto*: these data sources have been analyzed from the projects of previous years available on GitHub, but could not be reused due to their different purpose.

5.2 Producer activities

These activities aim at collecting "informal" resources from sources with an higher level of heterogeneity. The resources collected by the producer process are not compliant with the iTelos quality and reusability guidelines. Those are the resources that the producer transforms into quality resources at the end of the process.

5.2.1 Knowledge layer

Transportation (Core focus)

With regard to transportation, as discussed in the related consumer activity, GTFS datasets exhibited a considerable array of optional columns and standards for value formatting. Consequently, only a subset of relevant columns (for each file) has been selected. Some columns are added, dropped or renamed. The final structure of each GTFS file is summarized in the following table:

GTFS file	Columns
provider	id, name
route	id, operated, short_name, long_name, type
stop	id, name, type, localized
trip_stop	id, arrival_time, departure_time, stop_sequence, of, at
trip	id, headsign, direction, accessibility, characterized, accessibility, avaiability_schedule_exception, avaiability_schedule

Education (Core focus)

Regarding educational facilities, the schema has been obtained based on available data sources for middle, high-schools and universities as follows:

CSV file	Columns
educational_facility	id, legal_name, address, type, nearest_stops

End user (Contextual focus)

Regarding end users, the following schema has been created based on the purpose considered:

CSV file	Columns
end_user	id, name, special_needs, domiciled, reside, work
shift	id, from, to, arrive_before, leave_after, involvement, occurence_schedule_exception, occurence_schedule
domicile	id, domicile, nearest_stops_domicile
residence	id, reside, nearest_stops_residence

Common (Core focus)

Finally, three common entities have been created for *transportation, end user* and *education*:

CSV file	Columns
weekly_schedule	id, start_date, end_date monday, tuesday, wednesday, thursday, friday, saturday, sunday
schedule_exceptions	id, date, type
position	id, address, latitude, longitude

Note for the reader: the names of the files, entities, and data properties have been changed following the Language Definition phase outlined in section 6.

5.2.2 Data layer

This process aims to integrate what obtained from the consumer activities or produce new datasets. It has been organized in three different sections: *Transportation*, *Education*, and *End user*.

Transportation (Core focus)

Given the multiple and heterogeneous GTFS datasets, has risen the need to find common way to process and join them. For this purpose, the Python script 1_transport.py has been created. As explained in README.md, its goal is to take the files from the *"Raw data"* directory (structured by provider, region and urban/extraurban), clean and merge all datasets. The results of the process is written to the directory *"Parsed data"*. The resulting files are ready to be linked to the schema with *KarmaLinker*.
What is not provided by GTFS datasets has been integrated with data provided by the following sources:

Data source	Format	Description
E656	html	These website has been scrabed to obtain missing GTFS information (more on this in the next paragraphs).
Wikipedia	html	This website has been used to fetch missing data, for example FTM stops and exact locations (more on this in the next paragraphs).
PDF timetables	pdf	These timetables has been used to integrate missing data, for example the FTM trains *calendar*.
Trenitalia	-	This website has been used to check if E656 data are up-to-date.

Table 3: Data sources for transportation

This section of the process has been performed using *scrapy* together with some manual data parsing and correctness checks. The *"spiders"* are available in the GitHub repository. As a result, huge *json* files are obtained. These files are then converted into GTMS datasets by the 0_scraped_transport.py in the dedicated cells. This final dataset is written to the *"Raw data"* folder and later it is merged with the others as if they had been downloaded from the web.

Education (Core focus)

For the information regarding educational facilities, data are obtained using various resources and tools. The primary dataset is obtained from open sources, commonly known as open data, specifically from Tuttitalia - Trentino statistical portal, where information about middle and high-schools are available, and from Unitrento Digital University for the information about Trento, Povo, and Rovereto universities. The extraction is performed by web scraping as there is no suitable API for direct data access.

Data source	Format	Description
Tuttitalia - Trentino	csv	These datasets, obtained by activities of web scraping, contain information about middle and high schools in some municipality of the autonomous province of Trento
UniTrento Digital University	csv	This dataset, obtained by activities of web scraping, contains information about faculties of the Universities of Trento, Povo, and Rovereto

Table 4: Data sources for educational facilities

To work with the data and write code, Visual Studio Code is used as the source code editor, and Python is the programming language of choice. Scripts are available in the GitHub repository.

The *Requests* library is utilized to make HTTP requests for easy interaction with web services and downloading their content. *BeautifulSoup4* library is employed for data extraction from the HTML markup of the Tuttitalia and UniTrento Digital University portal. It enabled the navigation of the HTML document, searching for specific elements, and efficiently extracting the desired information.

Regarding the information about middle and high-schools, initially, a dataset is obtained for each type of school (middle and high-school) and for all municipalities of the autonomous province of Trento.

For all dataset, duplicate rows are removed, and a data cleaning operation is performed. Finally, all the dataset are merged using a function from 'Pandas', a Python library, to obtain a single file, called *educational_facilities.csv*, containing all the information for middle-schools, high-schools, and universities.

End user[1]

For this project, data about *End users* and related habits (*Shift*, and *Schedule*) are manually crafted from Personas and Competency Questions. This process is developed in two steps:

1. An "informal" version of the end users, recurring schedules and one-time events is provided;
2. A Python script, 3_users.py, parses and transforms data to schema compliant datasets.

The schema has been inspired from the GTFS calendar plus exceptions (the *"Common"* section of the schema). With such a schema it is possible to create queries that answer to the example questions presented in the last part of section 4.3.

[1] Recall that *End users* refer to *Personas*. This new word (concept) will better fit the point of view of the future applications that will use the KG to enrich the service provided.

Broadly speaking, the purpose indicates that this Knowledge Graph (KG) could be use in the integration of functionalities into existing applications. Consequently, data pertaining to individuals (*Person*) are produced directly from these applications.

5.3 Schema generation

Firstly, in order to achieve knowledge and data integration, a variety of tools are employed. To begin with, an ontology is created using the *Protege* development tool. This process is informed by the primary entities derived from the data structure and purpose. Below, some examples of what is done in *Protege*, i.e. the creation of the schema for each individual entity (saved in independent files. For more datails, GitHub link).

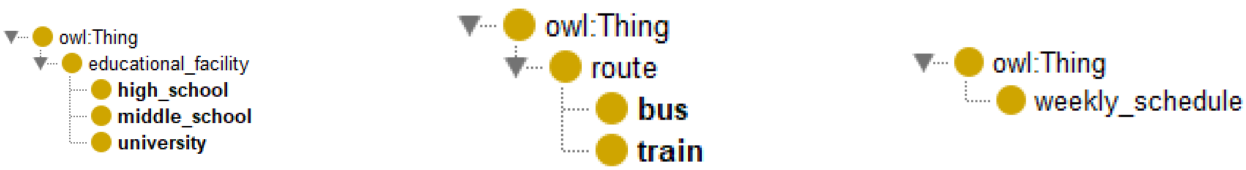


Figure 2: Examples of individual entity on Protege

Subsequently, data properties are initialized based on the data structure. All the entities properties are shown below. For each data property, the name is assigned with the following format: has_[EntityName]_[AttributeName]. This is done to enhance the overall project's comprehension and to quickly and intuitively understand which entity the attribute refers to. For instance, the property *"legal_name"* of an *Educational Facility* is has_educational_facility_legal_name.

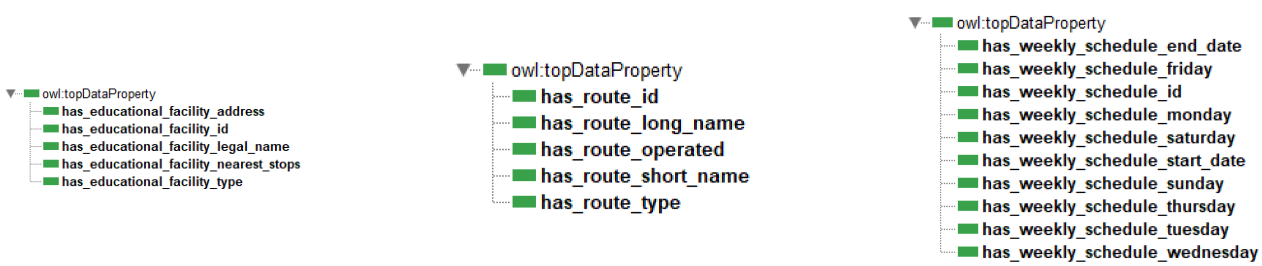


Figure 3: Examples of data properties for individual entity on Protege

Note: the above figures are intended to provide an intuition of the process and may be different in their final version.

In addition, the table below lists the data properties, the entities they refer to, and the range.

Property name	Domain(s)	Range(s)
Education		
has_legal_name	EducationFacility	xsd:string
has_type	EducationFacility	xsd:string
Transportation		
has_headsign	Trip	xsd:string
has_direction	Trip	xsd:int
has_accessibility	Trip	xsd:string
has_arrival_time	TripStop	xsd:dateTime
has_departure_time	TripStop	xsd:dateTime
has_stop_sequence	TripStop	xsd:int
has_name	Stop, Provider	xsd:string
has_type	Stop, Route	xsd:string
has_short_name	Route	xsd:string
has_long_name	Route	xsd:string
EndUser		
has_name	EndUser	xsd:string
has_special_needs	EndUser	xsd:string
has_arrive_before	Shift	xsd:dateTime
has_leave_after	Shift	xsd:dateTime
EndUser and Trasportation		
has_date	ScheduleException	xsd:dateTime
has_type	ScheduleException	xsd:int
has_monday	WeeklySchedule	xsd:string
has_tuesday	WeeklySchedule	xsd:string
has_wednesday	WeeklySchedule	xsd:string
has_thursday	WeeklySchedule	xsd:string
has_friday	WeeklySchedule	xsd:string
has_saturday	WeeklySchedule	xsd:string
has_sunday	WeeklySchedule	xsd:string
has_start_date	WeeklySchedule	xsd:dateTime
has_end_date	WeeklySchedule	xsd:dateTime
Education, EndUser, Transportation		
has_address	Position	xsd:string
has_latitude	Position	xsd:float
has_longitude	Position	xsd:float

Table 5: Data properties of all entities

5.4 Formal resources generation

Secondly, the obtained ontologies have been uploaded to *Karma Data Integration Tool* along with the final datasets. Based on the available information, related schema and the ER diagram, the links between the schema and the csv datasets columns of the data properties are established and the final result (i.e. Karma models) is extracted in TTL format (GitHub link). Here are some examples.

Note: the figures below are intended to provide an intuition of the process and may be different in their final version.

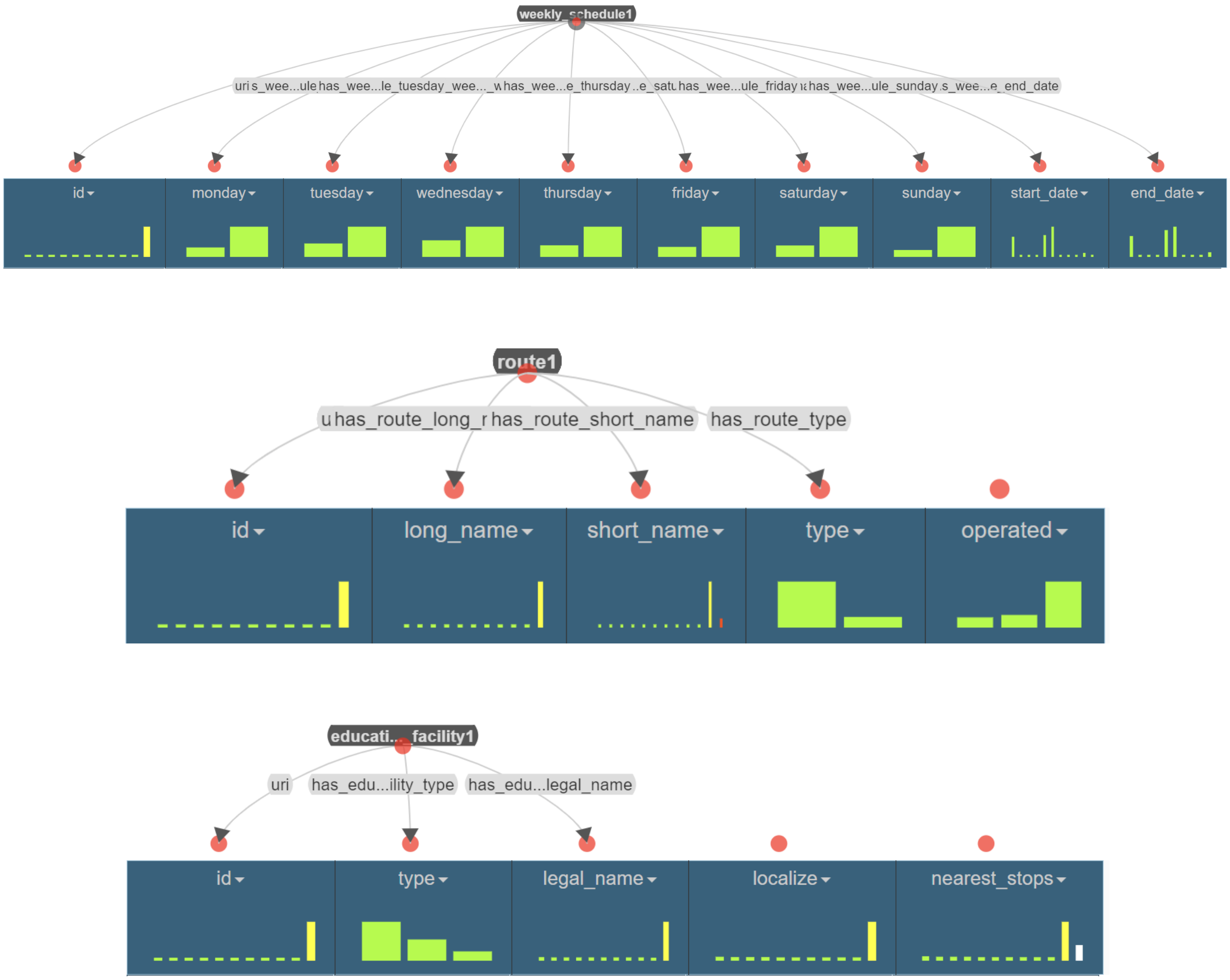


Figure 4: Examples of Karma models for individual dataset

5.5 Summary of the final datasets

Here is provided an overview of the obtained datasets: for each of them a brief description. The files listed in the following table is available in the *"Parsed data"* folder of the repository[2].
Note: the files in this folder have been excluded from *Git* since they depend on the content of files in *"Raw data"*. To generate them please follow the instruction provided in README.md.

[2]After running the scrips and moving the files contained in the folder *tmp_df*. These files have been excluded from the repository because of their size.

Name	Description
Education	Datasets regarding educational facilities.
educational_facilities.csv	This file contains information about middle/high schools and universities in the autonomous province of Trento. It is the result of the *merge of middle_school.csv, high_school.csv* and *univeristy.csv.*
Transportation	Datasets regarding busses and trains. For more details refer to GTFS reference but recall that in this project, in some cases, files may differ from the standard.
agencies.csv	Project related transportation agencies information.
routes.csv	The different "routes" an agency provides e.g. the 5 bus route from Piazza Dante to Oltrecastello.
stops.csv	Bus or train stops (linked to place).
trip_stops.csv	Stops due in different trips.
trips.csv	Bus or Train trips.
EndUser	Datasets regarding personas.
users.csv	This dataset contains the informations regarding a user such as the domicile, the residence, the place where he works.
shifts.csv	Users' tasks/appointments that require a *Trip*, they may or may not repeat. This file contains both geographical and temporal data.
Common	Dataset containing information from multiple of the just considered "domains".
calendars.csv	*Transportation*: Trips weekly schedule e.g. the trip t is available on Mondays and Fridays. *EndUser*: Users shifts weekly schedule i.e. recurring tasks/appointments.
calendar_exceptions.csv	*Transportation*: Single-day modifications to trips e.g. today the scheduled 8AM bus trip t will not be available. *EndUser*: Single-day modifications to shifts. They express one-time events or exceptions like "a class has been canceled" or "today there is an extra lecture".
positions.csv	*Transportation*: the geographical locations of bus or train stops. *Education*: the geographical locations of schools and universities. *EndUser*: the geographical locations of users residence and domicile.

Table 6: Summary of final datasets

6 Language Definition

The language definition phase aims to refine the language (concepts and words) employed for representing the information needed to fulfill the project purpose.
In this phase, producer activities are centered on creating and formalizing the concepts inside each individual dataset. The concepts are identified or created using the kge-annotator. For each (until now) used word/term, the tool is employed to identify the most appropriate concept by searching for synonyms or similar terms. If a sufficiently precise concept already existed in the UKC, it is adopted as is. However, in cases where concepts are either too general or completely absent, they are created as new. For instance, in this report, the semantic of the word *"calendar"* in the context of GTFS is found to be more closely related to the one of *"schedule"* (from the UKC). However, since it lacks specificity, particularly the absence of weekly periodicity, a new concept, *"weekly schedule"*, is introduced.

Consumer activities instead use what has been produced in the previous activities and, looking at a wider scope, i.e. the project purpose, consider the concepts involved in data composition, i.e.

relations between e-types.

Note for the reader: the names of the entities, object, and data properties, at the beginning, are defined with a not properly correct name which is changed and defined during this phase. The following tables summarize the producer activity.

Language concepts for e-types

Initial concept word	Language concept	ID	Description
Education			
educational organization	educational facility	GID-118022	a facility relating to the process of education
Transportation			
agency	provider	GID-57474	someone whose business is to supply a particular service or commodity
route	bus route	GID-45936	the route regularly followed by a passenger bus
route	train route	GID-16009*	the route regularly followed by a passenger train
stop	bus stop	GID-45937	a place on a bus route where buses stop to discharge and take on passengers
stop	train stop	GID-16010*	a place on a train route where trains stop to discharge and take on passengers
trip stop	trip stop	GID-16003*	programmed action of stopping during a journey with a duration
trip	trip	GID-1570	a journey for some purpose (usually including the return)
EndUser			
commitment	shift	GID-1704	the act of moving from one place to another
user/person	end user	GID-53816	the ultimate user for which something is intended
domicile	domicile	GID-17703	housing that someone is living in
residence	residence	GID-46085	any address at which you dwell more than temporarily
EndUser and	Transportation		
calendar	weekly schedule	GID-16001*	a weekly organized plan with a start date and an end date
calendar exception	schedule exception	GID-16002	an instance that does not conform to a schedule
EndUser, Transportation	and Education		
place	position	GID-27990	the spatial property of a place where or way in which something is situated

Table 7: In this table is presented how "generic" words (first column) regarding e-types have been formalized and assigned to a UKC concept (second and third columns). The last column provides a description of what the concept means. GIDs marked with a "*" are new concepts specifically created for this project.

Language concepts for e-types attributes (data properties)

Property name	ID	Description
Education		
has_educational_facility_legal_name	schema.org	The official name of the organization, e.g. the registered company name
has_educational_facility_type	GID-31834	a subdivision of a particular kind of thing
EndUser		
has_end_user_name	GID-16021*	the unique and identifying label assigned to an individual within a social or cultural context
has_end_user_occupation	GID-2999	any activity that occupies a person's attention
has_end_user_special_needs	GID-16020*	unique and specific requirements or assistance that individuals may require due to physical, cognitive, emotional, or developmental differences
Shift		
has_shift_arrive_before	GID-16022*	the specific temporal parameter indicating the designated time by which an individual, object, or event is expected to reach a particular location or point in time
has_shift_leave_after	GID-16023*	the predefined temporal parameter indicating the designated time after which an individual, object, or event is permitted or expected to depart from a specific location or conclude a particular activity
WeeklySchedule		
has_weekly_schedule_start_date	GID-16024*	the date at which something is supposed to begin
has_weekly_schedule_end_date	GID-16025*	the date at which something is supposed to end
has_weekly_schedule_monday	GID-80758	the second day of the week; the first working day
has_weekly_schedule_tuesday	GID-80759	the third day of the week; the second working day
has_weekly_schedule_wednesday	GID-80760	the fourth day of the week; the third working day
has_weekly_schedule_thursday	GID-80761	the fifth day of the week; the fourth working day
has_weekly_schedule_friday	GID-80762	the sixth day of the week; the fifth working day
has_weekly_schedule_saturday	GID-80763	the seventh and last day of the week; observed as the Sabbath by Jews and some Christians
has_weekly_schedule_sunday	GID-80757	first day of the week; observed as a day of rest and worship by most Christians
ScheduleException		
has_schedule_exception_date	GID-16027*	specific point on the temporal continuum, typically expressed through a combination of numerical values representing the day, month, and year within a calendrical system
has_schedule_exception_type	GID-31834	a subdivision of a particular kind of thing
Position		
has_position_address	schema.org	Physical address of the item
has_position_latitude	schema.org	The latitude of a location. For example 37.42242
has_position_longitude	schema.org	The longitude of a location. For example -122.08585
Provider		
has_provider_name	GID-16031	the unique and identifying label assigned to a provider

Route		
has_route_type	GID-31834	a subdivision of a particular kind of thing
has_route_long_name	GID-16034*	a nomenclature, title, or designation that exceeds the customary length or character count typically associated with names or labels
has_route_short_name	GID-16032*	a concise and brief nomenclature, label, or identifier used to represent an entity, concept, or object
Trip		
has_trip_headsign	GID-16035*	a distinct indicator or label, typically displayed on transportation vehicles such as buses or trains, that conveys the destination, route, or final destination of the vehicle. It serves as a concise and easily recognizable identifier for passengers, providing essential information about the vehicle's intended direction or endpoint
has_trip_direction	GID-16036*	the orientation, course, or path along which movement or progress occurs
has_trip_accessibility	GID-16037*	the inclusive quality or characteristic of a system, environment, or information that enables easy, equitable, and unimpeded use and interaction by individuals of diverse abilities, backgrounds, or circumstances
TripStop		
has_trip_stop_arrival_time	GID-80845	the time at which a public conveyance is scheduled to arrive at a given destination
has_trip_stop_departure_time	GID-80846	the time at which a public conveyance is scheduled to depart from a given point of origin
has_trip_stop_stop_sequence	GID-16038*	the systematic and ordered arrangement of stops or locations along a route, path, or journey
Stop		
has_stop_name	GID-16039*	a distinctive label or identifier assigned to a specific location or point along a transportation route
has_stop_type	GID-31834	a subdivision of a particular kind of thing

Table 8: In this table is presented how "generic" words (first column) regarding e-types properties have been formalized and assigned to a UKC concept (second and third columns). The last column provides a description of what the concept means. GIDs marked with a "*" are new concepts specifically created for this project.

Language concepts for e-types relations (object properties)

Language concept	ID	Description
Transport-Transport		
operated	GID-16011*	the action of functioning or carrying out tasks within an organization
of	GID-16041	a portion of
characterized	GID-16013*	the state of being distinctly identified or marked by specific qualities, features, or attributes
at	GID-16040*	a spatial point of reference, indicating a specific location, in relation to which an action, state, or event occurs
EndUser-EndUser		
domiciled	GID-16015*	state or condition of having a temporary established dwelling in a specific place
reside	GID-16016*	state or condition of having one's legal residence in a specific place
involvement	GID-16019*	the active participation, engagement, or inclusion of an individual in a particular activity, situation, or undertaking
Common (mixed)		
localized	GID-89701	confined or restricted to a particular location
nearest	GID-116831	(superlative of 'near' or close') within the shortest distance
avaiability_schedule_exception	GID-16043	a situation where there is a deviation or special circumstance affecting the regular availability schedule of a resource, service, or system
avaiability_schedule	GID-16042	a predetermined plan or timetable that outlines the periods during which a particular resource, service, or system is accessible, operational, or available for use
occurence_schedule_exception	GID-16047	a deviation or irregularity from the established timetable or plan outlining when specific events, incidents, or activities are expected to happen
occurence_schedule	GID-16046	timetable, plan, or systematic arrangement that outlines the timing or schedule of events, incidents, or instances of something happening
from	GID-16044	the point of origin or source position
to	GID-16045	the destination or target position
work	GID-2962	activity directed toward making or doing something

Table 9: In this table is presented how "generic" words (first column) regarding e-types relationships have been formalized and assigned to a UKC concept (second and third columns). The last column provides a description of what the concept means. GIDs marked with a "*" are new concepts specifically created for this project.

6.1 Dataset filtering

During this phase, concepts have been identified by simultaneously considering both data-related and general-purpose criteria, resulting in concepts that already conform to the data. As a result, no additional filtering or data modification is required.

In conclusion, having formalized the concepts involved in the project, from now on, any reference to e-types and relationships is complied with the *Language concept* column in tables 7 and 9. For instance *agency* are referred as *provider*.

7 Knowledge Definition

This section is dedicated to the description of the kTelos phase. Starting from the collected project resources, the formalized purpose (partly expressed by the ER model) and the collected data, the goal is to produce the final KG's teleontology. In particular, the knowledge resources produced in this phase aims at unifying the representation of the information, improving the interoperability and reusability of the final KG(s). By building knowledge resources reusing as much as possible well-known standard domain ontologies and data schema.

7.1 Ontologies

This section describes the *Top-Down* knowledge definition phase of the *kTelos* process. The goal is to reuse Lightweight Ontologies already aligned to the UKC to define an high-level view of the entities involved in the project.
The primary source of ontologies is Datascientia LiveKnowledge, while Schema.org has been used as a fallback.

7.1.1 Datascientia LiveKnowledge

The initial idea is to incorporate certain entities from the OSM Lightweight Ontology. For example, the entity *building* is initially considered, with the intention of later specializing it in teleology to specifically identify educational facilities and residential buildings. Unfortunately, the OSM ontology lacks any discernible structure. Consequently, this deficiency leads to the adoption of the structure from Schema.org for this section. The OSM ontology also includes entities related to transportation, but given their focus on the concept of *"place"* and the availability of alternative options, these are disregarded.

Additionally, the GTFS ontology, detailing entities pertinent to the GTFS standard, is employed. Only the relevant entities essential to our teleology are extracted from this ontology. The specific choices made are summarized in Table 10. For instance, while *"Shape"* is a common entity in the GTFS standard, it is not pertinent to the objectives of this project, therefore it has been excluded.

7.1.2 Schema.org

What still needs to be organized pertains to specific locations, educational facilities, and users' residences. As it is not feasible to repurpose existing *Datascientia LiveKnowledge* ontologies, the decision is made to use entities from Schema.org. Specifically, the chosen entities include:

- Thing > Place > CivicStructure > *EducationalOrganization*

- Thing > Intangible > StructuredValue > *GeoCoordinates*

- Thing > Place > *Residence*

7.1.3 Custom

Unluckily, given the specificity of the user segment of this project, no available ontology fits the purpose. For this reason, a really simple ontology has been created; it includes the entity *PersonLife* which specializes in *Schedule*, *Event* and *Person*. In future this ontology should be replaced with a more appropriate one.

7.1.4 Final result

Here a summary of the referenced ontologies and the selected entities.

Source ontology	Entity
GTFS ontology	agent
GTFS ontology	route
GTFS ontology	service rule
GTFS ontology	stop
GTFS ontology	trip stop
GTFS ontology	trip
Schema.org	Residence
Schema.org	EducationalOrganization
Schema.org	GeoCoordinates
Schema.org	Place
Schema.org	CivicStructure
Schema.org	Intangible
Schema.org	StructuredValue

Table 10: Summary of the final project ontology

7.2 Teleology

This section describes the *Bootm-Up* knowledge definition phase of the *kTelos* process. The goal is to model a teleology which fits the project purpose and data, in other words, define a teleology that is aligned to the requirements modelled as Competency Questions.

As described in the previous sections, there are three sub-domains to consider: transportation, education and end user. Starting from the ER model and the datasets descriptions, the following three sections of the teleology have been defined.
Transportation (figure 5) an extract of GTFS entities, the different names are due to the language alignment.
End user (figure 6) describes the personas, where they live and the shifts they do.
Education (figure 7) describes where educational facilities are located.

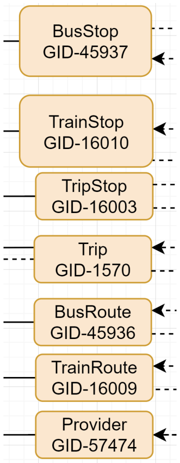


Figure 5: Transportation teleology slice

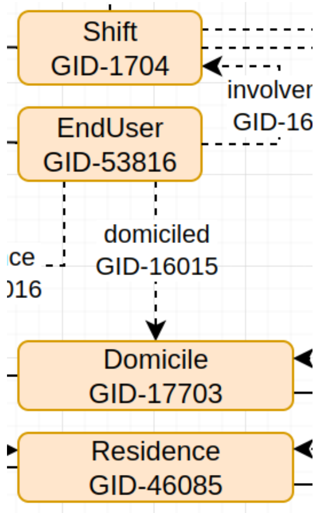


Figure 6: End user teleology slice

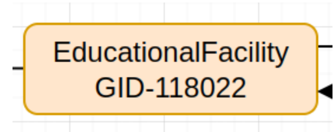


Figure 7: Education teleology slice

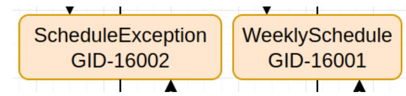


Figure 8: End user and transportation common teleology slice

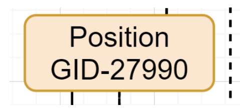


Figure 9: Common teleology slice

Common (figure 8 and 9) describes the weekly recurrence of some events, it is used in the context of transportation trips and end user shifts.

Finally, relations between teleology entities have been defined.

Entities
BusStop
TrainStop
BusRoute
TrainRoute
TripStop
Trip
Provider
ScheduleException
WeeklySchedule
Shift
EndUser
Domicile
Residence
EducationalFacility
Position

Entities	Relation	Entities
BusRoute, TrainRoute	operated	Provider
Trip	characterized	BusRoute, TrainRoute
TripStop	of	Trip
BusStop, TrainStop	at	TripStop
BusStop, TrainStop	localized	Position
Trip	avaiability schedule	WeeklySchedule
Trip	avaiability schedule exception	ScheduleException
Shift	occurence schedule	WeeklySchedule
Shift	occurence schedule	ScheduleException
Shift	involvement	EndUser
Shift	from	Position
Shift	to	Position
EndUser	domiciled	Domicile
EndUser	reside	Residence
EndUser	work	Position
Domicile, Residence	localized	Position
Domicile, Residence	nearest	Position
EducationalFacility	localized	Position
EducationalFacility	nearest	Position

Table 11: Summary of the final project's teleology entities and relations between them

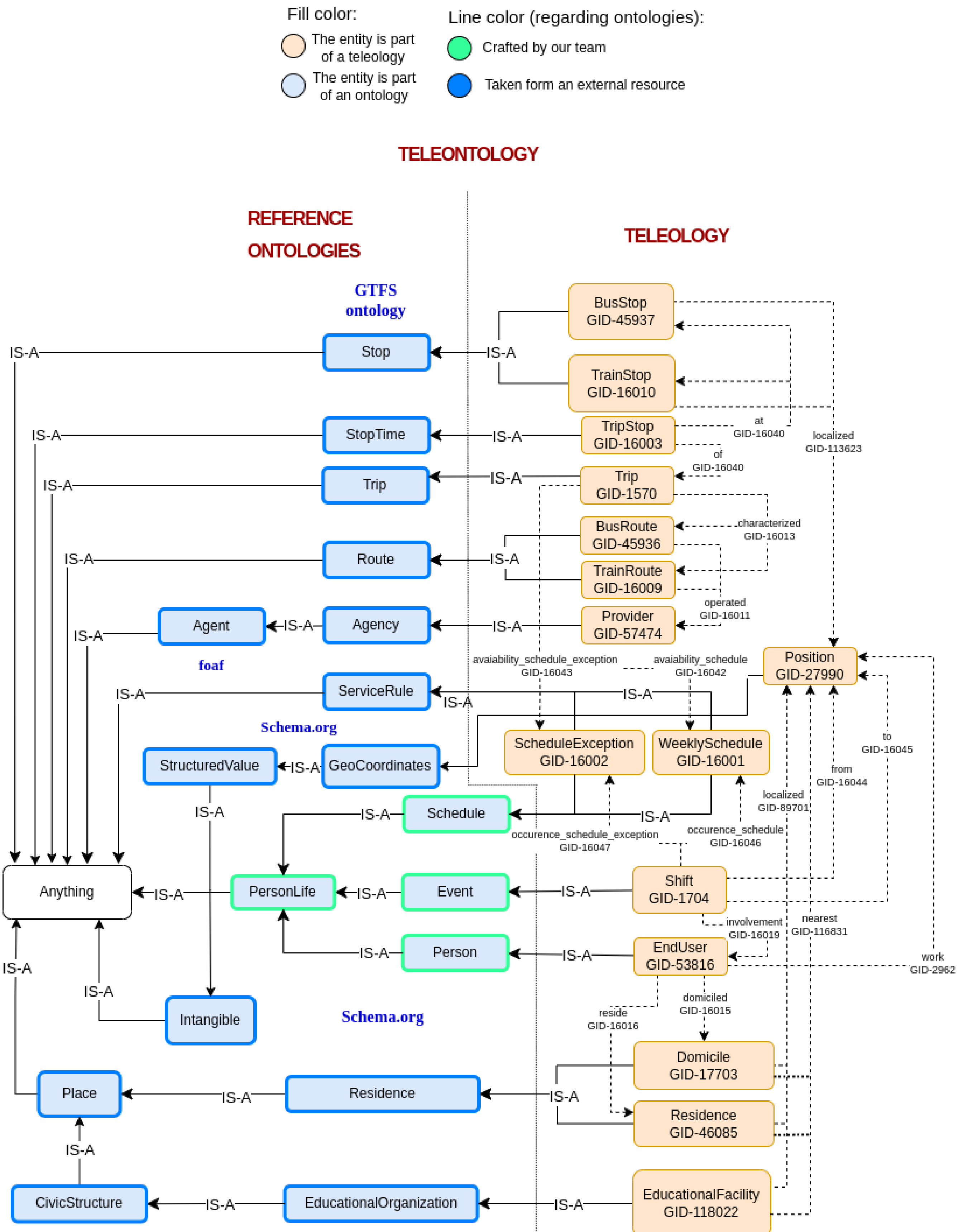
7.3 Teleontology

This section describes the *Middle-Out* knowledge definition phase of the *kTelos* process. The goal is to align the teleology (specific for this project) grounded into the Lightweight Ontology (general purpose) to generate a Teleontology. The identified hierarchical (*IS-A*) relations are summarized in Table 12.

Ontology entity	Teleology entities
Stop	BusStop, TrainStop
TripStop	TripStop
Trip	Trip
Route	BusRoute, TrainRoute
Agent	Provider
ServiceRule	ScheduleException, WeeklySchedule
Schedule	ScheduleException, WeeklySchedule
Event	Shift
Person	EndUser
Residence	Domicile, Residence
EducationalOrganization	EducationalFacility

Table 12: Summary of the final project teleontology IS-A relations

Figure 10 contains the final teleontology schema. This will be the reference to build the final teleontology in *Protege*.


Fill color:
The entity is part of a teleology
The entity is part of an ontology
Line color (regarding ontologies):
Crafted by our team
Taken form an external resource
TELEONTOLOGY
REFERENCE ONTOLOGIES
TELEOLOGY
GTFS ontology
Stop
StopTime
Trip
Route
Agent
Agency
foaf
ServiceRule
Schema.org
StructuredValue
GeoCoordinates
Schedule
Anything
PersonLife
Event
Person
Intangible
Schema.org
Place
Residence
CivicStructure
EducationalOrganization
BusStop GID-45937
TrainStop GID-16010
TripStop GID-16003
Trip GID-1570
BusRoute GID-45936
TrainRoute GID-16009
Provider GID-57474
Position GID-27990
ScheduleException GID-16002
WeeklySchedule GID-16001
Shift GID-1704
EndUser GID-53816
Domicile GID-17703
Residence GID-46085
EducationalFacility GID-118022
at GID-16040
of GID-16040
localized GID-113623
characterized GID-16013
operated GID-16011
avaiability_schedule_exception GID-16043
avaiability_schedule GID-16042
to GID-16045
from GID-16044
localized GID-89701
occurence_schedule_exception GID-16047
occurence_schedule GID-16046
involvement GID-16019
nearest GID-116831
work GID-2962
reside GID-16016
domiciled GID-16015
IS-A

The teleontology allows the reuse of this project data. During this phase, as for the previous ones, activities have been classified in two sides: producer and consumer. At producer side the objective is to model interoperable ontologies, for each datasets to be created. This means that more ontologies files are produced, one for each KG to be generated by the Producer.
On the other hand, at consumer side the objective is to model a single unique interoperable ontology, for the single composed final KG. In this case a single ontology file is produced.

Every term/word/concept used during this phase has been aligned to the UKC, meaning that has a unique identifier (GID) that refers to the exact intended meaning of a word in its context. For instance, the terms *specifies avaiability_GID-16018* and *specifies occurence_GID-16017* are really similar but, in the first case the focus in on *a service* that is *"made available"*, in the second one instead the focus is on *"something that happens"* to a *person*; different GIDs allow to make this distinction.

For more details on how this words/concepts have been identified refer to the previous section Language Definition.

8 Data Definition

This section is dedicated to describing the last phase of the methodology called Data Definition. Unlike the previous section, the organization of the current one follows a single dimension, the one considering the distinction between producer and consumer processes. The division between knowledge and data activities in this section is not defined, because in this phase the two layers are merged to form a single data structure composed by the knowledge structures defined in the last section, and the aligned dataset. Starting from the data resources cleaned and aligned, plus the teleontology, the goal is to produce a structured Knowledge Graph including both the two layers.

It is necessary to handle the meaning heterogeneity to produce a KG suitable to satisfy the initial purpose. To this end, the last phase of the iTelos methodology is structured in three different activities: *entity matching*, *entity identification*, and *entity mapping*.

8.1 Entity Matching

The real words entities, represented by their values, can be represented through different properties, and properties values, within different datasets. This is known as the entity matching problem, and it has also two main consequences:

1. schema layer: the need to find the right set of proprieties between the different datasets where multiple representations of the same entity can be present;
2. data layer: the need to set the correct property values, if multiples representations share the same properties, but having different values.

Throughout the course of this project, neither of the aforementioned issues was encountered while considering available datasets. However, some Competency Questions require information about transportation in Val Di Non, unfortunately, no GTFS dataset is publicly available regarding FTM, Valsugana and Verona-Bolzano trains. As described in Section 5.2.2, data has been scraped from the web. The results of the scraping are non-structured and mixed json files. These files have been manipulated so to obtain a GTFS-like dataset to be then merged with the others.

In the event that either of these issues occured in other dataset, they could have been readily addressed by the scripts designed to parse, clean, and merge data.

8.2 Entity Identification

The next step is to identify an entity within a single dataset and then adopt the same type of identification, if the same entity represented in two (or more) different ways, within different datasets.

The data integration process involves the reworking of information across diverse datasets. The decision is made to consolidate a universal dataset containing location details, including address, latitude, and longitude, for entities such as Educational Facilities, Stops, and End Users. Each record in the dataset is uniquely identified by an "id" column, facilitating easy association with the corresponding entity because the ids are prefixed based on the entity type, with conventions like *pos_edu_fac_* denoting the address of an educational facility, *pos_user_dom_* for the domicile address of an end-user, *pos_user_res_* for the residence address of an end-user, *pos_stop_bus_* for a bus stop address, and *pos_stop_train_* for a train stop position.

Similarly for datasets *WeeklySchedule* and *ScheduleExcpetion* which are common to *Users* and *Transportation*.

8.3 Entity Mapping

The last activity aims at concretely merging the information representation defined in the teleontology, with the relative information values in the datasets. The activity is composed by many mapping operations that concretely define the solution to the entity matching problem. Moreover, a specific type of mapping operation is performed to concretely define the identifiers for the entities, to be considered the final KG. The entity mapping is performed by using the Karma tool.

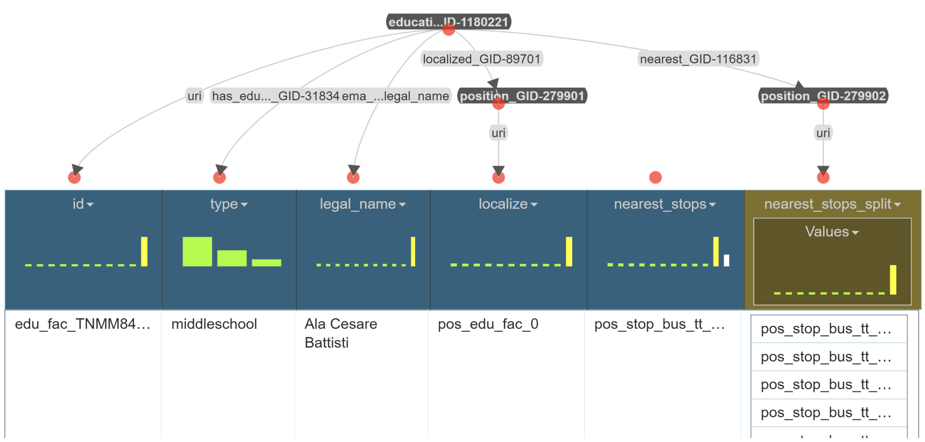


Figure 11: Educational Facilities - Karma Entity Mapping

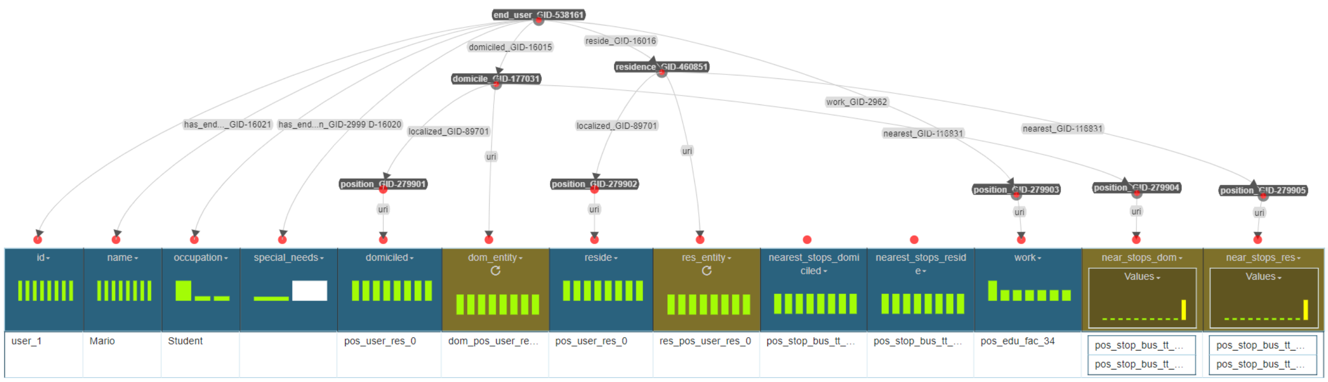


Figure 12: End User - Karma Entity Mapping

Karma allows the user to produce a KG file for each dataset handled. This means that, also at the Consumer side, the output of the Data Definition phase is a set of KG files.

8.3.1 Producer activities

At producer side, the files remain separate in order to be exploited for other purposes. The KG produced is formalized at language side (Aligned with UKC concepts) and at knowledge side (structured with a teleontology). This means that more KG files are produced, one for each KG to be generated.

Figures 11 and 12 are examples of how data is linked with the corresponding Knowledge and Data schema. For each column in the each dataset, the corresponding data property within the teleontology has been identified. For example, in Figure 11, the column *type* in the *educational_facility* dataset has been mapped to the data property *has_educational_facility_type*.

Furthermore, for columns where multiple values are found, a Karma function called split values is used. This allows to have multiple values within a single column, such as *nearest_stops* in Figure 11 and 12. In fact, near a specific location, such as a user's home or the building where someone works, there may be multiple stops available. A stop is considered near in a range of 500 meters.

To conclude, in Figure 12, it is evident how columns with values that are related between two datasets have been mapped. For instance, *domiciled* and *reside* columns in the *end_user* dataset are linked to the corresponding column *id* of the created new entity *domicile* and *residence* dataset through the following object properties: *domiciled* and *reside*. As a result, in the *end_user* dataset only the corresponding id for these columns is found, while, in *position*, associated with the corresponding id, the value of address, latitude, and longitude are located.

8.3.2 Consumer activities

At consumer side, the files are composed together into a single file to define the single purpose. The final KG satisfies the requirements extracted from the user purpose (Competency Questions).

Concretely speaking, for each dataset model, a file containing the data has been obtained with karma. Finally they have been merged (appended one to another) by a script to obtain the final KG.

9 Evaluation

This section aims at describing the evaluation performed at the end of the whole process (producer plus consumer) over the final outcome of the iTelos methodology. The criteria, described below, consider both the Knowledge and Data Layer.

9.1 Knowledge Layer Evaluations

The iTelos methodology provides a set of metrics to be used for the above evaluations. Between them one of the most useful is *Coverage* which describes how much a portion of knowledge is covered by a KG. The Coverage is used as follows, to evaluate the Knowledge Layer for two different objectives:

1. Primary objective based on the purpose satisfaction (Teleontology vs CQs): it is based on how much the final KG is able to satisfy the Competency Queries. So, this means how much the teleontolgy covers the entities and properties extracted from the CQs;
2. Second objective based on the reusability (Teleontology vs Reference Ontologies): how much the teleontology covers the etypes, and properties, extracted from the reference ontologies.

9.1.1 Teleontology

In the table below, there is a summary that takes into account the total number of etypes, object properties, and data properties, used for the calculation of the coverage.

	Instances Count
Etypes	33
Object Properties	16
Data Properties	33

Table 13: Final Teleontology Evaluation Summary

9.1.2 Teleontology vs CQs

Considering the table 2, and the Competency Questions 4.3, in reference to the final values obtained from the Teleontology's table 13, the coverage of the etypes, object properties, and data properties is calculated as follows. For example, for the etype, given a set of (CQ_E), the etype coverage (Cov_E) of the Teleontology (T) is:

$$Cov_E(CQ_E) = \frac{|CQ_E \cap T_E|}{|CQ_E|}$$

where CQ_E is the number of etypes extracted from the CQs, and T_E is the number of etypes of the Teleontology.

Below, there is a table with the final evaluation, considering the etypes, object properties, and data properties coverage.

	Etypes	Cov_E	Object Properties	Cov_{OP}	Data Properties	Cov_{DP}
Total identified from CQs	13		15		28	
Total defined for the project	33	100%	16	100%	33	100%

Table 14: Teleontology vs Competency Questions Coverage

This table shows that for each criteria, the final Teleontology defines more etypes, data and object properties. This is due to the fact that during the initial phases of this project, the specific knowledge design choices and needs were not complete. They have been refined during the development of the project which lead to defining a better and complete knowledge structure to fulfil the purpose.

9.1.3 Teleontology vs Reference Ontologies

Considering the used ontologies (table 10), in reference to the final values obtained from the Teleontology's table 13, the coverage of the etypes (Cov_E), object properties (Cov_{OP}), and data properties (Cov_{DP}) is calculated as follows. For example, for the etype, given a set of (RO), the etype coverage (Cov_E) of the Teleontology (T) is:

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

where RO_E is the number of etypes extracted from the ROs, and T_E is the number of etypes of the Teleontology.

Below, there is a table with the final evaluation, considering the etypes, object properties, and data properties coverage.

	Etypes	Cov_E	Object Properties	Cov_{OP}	Data Properties	Cov_{DP}
GTFS ontology						
Total in the ontology	30		19		22	
Total reused in the project	9	30%	0	0%	10	45%
OSM ontology						
Total in the ontology	791		0		0	
Total reused in the project	0	0%	0	0%	0	0%
Schema.org						
Total in the ontology	-		-		-	
Total reused in the project	7	-	-	-	4	-

Table 15: Teleontology vs Reference Ontologies Coverage

In reference to the table above, the object properties of the GTFS ontology are not utilized due to their incompleteness and vague language choices.

Regarding the OSM ontology, initially it seems to be a great reference for this project but then, analyzing the etypes it provided, it become clear that their purpose is different from ours.

Regarding *Schema.org*, the missing values in the table are due to the dimension of the ontology: it counts too many etypes and properties, therefore, it is decided to only include in the table the select types and data properties used in the project.

9.2 Data Layer Evaluations

The Data Layer Evaluations is made only for the primary objective which is based on the purpose satisfaction and it aims to understand how the KG is dense and connected. The KG's connectivity is evaluated in two different moments:

- on the final KG, to understand how much the KG is connected at the end of the process;
- during the KG's construction, to understand how much each single dataset improves the connectivity of the final KG.

The connectivity of a KG can be evaluated over two dimensions:

1. entity connectivity which evaluates the grades of connection between the different entities in the KG;
2. property connectivity which evaluates the grades of connection between each single KG's entity and its properties values.

The entity and property connectivity can be calculated by using the connectivity matrix, as represented in the table below.

	Trip	Shift	Position	WeeklySchedule	...
Trip	0.80|0|0.13 $tot : 0.31$			0 $tot : 0$	...
Shift		0.47|0.53 $tot : 0.5$	0.04|0.00 $tot : 0.02$	0.21 $tot : 0.21$	...
Position			0.99|0.04|0.04 $tot : 0.35$		...
WeeklySchedule				0|...|0 $tot : 0$	...
...	...	...	...	...	...

Table 16: Connectivity Matrix (slice)

Each cell of the connectivity matrix shows how much entities (referenced by their etype) are connected to others.

The main diagonal of the matrix shows the percentage of *null* values in the corresponding etype entities data properties. Each value represents a single property while the last one considers them all together. Table 17 provides the information to correlate each value to the corresponding data property (based on the ordering).

Other cells show the connectivity between entities i.e. *non-null* object properties percentage. Rows refer to entity that have a reference in the direction of the others. As for data properties, Table 17 allows to understand exactly which object property values refer to.

This report provides only an example of the matrix since a A4 page may not be enough to visualize it. For the complete version, please refer to the GitHub repository matrix and legend files.

Etype - Etype	Property names
trip - trip	headsign | direction | accessibility
shift - shift	arrive_before | leave_after
position - position	address | latitude | longitude
weekly_schedule - weekly_schedule	monday | ... | sunday | start_date | end_date
trip - weekly_schedule	avaiability_schedule
shift - weekly_schedule	occurence_schedule
shift - position	from | to

Table 17: Connectivity Matrix Legend

9.2.1 Knowledge Graph information statistics

Some statistics on the final Knowledge Graph entities.

Etype	Total number of entities
TripStop	509,214
ScheduleException	246,192
Trip	47,885
WeeklySchedule	41,571
Stop	9,273
Position	9,413
Route	1,347
EducationalFacility	139
Shift	19
EndUser	7
Residence	7
Domicile	7
Provider	3

Table 18: KG information statistics

9.3 Query execution

To conclude the evaluation phase, competency questions have been translated to *SPARQL* queries in order to test the suitability of the KG to satisfy the project purpose. Given their similarity, they all share the same template which consists in the following (ordered) phases:

1. Hyperparameters selection (general parameters):
   - *maximum walk distance*;
   - *maximum wait time*;
   - *trip direction* (for optimization);
   - *trips filter* (e.g. only Trento urban buses)

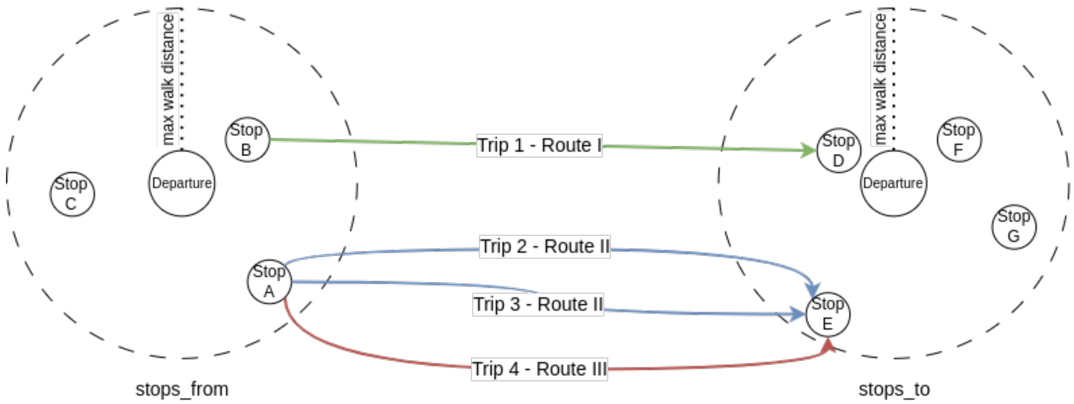


Figure 13: Transportation query intuition

2. Parameters (CQ-dependent):
   - *user name*;
   - *current date and time*;
3. Shift identification: obtained from *weekly schedule* and *exceptions* depending on *current date* and *time*;
4. Possible stops from/to identification: either by
   - physical distance (fig. 13 dashed circles)
   - or statically defined *"nearest stops"* (optimization)

   Once obtained all possible from/to stop, the set $S = stops\_from \times stops\_to$ is obtained.
5. Filter *Trips* depending on their weekly schedule and exceptions: let T be the set containing all trips, the result of this step is the set $T' = date\_trips(current\_date, T)$ where *date_trips* is a predicate that returns the trips in T that happen in the given date (first parameter);
6. Keep only *Trips* that stop at both *from* and *to* stops: more formally, let *stop* be a predicate that, given a trip, returns all the stops it passes by:
   $T'' = \{t \in T' \mid ((stops(t) \cap stops\_from) \cap stops\_to) \neq \emptyset\}$;[3]
7. Filter *Trips* depending on *arrive_before* and/or *leave_after* (defined by the shift);
8. Score computation: linear combination of *walking distances*, *wait time* and *trip duration*;
9. Ordering by score (ascendant) so that the best results are displayed first;
10. Eventual group by trip. This may be needed in the case that trips stop are really close one to another. Eventual limit on results obtained.

[3]This may be performed before step 5 because of better optimization given by the relation $at\_GID - 16040$.

Listing 1: Query template

```
PREFIX prop: <http://knowdive.disi.unitn.it/etype#>
PREFIX omgeo: <http://www.ontotext.com/owlim/geo#>
PREFIX ofn: <http://www.ontotext.com/sparql/functions/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX spif: <http://spinrdf.org/spif#>

SELECT DISTINCT ?shift ?stop_from_name ?stop_to_name ?route_name ?departure_time ?arrival_time
    ?wait_time ?walk_from_m ?walk_to_m ?trip_duration ?score ?accessibility
WHERE {
    ### 1. Hyperparams
    BIND ("false"^^xsd:boolean as ?wheelchair_need)
    BIND ("30"^^xsd:long as ?max_wait_time)
    BIND ("0.5"^^xsd:float as ?max_stop_dist)
    BIND ("_urban" as ?stops_filter)
    BIND ("1" as ?direction)

    { ### 3. Shift identification
        SELECT ?user ?shift ?arrive_before ?leave_after ?curr_datetime ?curr_weekday
        WHERE {
            ### 2. Parameters (these are here because of SPARQL subqueries evaluation order)
            BIND ("2024-01-29T06:00:00"^^xsd:dateTime as ?curr_datetime)
            BIND ("Jessica" as ?username)

            ## Select only one user
            ?user prop:has_end_user_name_GID-16021 ?username.

            ## Compute weekday
            BIND (ofn:daysBetween(?curr_datetime, "1967-01-01T10:00:00+01:00"^^xsd:dateTime) AS ?days)
            BIND (floor(?days - (7 * floor(?days / 7))) AS ?curr_weekday)

            ## Join user shifts, schedules and related data properties
            ?shift prop:involvement_GID-16019 ?user.
             OPTIONAL {
                ?shift prop:occurence_schedule_GID-16046 ?ws.
                ?ws prop:has_weekly_schedule_monday_GID-80758 ?monday;
                    prop:has_weekly_schedule_tuesday_GID-80759 ?tuesday;
                    prop:has_weekly_schedule_wednesday_GID-80760 ?wednesday;
                    prop:has_weekly_schedule_thursday_GID-80761 ?thursday;
                    prop:has_weekly_schedule_friday_GID-80762 ?friday;
                    prop:has_weekly_schedule_saturday_GID-80763 ?saturday;
                    prop:has_weekly_schedule_sunday_GID-80757 ?sunday.
            }.
            OPTIONAL {
                ?shift prop:occurence_schedule_exception_GID-16047 ?ex.
                ?ex prop:has_schedule_exception_date_GID-16027 ?date_ex;
                    prop:has_schedule_exception_type_GID-31834 ?type_ex.
            }.
            OPTIONAL { ?shift prop:has_shift_arrive_before_GID-16022 ?arrive_before }.
            OPTIONAL { ?shift prop:has_shift_leave_after_GID-16023 ?leave_after }.

            ## Compute time difference between shift and current time
            BIND (xsd:dateTime(CONCAT(spif:dateFormat(?curr_datetime, "yyyy-MM-dd"), "T",
                    IF(BOUND(?arrive_before),?arrive_before,?leave_after))) as ?arr_leave_datetime)
            BIND (ofn:minutesBetween(?arr_leave_datetime, ?curr_datetime) as ?time_diff)
```

```
    ## Exclude already passed shifts
    FILTER (IF(BOUND(?arr_leave_datetime), ?curr_datetime <= ?arr_leave_datetime, "true"^^xsd:boolean))

        ## Filter based on schedule exception (higher priority) or weekly schedule
        FILTER (IF(BOUND(?ex),
                ## Exception is present
                ?type_ex = "1" && ?date_ex = spif:dateFormat(?curr_datetime, "yyyyMMdd"),
                # NOTICE that here ?type_ex = "0" is not handle because we did not face this case
                # please, refer to the transportation section for the handling of the analogous problem
                ## Exception is NOT present
                (?curr_weekday = "0"^^xsd:decimal && ?monday = "1") ||
                (?curr_weekday = "1"^^xsd:decimal && ?tuesday = "1") ||
                (?curr_weekday = "2"^^xsd:decimal && ?wednesday = "1") ||
                (?curr_weekday = "3"^^xsd:decimal && ?thursday = "1") ||
                (?curr_weekday = "4"^^xsd:decimal && ?friday = "1") ||
                (?curr_weekday = "5"^^xsd:decimal && ?saturday = "1") ||
                (?curr_weekday = "6"^^xsd:decimal && ?sunday = "1")
            )
        )
    }
    ORDER BY DESC(?ex) ?time_diff  # Get the "closest" shift as first result
    LIMIT 1       # Get only one shift
} ### 3. Shift identification END

### 4. Possible stops from/to identification

## Shift positions from / to
?shift prop:from_GID-16044 ?shift_pos_from;
      prop:to_GID-16045 ?shift_pos_to.

{ ## Find "from" nearest stops
    ?shift_pos_from prop:schema_has_position_latitude ?shift_pos_from_lat;
                prop:schema_has_position_longitude ?shift_pos_from_lon.

    ## Using "nearest"
    ?edu_res_dom_from prop:localized_GID-89701 ?shift_pos_from.
    ?edu_res_dom_from prop:nearest_GID-116831 ?stop_pos_from.

    # { ## OR using geo distance
    # {
    # { ?stop_from a prop:bus_stop_GID-45937. }
    #  UNION
    # { ?stop_from a prop:train_stop_GID-16010. }
    # }
    # }

    ?stop_from prop:localized_GID-89701 ?stop_pos_from.

    ## Compute distance anyay (for score)
    ?stop_pos_from prop:schema_has_position_latitude ?stop_from_lat;
               prop:schema_has_position_longitude ?stop_from_lon.
    BIND (omgeo:distance(?shift_pos_from_lat, ?shift_pos_from_lon,
                     ?stop_from_lat, ?stop_from_lon) as ?walk_from)
```

```
    ## Eventual filter on max walk distance
    FILTER (IF(BOUND(?max_stop_dist), ?walk_from <= ?max_stop_dist, "true"^^xsd:boolean))
}

{ ## Find "to" nearest stops
    ?shift_pos_to prop:schema_has_position_latitude ?shift_pos_to_lat;
                 prop:schema_has_position_longitude ?shift_pos_to_lon.

    ## Using "nearest"
    ?edu_res_dom_to prop:localized_GID-89701 ?shift_pos_to.
    ?edu_res_dom_to prop:nearest_GID-116831 ?stop_pos_to.

    # { ## OR using geo distance
    # {
    # { ?stop_to a prop:bus_stop_GID-45937. }
    #  UNION
    # { ?stop_to a prop:train_stop_GID-16010. }
    # }
    # }

    ?stop_to prop:localized_GID-89701 ?stop_pos_to.

    ## Compute distance anyay (for score)
    ?stop_pos_to prop:schema_has_position_latitude ?stop_to_lat;
                prop:schema_has_position_longitude ?stop_to_lon.
    BIND (omgeo:distance(?shift_pos_to_lat, ?shift_pos_to_lon,
                   ?stop_to_lat, ?stop_to_lon) as ?walk_to)

    ## Eventual filter on max walk distance
    FILTER (IF(BOUND(?max_stop_dist), ?walk_to <= ?max_stop_dist, "true"^^xsd:boolean))
}

## Filter to have more control over results (could be moved above...)
FILTER (IF(BOUND(?stops_filter), REGEX(STR(?stop_from), ?stops_filter)
    && REGEX(STR(?stop_to), ?stops_filter), "true"^^xsd:boolean))

### 4. Possible stops from/to identification END
### 6. Keep only Trips that stop at both from and to stops

{ ## Find trips that stop at stop_from
    ?trip_stop_from prop:at_GID-16040 ?stop_from.
    ?trip_stop_from prop:of_GID-16041 ?trip.

    ?trip prop:has_trip_direction_GID-16036 ?trip_direction.
    FILTER (?trip_direction = ?direction)
}

{ ## Filter trips that stop at stop_from only to those that also stop at stop_to
    ?trip_stop_to prop:of_GID-16041 ?trip.
  ?trip_stop_to prop:at_GID-16040 ?stop_to.
}

### 6. Keep only Trips that stop at both from and to stops END
### 5. Filter Trips depending on their weekly schedule and exceptions
```

```
# Filter by trip schedules and exceptions
OPTIONAL {
    ?trip prop:avaiability_schedule_GID-16042 ?trip_ws.
    ?trip_ws prop:has_weekly_schedule_monday_GID-80758 ?monday;
        prop:has_weekly_schedule_tuesday_GID-80759 ?tuesday;
        prop:has_weekly_schedule_wednesday_GID-80760 ?wednesday;
        prop:has_weekly_schedule_thursday_GID-80761 ?thursday;
        prop:has_weekly_schedule_friday_GID-80762 ?friday;
        prop:has_weekly_schedule_saturday_GID-80763 ?saturday;
        prop:has_weekly_schedule_sunday_GID-80757 ?sunday;
        prop:has_weekly_schedule_start_date_GID-16024 ?trip_ws_start_date;
        prop:has_weekly_schedule_end_date_GID-16025 ?trip_ws_end_date.
}.
OPTIONAL {
    ?trip prop:avaiability_schedule_exception_GID-16043 ?trip_ex.
    ?trip_ex prop:has_schedule_exception_date_GID-16027 ?trip_ex_date;
            prop:has_schedule_exception_type_GID-31834 ?trip_ex_type.
    FILTER (?trip_ex_date = spif:dateFormat(?curr_datetime, "yyyyMMdd"))
}.

## Check schedule exceptions
FILTER (IF(BOUND(?trip_ex),
        ## Exception is present
        (?trip_ex_type = "1") || !(?trip_ex_type = "2"),
        ## Exception is NOT present
        IF(BOUND(?trip_ws),
      ## Check weekly schedule validity
            (xsd:date(?curr_datetime) >= spif:parseDate(?trip_ws_start_date, "yyyyMMdd"))
            && (xsd:date(?curr_datetime) <= spif:parseDate(?trip_ws_end_date, "yyyyMMdd"))
            ## Check weekdays
            && (
                (?curr_weekday = "0"^^xsd:decimal && ?monday = "1") ||
                (?curr_weekday = "1"^^xsd:decimal && ?tuesday = "1") ||
                (?curr_weekday = "2"^^xsd:decimal && ?wednesday = "1") ||
                (?curr_weekday = "3"^^xsd:decimal && ?thursday = "1") ||
                (?curr_weekday = "4"^^xsd:decimal && ?friday = "1") ||
                (?curr_weekday = "5"^^xsd:decimal && ?saturday = "1") ||
                (?curr_weekday = "6"^^xsd:decimal && ?sunday = "1")
            ),
            "false"^^xsd:boolean # Neither schedule or exception available
        )
    )
)

### 5. Filter Trips depending on their weekly schedule and exceptions END
### 7. Filter Trips depending on arrive before and/or leave after

## Extract times
?trip_stop_from prop:has_trip_stop_departure_time_GID-80846 ?departure_time.
?trip_stop_to prop:has_trip_stop_arrival_time_GID-80845 ?arrival_time.

## Support variables
BIND (xsd:dateTime(CONCAT("2000-01-01T", ?departure_time)) as ?departure_date).
BIND (xsd:dateTime(CONCAT("2000-01-01T", ?leave_after)) as ?leave_after_date).
```

```
    BIND (xsd:dateTime(CONCAT("2000-01-01T", ?arrival_time)) as ?arrival_date).
    BIND (xsd:dateTime(CONCAT("2000-01-01T", ?arrive_before)) as ?arrive_before_date).

    # IF BOUND arrive_before  => wait time = diff(arrival, arrive_before)
    # ELSE      => wait time = diff(departure, leave_after)
    BIND (IF(BOUND(?arrive_before),
           ofn:minutesBetween(?arrival_date, ?arrive_before_date),
           ofn:minutesBetween(?departure_date, ?leave_after_date)
       ) as ?wait_time)

    # BOUND arrive_before => trip_arrival BEFORE arrive_before
    FILTER (!BOUND(?arrive_before) || (?arrival_date <= ?arrive_before_date))
    # BOUND leave_after => trip_departure AFTER leave_after
    FILTER (!BOUND(?leave_after) || (?departure_date >= ?leave_after_date))

    # Exclude if wait time is more than the limit
    FILTER (IF(BOUND(?max_wait_time), ?wait_time <= ?max_wait_time, "true"^^xsd:boolean))

    # Filter on accessibility: ?wheelchair_need => ?accessibility = "1"
    FILTER (!?wheelchair_need || ?accessibility = "1")

    ### 7. Filter Trips depending on arrive before and/or leave after END

    ## Projection variables
    ?stop_from prop:has_stop_name_GID-16039 ?stop_from_name.
    ?stop_to prop:has_stop_name_GID-16039 ?stop_to_name.
    ?trip prop:characterized_GID-16013 ?route;
       prop:has_trip_accessibility_GID-16037 ?accessibility.
    ?route prop:has_route_short_name_GID-16032 ?route_name.

    ### 8. Score computation
    BIND (xsd:int(?walk_from * 100) as ?walk_from_m)
    BIND (xsd:int(?walk_to * 100) as ?walk_to_m)
    BIND (ofn:minutesBetween(?arrival_date, ?departure_date) as ?trip_duration)
    BIND ((
           ?wait_time
           + ?trip_duration
           + "0.5"^^xsd:float * ?walk_from_m
           + "0.5"^^xsd:float * ?walk_to_m
       ) as ?score
    )
    ### 8. Score computation END

} ORDER BY ASC(?score) ### 9. Ordering by score
```

CQ1 - Mario - FTM

Mario lives in Mollaro and from Monday to Friday needs to go to the middle-school in Mezzolombardo from 8AM to 1PM, except on Thursday he needs to stay in school until 4PM. He only has a train subscription and he must be independent because his parents work all day.

Query template changes

Here the query for Mario. While the parameters for the outward are set on 2024-02-13 which is Tuesday, the return is set on Thursday to report the management of the return at home after 4PM.

```
BIND ("false"^^xsd:boolean as ?wheelchair_need)
BIND ("40"^^xsd:long as ?max_wait_time)
BIND ("0.8"^^xsd:float as ?max_stop_dist)
BIND ("_scraped" as ?stops_filter)

BIND ("1" as ?direction) # For outward while "0" for return

BIND ("2024-02-13T06:00:00"^^xsd:dateTime as ?curr_datetime)
# For outward while "2024-02-15T16:00:00" for return

BIND ("Mario" as ?username)
```

Results outward

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	http://localhost:8080/source/shift_1	"mollaro"	"mezzolombardo"	"FTM"	"07:29:00"	"07:47:00"	"13"^^xsd:long	"44"^^xsd:int	"27"^^xsd:int	"18"^^xsd:long	"66.5"^^xsd:float	"1"
2	http://localhost:8080/source/shift_1	"mollaro"	"mezzolombardo"	"FTM"	"07:06:00"	"07:21:00"	"39"^^xsd:long	"44"^^xsd:int	"27"^^xsd:int	"15"^^xsd:long	"89.5"^^xsd:float	"1"

Results return

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	http://localhost:8080/source/shift_4	"mezzolombardo"	"mollaro"	"FTM"	"16:31:00"	"16:49:00"	"31"^^xsd:long	"27"^^xsd:int	"44"^^xsd:int	"18"^^xsd:long	"84.5"^^xsd:float	"1"
2	http://localhost:8080/source/shift_4	"mezzolombardo"	"mollaro"	"FTM"	"17:08:00"	"17:26:00"	"68"^^xsd:long	"27"^^xsd:int	"44"^^xsd:int	"18"^^xsd:long	"121.5"^^xsd:float	"1"

CQ2 - Jessica - Trento suburbs

Jessica lives in Povo and from Monday to Saturday she has to go to the high-school in the center of Trento. Her hours are from 8AM to 1PM.

Query template changes

In this query, Jessica's shift is represented with the setting of the parameters to find out a lot of different stop buses where Jessica can take the bus. Here, there is a clear example of what is the score - described in the section above - based on linear combination of *max_stop_dist* considered the maximum distance as 0.5 kilometers, *max_wait_time* sets to 45 minutes, and *trip_duration*. Here the solutions for the outward and the return.

```
BIND ("false"^^xsd:boolean as ?wheelchair_need)
BIND ("45"^^xsd:long as ?max_wait_time)
# BIND ("0.5"^^xsd:float as ?max_stop_dist)
BIND ("_urban" as ?stops_filter)

BIND ("1" as ?direction) # For outward while "0" for return

BIND ("2024-01-29T06:00:00"^^xsd:dateTime as ?curr_datetime)
# For outward while "2024-01-29T12:00:00" for return

BIND ("Jessica" as ?username)
```

Results outward

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_5	"Povo Piazza Manci"	"Venezia Cave"	"5"	"07:37:00"	"07:44:00"	"16"^^xsd:long	"12"^^xsd:int	"24"^^xsd:int	"7"^^xsd:long	"41.0"^^xsd:float	"1"
2	data:shift_5	"Povo "Centro Civico""	"Venezia Cave"	"5"	"07:38:00"	"07:44:00"	"16"^^xsd:long	"30"^^xsd:int	"24"^^xsd:int	"6"^^xsd:long	"49.0"^^xsd:float	"1"
3	data:shift_5	"Povo "Polo Sociale""	"Venezia Cave"	"5"	"07:36:00"	"07:44:00"	"16"^^xsd:long	"37"^^xsd:int	"24"^^xsd:int	"8"^^xsd:long	"54.5"^^xsd:float	"1"
4	data:shift_5	"Povo Piazza Manci"	"Venezia Cave"	"5"	"07:17:00"	"07:24:00"	"36"^^xsd:long	"12"^^xsd:int	"24"^^xsd:int	"7"^^xsd:long	"61.0"^^xsd:float	"1"
5	data:shift_5	"Povo Piazza Manci"	"Venezia Cave"	"5"	"07:14:00"	"07:20:00"	"40"^^xsd:long	"12"^^xsd:int	"24"^^xsd:int	"6"^^xsd:long	"64.0"^^xsd:float	"1"
6	data:shift_5	"Povo "Centro Civico""	"Venezia Cave"	"5"	"07:18:00"	"07:24:00"	"36"^^xsd:long	"30"^^xsd:int	"24"^^xsd:int	"6"^^xsd:long	"69.0"^^xsd:float	"1"
7	data:shift_5	"Povo PantÃ¨"	"Venezia Cave"	"5"	"07:15:00"	"07:20:00"	"40"^^xsd:long	"34"^^xsd:int	"24"^^xsd:int	"5"^^xsd:long	"74.0"^^xsd:float	"1"
8	data:shift_5	"Povo "Polo Sociale""	"Venezia Cave"	"5"	"07:16:00"	"07:24:00"	"36"^^xsd:long	"37"^^xsd:int	"24"^^xsd:int	"8"^^xsd:long	"74.5"^^xsd:float	"1"
9	data:shift_5	"Povo "Polo Sociale""	"Venezia Cave"	"5"	"07:13:00"	"07:20:00"	"40"^^xsd:long	"37"^^xsd:int	"24"^^xsd:int	"7"^^xsd:long	"77.5"^^xsd:float	"1"

Results return

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_6	"Venezia Cave"	"Povo Piazza Manci"	"5"	"13:14:00"	"13:22:00"	"14"^^xsd:long	"18"^^xsd:int	"13"^^xsd:int	"8"^^xsd:long	"37.5"^^xsd:float	"1"
2	data:shift_6	"Venezia Cave"	"Povo Piazza Manci"	"5"	"13:17:00"	"13:26:00"	"17"^^xsd:long	"18"^^xsd:int	"13"^^xsd:int	"9"^^xsd:long	"41.5"^^xsd:float	"1"
3	data:shift_6	"S.Francesco Porta Nuova"	"Povo Piazza Manci"	"5"	"13:13:00"	"13:22:00"	"13"^^xsd:long	"29"^^xsd:int	"13"^^xsd:int	"9"^^xsd:long	"43.0"^^xsd:float	"1"
4	data:shift_6	"Venezia Cave"	"Povo S.Agata"	"5"	"13:14:00"	"13:22:00"	"14"^^xsd:long	"18"^^xsd:int	"28"^^xsd:int	"8"^^xsd:long	"45.0"^^xsd:float	"1"
5	data:shift_6	"Venezia Cave"	"Povo PantÃ¨"	"5"	"13:14:00"	"13:21:00"	"14"^^xsd:long	"18"^^xsd:int	"32"^^xsd:int	"7"^^xsd:long	"46.0"^^xsd:float	"1"
6	data:shift_6	"S.Francesco Porta Nuova"	"Povo Piazza Manci"	"5"	"13:16:00"	"13:26:00"	"16"^^xsd:long	"29"^^xsd:int	"13"^^xsd:int	"10"^^xsd:long	"47.0"^^xsd:float	"1"
7	data:shift_6	"Venezia Cave"	"Povo "Centro Civico""	"5"	"13:17:00"	"13:25:00"	"17"^^xsd:long	"18"^^xsd:int	"27"^^xsd:int	"8"^^xsd:long	"47.5"^^xsd:float	"1"

CQ3 - Luca - Trento center (wheelchair)

Luca lives in Trento and he goes to high-school from Monday to Saturday in the center of Trento from 8AM to 1PM, but he is in a wheelchair, and he must be independent using suitable and accessible buses.

Query template changes

In this query, Luca has a wheelchair so he needs an accessible bus, i.e. *accessibility="1"*. As a result, the proximity to the bus stop has also been reduced as much as possible, with a *max_stop_dist* of 0.2 kilometers. Also the *max_wait_time* sets at 15 or 25 minutes. Here the solutions for the outward and the return.

```
BIND ("true"^^xsd:boolean as ?wheelchair_need)

BIND ("15"^^xsd:long as ?max_wait_time) # For outward while "25" for return

BIND ("0.2"^^xsd:float as ?max_stop_dist)
BIND ("_urban" as ?stops_filter)
BIND ("0" as ?direction) # For outward while "1" for return

BIND ("2024-01-29T06:00:00"^^xsd:dateTime as ?curr_datetime)
# For outward while "2024-01-29T12:00:00" for return

BIND ("Luca" as ?username)
```

Results outward

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_7	"Degasperi Anna Frank"	"Rosmini "Cimitero" / Muse"	"4"	"07:48:00"	"07:55:00"	"5"^^xsd:long	"15"^^xsd:int	"5"^^xsd:int	"7"^^xsd:long	"22.0"^^xsd:float	"1"
2	data:shift_7	"Degasperi Anna Frank"	"Rosmini "Cimitero" / Muse"	"4"	"07:44:00"	"07:51:00"	"9"^^xsd:long	"15"^^xsd:int	"5"^^xsd:int	"7"^^xsd:long	"26.0"^^xsd:float	"1"

Results return

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_8	"Rosmini "Cimitero" / Muse"	"Degasperi Anna Frank"	"4"	"13:09:00"	"13:02:00"	"9"^^xsd:long	"5"^^xsd:int	"15"^^xsd:int	"7"^^xsd:long	"26.0"^^xsd:float	"1"
2	data:shift_8	"Rosmini "Cimitero" / Muse"	"Degasperi Anna Frank"	"4"	"13:24:00"	"13:17:00"	"24"^^xsd:long	"5"^^xsd:int	"15"^^xsd:int	"7"^^xsd:long	"41.0"^^xsd:float	"1"

CQ4 - Gian - Trento-Rovereto

Gian is a Rovereto university student who lives in Trento and he needs to reach the university from Monday to Wednesday from 10:30AM, and he needs to arrive at home before 4:30PM.

Query template changes

The query represents the solutions for Gian to reach the University in Rovereto, and also to return at home in Trento from Rovereto. The chosen parameters are set to 60 minutes as *max_wait_time*, and 0.5 kilometers as the *max_stop_dist*.

```
BIND ("false"^^xsd:boolean as ?wheelchair_need)
# BIND ("60"^^xsd:long as ?max_wait_time) # Uncommented for return
# BIND ("0.5"^^xsd:float as ?max_stop_dist)
# BIND ("_urban" as ?stops_filter)
BIND ("0" as ?direction) # For outward while "1" for return

BIND ("2024-02-06T06:00:00"^^xsd:dateTime as ?curr_datetime)
# For outward while "2024-02-06T16:00:00" for return

BIND ("Gian" as ?username)
```

Results outward

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_9	"Trento 3 Nov.Centros.Chiara"	"Rovereto-Via Barattieri"	"B301"	"09:45:00"	"10:28:00"	"2"^^xsd:long	"23"^^xsd:int	"38"^^xsd:int	"43"^^xsd:long	"75.5"^^xsd:float	"NaN"
2	data:shift_9	"Trento 3 Nov.Centros.Chiara"	"Rovereto-Via Barattieri"	"B301"	"09:20:00"	"10:03:00"	"27"^^xsd:long	"23"^^xsd:int	"38"^^xsd:int	"43"^^xsd:long	"100.5"^^xsd:float	"NaN"
3	data:shift_9	"Trento 3 Nov.Centros.Chiara"	"Rovereto-Via Barattieri"	"B301"	"08:48:00"	"09:31:00"	"59"^^xsd:long	"23"^^xsd:int	"38"^^xsd:int	"43"^^xsd:long	"132.5"^^xsd:float	"NaN"

Results return

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_10	"Rovereto Via Paoli"	"Trento Via Perini"	"B332"	"15:43:00"	"16:26:00"	"4"^^xsd:long	"39"^^xsd:int	"28"^^xsd:int	"43"^^xsd:long	"80.5"^^xsd:float	"NaN"
2	data:shift_10	"Rovereto Via Paoli"	"Trento Ponte Cavaleggeri"	"B332"	"15:43:00"	"16:24:00"	"6"^^xsd:long	"39"^^xsd:int	"36"^^xsd:int	"41"^^xsd:long	"84.5"^^xsd:float	"NaN"

CQ5 - Gaia - Multiple shifts and schedules

Gaia lives at Sanbapolis. From Monday to Friday she needs to reach the University in Povo from 7:15AM to 1PM, but on Wednesday she goes to uni library in University of Sociology in Trento from 9:30AM to 10:30AM. However, on 2024-03-07 she will have a special lecture in University of Sociology in Trento at 12:30AM.

Query template changes

In this case,

```
BIND ("false"^^xsd:boolean as ?wheelchair_need)
BIND ("60"^^xsd:long as ?max_wait_time)
# BIND ("2"^^xsd:float as ?max_stop_dist)
BIND ("(_urban|scraped)" as ?stops_filter)
BIND ("1" as ?direction) # Sometimes 0 depending on the shift

BIND ("2024-03-05T06:00:00"^^xsd:dateTime as ?curr_datetime) # A "normal" tuesday
# "2024-03-06T10:00:00": A "normal" wednesday
# "2024-03-07T10:00:00": A "special" thursday

BIND ("Gaia" as ?username)
```

Results Sanbapolis - Povo (Monday)

Gaia takes the train at 06:11AM from Sanbapolis to reach the University in Povo at Monday morning

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	http://localhost:8080/source/shift_11	"san bartolameo"	"povo"	"Valsugana"	"06:11:00"	"06:20:30"	"54"^^xsd:long	"13"^^xsd:int	"51"^^xsd:int	"9"^^xsd:long	"95.0"^^xsd:float	"1"

Continues on next page...

Results Center of Trento - Sanbapolis (Wednesday)

Gaia, in this case, has different options to return at home in Sanbapolis from the University of Sociology in the center of Trento, where she has lecture every Wednesday.

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	http://localhost:8080/source/shift_14	"Rosmini "Cimitero" / Muse"	"Verona "Questura""	"13"	"10:30:00"	"10:39:00"	"0"^^xsd:long	"24"^^xsd:int	"18"^^xsd:int	"9"^^xsd:long	"30.0"^^xsd:float	"1"
2	http://localhost:8080/source/shift_14	"Travai Al Nuoto / Muse"	"Verona "Questura""	"3"	"10:34:00"	"10:42:00"	"4"^^xsd:long	"19"^^xsd:int	"18"^^xsd:int	"8"^^xsd:long	"30.5"^^xsd:float	"1"
3	http://localhost:8080/source/shift_14	"Rosmini S.Maria Maggiore"	"Verona "Questura""	"3"	"10:32:00"	"10:42:00"	"2"^^xsd:long	"22"^^xsd:int	"18"^^xsd:int	"10"^^xsd:long	"32.0"^^xsd:float	"1"
4	http://localhost:8080/source/shift_14	"Travai Piazza Fiera"	"Verona "Questura""	"3"	"10:35:00"	"10:42:00"	"5"^^xsd:long	"24"^^xsd:int	"18"^^xsd:int	"7"^^xsd:long	"33.0"^^xsd:float	"1"
5	http://localhost:8080/source/shift_14	"Rosmini "Cimitero" / Muse"	"Verona "Big Center""	"13"	"10:30:00"	"10:40:00"	"0"^^xsd:long	"24"^^xsd:int	"27"^^xsd:int	"10"^^xsd:long	"35.5"^^xsd:float	"1"
6	http://localhost:8080/source/shift_14	"Travai Al Nuoto / Muse"	"Verona "Big Center""	"3"	"10:34:00"	"10:43:00"	"4"^^xsd:long	"19"^^xsd:int	"27"^^xsd:int	"9"^^xsd:long	"36.0"^^xsd:float	"1"
7	http://localhost:8080/source/shift_14	"Rosmini S.Maria Maggiore"	"Verona "Big Center""	"3"	"10:32:00"	"10:43:00"	"2"^^xsd:long	"22"^^xsd:int	"27"^^xsd:int	"11"^^xsd:long	"37.5"^^xsd:float	"1"
8	http://localhost:8080/source/shift_14	"Travai Al Nuoto / Muse"	"Verona "Questura""	"8"	"10:41:00"	"10:49:00"	"11"^^xsd:long	"19"^^xsd:int	"18"^^xsd:int	"8"^^xsd:long	"37.5"^^xsd:float	"1"

Results of schedule exception on 2024-03-07 (Thursday)

Even if that day is a Thursday, since a schedule exception is presents, the results refer to a trip that goes from her home to Trento station which is near to the University.

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	http://localhost:8080/source/shift_15	"Verona "Questura""	"Piazza di Fiera"	"8"	"12:20:00"	"12:27:00"	"3"^^xsd:long	"18"^^xsd:int	"28"^^xsd:int	"7"^^xsd:long	"33.0"^^xsd:float	"1"
2	http://localhost:8080/source/shift_15	"Verona "Big Center""	"Piazza di Fiera"	"8"	"12:19:00"	"12:27:00"	"3"^^xsd:long	"26"^^xsd:int	"28"^^xsd:int	"8"^^xsd:long	"38.0"^^xsd:float	"1"
3	http://localhost:8080/source/shift_15	"Verona "Questura""	"Piazza di Fiera"	"3"	"12:12:00"	"12:19:00"	"11"^^xsd:long	"18"^^xsd:int	"28"^^xsd:int	"7"^^xsd:long	"41.0"^^xsd:float	"1"
4	http://localhost:8080/source/shift_15	"Verona "Big Center""	"Piazza di Fiera"	"3"	"12:11:00"	"12:19:00"	"11"^^xsd:long	"26"^^xsd:int	"28"^^xsd:int	"8"^^xsd:long	"46.0"^^xsd:float	"1"
5	http://localhost:8080/source/shift_15	"Verona "Questura""	"Piazza di Fiera"	"8"	"12:05:00"	"12:12:00"	"18"^^xsd:long	"18"^^xsd:int	"28"^^xsd:int	"7"^^xsd:long	"48.0"^^xsd:float	"1"
6	http://localhost:8080/source/shift_15	"Verona "Questura""	"Rosmini S.Maria Maggiore"	"13"	"11:58:00"	"12:07:00"	"23"^^xsd:long	"18"^^xsd:int	"20"^^xsd:int	"9"^^xsd:long	"51.0"^^xsd:float	"1"
7	http://localhost:8080/source/shift_15	"Verona "Big Center""	"Piazza di Fiera"	"8"	"12:04:00"	"12:12:00"	"18"^^xsd:long	"26"^^xsd:int	"28"^^xsd:int	"8"^^xsd:long	"53.0"^^xsd:float	"1"
8	http://localhost:8080/source/shift_15	"Verona "Questura""	"Rosmini "Cimitero" / Muse"	"13"	"11:58:00"	"12:05:00"	"25"^^xsd:long	"18"^^xsd:int	"27"^^xsd:int	"7"^^xsd:long	"54.5"^^xsd:float	"1"
9	http://localhost:8080/source/shift_15	"Verona "Big Center""	"Rosmini S.Maria Maggiore"	"13"	"11:57:00"	"12:07:00"	"23"^^xsd:long	"26"^^xsd:int	"20"^^xsd:int	"10"^^xsd:long	"56.0"^^xsd:float	"1"
10	http://localhost:8080/source/shift_15	"Verona "Big Center""	"Rosmini "Cimitero" / Muse"	"13"	"11:57:00"	"12:05:00"	"25"^^xsd:long	"26"^^xsd:int	"27"^^xsd:int	"8"^^xsd:long	"59.5"^^xsd:float	"1"

CQ6 - Carla - Flixbus (Torino)

Carla is an out-door-students and she lives in Lungadige in Trento. She return at home in Turin from Trento every Friday to stay at home in the weekend.

Query template changes

In this query, it is reported the best solution for Carla who every Friday returns to home in Turin from Trento. Here, the challenge of the implementation of the Flixbus's timetable. However, the data of Flixbus are not updated at 2024, so the date refers to a Friday in 2023.

```
BIND ("false"^^xsd:boolean as ?wheelchair_need)
# BIND ("60"^^xsd:long as ?max_wait_time)
# BIND ("2"^^xsd:float as ?max_stop_dist)
BIND ("flixbus" as ?stops_filter)
BIND ("1" as ?direction)

BIND ("2023-11-03T14:00:00"^^xsd:dateTime as ?curr_datetime) # A friday

BIND ("Carla" as ?username)
```

Results

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility
1	data:shift_17	"Trento (Lungadige Monte Grappa)"	"Turin (Vittorio Emanuele)"	"FlixBus 407"	"15:35:00"	"21:10:00"	"155"^^xsd:long	"136"^^xsd:int	"60"^^xsd:int	"335"^^xsd:long	"588.0"^^xsd:float	"1"

CQ7 - Fausto - Trenitalia (transfer)

Fausto is a Professor at University of Milan, and he needs to book a ticket train from Milano Centrale to reach the University of Sociology on the 13st, Nov 2023 (Monday) before 10:30AM to have an important seminary.

Query template changes

In this query, there is the result of Fausto's CQ. It is reported the implementation of Trenitalia's timetable, but as before, the train schedules are not updated, and so the date is 2023-11-13. Moreover, there is the management of a transport change. Indeed, Fausto has a Milan-Verona route, and then a train change to proceed from Verona to Trento. In this case, the nearest stop where Fausto arrives in Verona - and where he after leaves - is identified using not the object properties *nearest* but, instead, the geo distance. The results of the two queries are reported. In conclusion, the best option is the train which departs from Milan at 06:17, arriving in Verona at 8:17AM, departing at 8:50AM or 09:06AM from Verona to reach the University of Sociology in Trento.

```
BIND ("false"^^xsd:boolean as ?wheelchair_need)
BIND ("90"^^xsd:long as ?max_wait_time)
BIND ("0.2"^^xsd:float as ?max_stop_dist)
BIND ("trenitalia" as ?stops_filter)
BIND ("0" as ?direction)

BIND ("2023-11-13T04:00:00"^^xsd:dateTime as ?curr_datetime) # For the first trip
# "2023-11-13T08:00:00 for the second trip

BIND ("Fausto" as ?username)

...

# For the first trip:
## Using "nearest" COMMENTED
# ?edu_res_dom_to prop:localized_GID-89701 ?shift_pos_to.
# ?edu_res_dom_to prop:nearest_GID-116831 ?stop_pos_to.

## UNCOMMENTED
{ ## OR using geo distance
    {
        { ?stop_to a prop:bus_stop_GID-45937. }
        UNION
        { ?stop_to a prop:train_stop_GID-16010. }
    }
}

# For the second trip the previous has been done for ?stop_from
```

Results Milano - Verona

Fausto can arrive at Verona trains station either at 08:17 or 09:17.

	shift	stop_from_name	stop_to_name	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility	trip_ex	trip_ws	trip
1	data:shift_18	"MILANO CENTRALE"	"VERONA PORTA NUOVA"	"RE6"	"06:25:00"	"08:17:00"	"25"^^xsd:long	"19"^^xsd:int	"0"^^xsd:int	"112"^^xsd:long	"146.5"^^xsd:float	"NaN"		data:ws_trenitalia/lombardia_249472015856862	data:trip_trenitalia/lombardia_15856862
2	data:shift_18	"MILANO CENTRALE"	"VERONA PORTA NUOVA"	"RE6"	"07:25:00"	"09:17:00"	"85"^^xsd:long	"19"^^xsd:int	"0"^^xsd:int	"112"^^xsd:long	"206.5"^^xsd:float	"NaN"		data:ws_trenitalia/lombardia_249253515856004	data:trip_trenitalia/lombardia_15856004

Results Verona - Trento

Fausto can take a train to Trento either at 09:06 or 08:50. Given the results of the previous query, the best choice is:

$$\text{Milano 06:25} \Rightarrow \text{Verona 08:17 - wait until 08:50 (33 minutes)} \Rightarrow \text{Trento 09:50}$$

	shift	stop_from	stop_to	route_name	departure_time	arrival_time	wait_time	walk_from_m	walk_to_m	trip_duration	score	accessibility	trip_ex	trip_ws	trip
1	data:shift_19	data:stop_train_scraped_2173	data:stop_train_scraped_2264	"Verona-Bolzano"	"09:06:00"	"10:30:00"	"0"^^xsd:long	"3"^^xsd:int	"49"^^xsd:int	"84"^^xsd:long	"110.0"^^xsd:float	"1"		data:ws_scraped_543210001	data:trip_scraped_16678
2	data:shift_19	data:stop_train_scraped_2173	data:stop_train_scraped_2264	"Verona-Bolzano"	"08:50:00"	"09:50:00"	"40"^^xsd:long	"3"^^xsd:int	"49"^^xsd:int	"60"^^xsd:long	"126.0"^^xsd:float	"1"		data:ws_scraped_543210001	data:trip_scraped_16638

10 Metadata Definition

In this section, the report collects the definitions of all the metadata defined for the different resources produced along the whole process (producer and consumer). The metadata defined in this phase describes both the final outcome of the project, and the intermediate outcome of each phase.

The definition of the metadata, is crucial to enable the distribution (sharing) of the resource produced. For this reason, it is important to describe also where such metadata are published to distribute the resources it describes (for example the DataScientia catalogs).

In particular the structure of this section is organized in three parts: *Language resources*, *Knowledge resources*, and *Data resources*. The objective is to describe the metadata relative to all the type of resources produced by the project.

10.1 Language resources

About the Language resources, the aim is to refine the language (concepts and words) employed for representing the information needed to fulfill the project purpose.
As reported in the section 6 Language Definition, all is centered on creating and formalizing the concepts inside each individual dataset. The concepts are identified or created using the kge-annotator. For each (until now) used word/term, the tool is employed to identify the most appropriate concept by searching for synonyms or similar terms. If a sufficiently precise concept already existed in the UKC, it is adopted as is. However, in cases where concepts are either too general or completely absent, they are created as new. For more details about the language resources metadata, refer to the following tables: E-types (Table 7), Data properties (Table 8), and Object properties (Table 9)

10.2 Knowledge resources

The knowledge resources are produced for unifying the representation of the information, improving the interoperability and reusability of the final KG(s). By building knowledge resources reusing as much as possible well-known standard domain ontologies and data schema. As reported in the table below, there is an explanation of the used ontologies. However, it is necessary also to create an ontology refers to the part of the End-user, made by our team. For more details about the metadata knowledge resources, refer to the section 7 Knowledge Definition in the report.

Title	Description	Theme	Creator
OSM Lightweight Ontology	A lightweight ontology developed based on data from Open Street Maps	Upper Level	Xiaoyue Li
General Transit FeedSpecification	This ontology is a translation of the General Transit Feed Specification towards URIs. Its intended use is creating an exchange platform where the Linked GTFS model can be used as a start to get the right data into the right format	Upper Level	Pieter Colpaert, Andrew Byrd
Schema.org	It is a collaborative community activity with a mission to create maintain and promote schemas for structured data on the Internet	Upper Level	Community of people
Person Life	An ontology that connects individuals to their generic commitments	Purpose Level	Pasquali Thomas, Peiretti Riccardo

Table 19: Metadata summary for Knowledge Resources

10.3 Data resources

In the table below, there is a description of each dataset created to fulfill the project's purpose. In fact, the column *Creator* indicates our team of working, composed by Pasquali Thomas, and Peiretti Riccardo.

Title	Description	Theme	Creator
Provider	A dataset where there are some providers which supply a particular service	Purpose Level	Our team
Shift	A dataset where there are commitments of an user from a place to another	Purpose Level	Our team
End User	A dataset with elements referred to a person	Purpose Level	Our team
Schedule Exception	A dataset with the exception of something not conform to a schedule	Purpose Level	Our team
Weekly Schedule	A dataset about the weekly organized plan	Purpose Level	Our team
Bus Route	A dataset where there are the routes regularly follower by a bus	Purpose Level	Our team
Train Route	A dataset where there are the routes regularly follower by a train	Purpose Level	Our team
Position	A dataset where there are all the important elements about a place as address, latitude, and longitude	Purpose Level	Our team
Educational Facility	A dataset where there are the informations about the middle/high-school and university of the autonomous province of Trento	Purpose Level	Our team
Bus Stop	A dataset where we can find the places where a bus stop to discharge and take on passengers	Purpose Level	Our team
Train Stop	A dataset where we can find the places where a train stop to discharge and take on passengers	Purpose Level	Our team
Trip	A dataset which provides detailed information about travel routes	Purpose Level	Our team
Trip Stop	A dataset which includes information about stops along travel routes	Purpose Level	Our team

Table 20: Metadata summary for Data Resources

To conclude, *"metadata is structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use or manage an information resource."*[4]

Metadata, in general, has three main purposes: facilitate description of information resources, facilitate organization of information resources, and facilitate discovery of information resources.

For more details and information of metadata, refer to the Metadata Sheet on the GitHub Repository.

[4]Taken from NISO, 2017

11 Open Issues

Here are presented the most relevant issues of this project that (for lack of time and complexity), remained open along the iTelos process.

Shifts that require more than one trip: during the KG evaluation process (queries) have only been presented shifts that require exactly one trip (only CQ7 9.3 gives an idea of how trips can be combined). Though, it is common that a displacement requires more than a trip i.e. more than one transfer from a route to another. This case has been handled with multiple shifts that should be instead considered as one. To do so, we provide an intuition of what could be done to handle this situations: add to the existing KG a sub-graph that represents how routes can be concatenated to allow the creations of more complex trips. The example in fig. 14 would allow to execute queries like:

?from prop:transfer{3} ?to

for instance, if ?from has the value R1, ?to will get the values {R4, R5}.

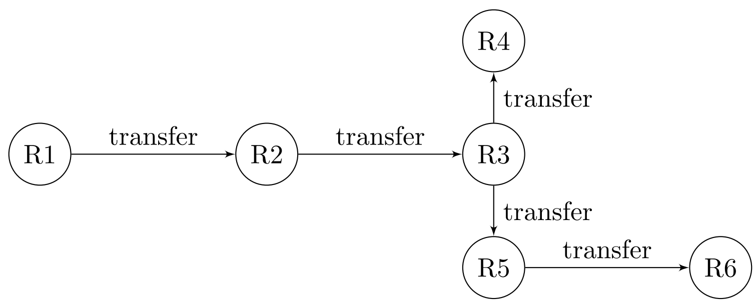


Figure 14: Routes concatenation example

This knowledge could then be used for "upgrading" the query template so that it gradually finds trips that belong to the selected routes, compatibly with time and departure/destination stops.

Query optimization: some ideas on how to further optimize queries:

- Divide the KG in sub-graphs that only cover specific regions for transportation;
- Add indexes to the KG.

Raise scraped data quality: either by creating better scripts or finding available datasets.
Update datasets: for example Trenitalia Lombardia trains.
Better data filtering: for instance, Flixbus non Italian trips and stops.
Find or create a better ontology for users.

12 Conclusion

Overall, the iTelos methodology allowed a consistent and valid development of this project. This report should allow the reader to understand the resources produced at each phase, their meaning and the reasoning that lead to the results. The purpose defined at the start of the project slightly changed but the main concepts remained unchanged. As shown in Section 9.3, the queries obtained from the CQs showed that the KG gives valid results therefore fulfilling the purpose.
Unluckily, references ontologies proved to be not the best for our situations, this may be due to our inexperience.

Lastly, we provide some personal considerations about the iTelos methodology and how it performed for this particular project.

Informal purpose and initial briefing

During the initial phases of the project, all ideas were abstract. We were not sure about their feasibility because of our inexperience on the field of knowledge graphs, nevertheless, we tried as much as possible to think in practical terms. This paid out during the following phases.

Purpose Formalization

As for the previous phase, feasibility uncertainty lead the process, only after the Knowledge definition everything became clear.

Information Gathering and Language Definition

This is the first phase where things became more concrete, still, not knowing what to expect from knowledge graphs (e.g. ontologies, sparql), we aimed as much as possible to automating the data processing so that changes could be made faster and more reliably.

At this point, the need of using a common way for referencing concepts was necessary. The way iTelos proposed this phase is very convenient.

Knowledge and Data Definition

During this phase, the purpose, initial ideas and scripts started to follow the right direction, eventually reaching their final version for this project.

Evaluation

This proved to be the most demanding phase, in fact all knowledge and data "weaknesses" came out. This allowed us to fix and improve the final project outcome making sure that the purpose is fulfilled.